Universidade do Porto
Faculdade de Engenharia
FEUP



Eduardo Alexandre Pereira da Silva

"Controlo de Veículos Autónomos"

Porto, 2002

CONTROLO DE VEÍCULOS AUTÓNOMOS

Eduardo Alexandre Pereira da Silva



Faculdade de Engenharia da
Universidade do Porto
2002

CONTROLO DE VEÍCULOS AUTÓNOMOS

Eduardo Alexandre Pereira da Silva

Dissertação elaborada para obtenção do grau de Doutor em
Engenharia Electrotécnica e de Computadores pela
Faculdade de Engenharia da Universidade do Porto

Orientador
Fernando Manuel Ferreira Lobo Pereira

Professor Associado com Agregação do Departamento de
Engenharia Electrotécnica e de Computadores pela
Faculdade de Engenharia da Universidade do Porto

621.3(043)/SIL e/CON
UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA M
N.º 73443
CDU 621.3(043)
Data 15 / 4 / 2005

às minhas princesas,

cristina

e

francisca.

O meu lado esquerdo
É mais forte do que o outro
É o lado da intuição
É o lado onde mora o coração

O meu lado esquerdo
Oriente do meu instinto
É o lado que me guia no escuro
É o lado com que eu choro e com que eu sinto

Meu , é o meu, foi o meu lado esquerdo
Que me levou até ti
Quando eu já pensava
Que não existias para mim no mundo

O meu lado esquerdo não sabe o que é a razão
É ele que me faz sonhar
É ele que tantas vezes diz não

Meu, é o meu, foi o meu lado esquerdo
Que me levou até ti
Quando eu já pensava
Que não existias para mim no mundo

Carlos Tê (com a música dos Clã)

Esta página foi intencionalmente deixada em branco

AGRADECIMENTOS

Agradeço ao meu orientador Professor Doutor Fernando Lobo Pereira, as oportunidades concedidas nos projectos e equipa de investigação que lidera, o espirito de equipa, a confiança que nunca esmoreceu, o rigor colocado no trabalho, as valiosas sugestões, as discussões estimulantes e acima de tudo a amizade com que orienta a minha emancipação.

Agradeço ao João Tasso de Sousa, a preciosa influência que exerce no meu percurso bem como no de toda a equipa, o entusiasmo que coloca nos projectos em que participa, as contribuições, a honestidade, o rigor, a frontalidade, a camaradagem e a amizade que sempre manifestou, bem como o papel estruturante no desenvolvimento dos Laboratórios de Investigação, (em especial para mim o LSA), e a esperança de um bom futuro científico.

Agradeço ao José Miguel Almeida e ao Alfredo Martins a amizade, a confiança, as noitadas, as sugestões, as contribuições, as soluções, a cumplicidade, a paciência para me ouvir, o percurso do Iseporto Team e o forte empenho que colocam no desenvolvimento do Laboratório dos Sistemas Autónomos do ISEP. Um bom futuro está a chegar.

Agradeço também ao:

João Paulo Baptista a sua amizade, a calma, a tranquilidade bem como a cumplicidade na aventura que é a criação do Laboratório dos Sistemas Autónomos;

Vitor Santos o seu empenho para nos proporcionar as condições necessárias no ISEP, para o desenvolvimento do Laboratório dos Sistemas Autónomos;

Oliveira e Sá o seu empenho para me proporcionar as condições necessárias no DEE do ISEP, para a conclusão deste trabalho;

A todos os membros do LSA, em especial ao Vitor, ao Carlos, ao Lima, ao Picas e ao Patacho, o entusiasmo que colocam no desnvolvimento das suas tarefas;

Ao Franklim o seu exemplo, o seu espírito crítico, bem como a confiança que sempre depositou nas ideias que partilhamos;

Ao Nuno Cruz e Anibal Matos a camaradagem e a colaboração no âmbito dos projectos comuns;

Ana Cristina Pinto, pela rigor e competência na gestão das burocracias e pelo tempo precioso que nos poupou....;

Não queria terminar sem agradecer às minhas princesas Cristina e Francisca as horas que lhes roubei, todas as coisas que não posso dizer aqui, mas acima de tudo por ajudarem a transformar o mundo para melhor.

Também não posso esquecer a minha irmã Micas, os meus irmãos Cocas e Zé, pela confiança que sempre depositaram no irmão.

Aos meus Pais Eduarda e Alexandre as suas opções e a visão do futuro para os vossos filhos.

Aos meus Sogros e amigos Adelaide e Ricardo, a sua amizade, o seu apoio e acima tudo por terem resolvido fazer uma menina que é a mulher que eu amo.

Esta página foi intencionalmente deixada em branco

ABSTRACT

This thesis addresses the design, development and implementation of autonomous systems with emphasis on the control perspective. Two directions were considered: While in one of them, two different applications autonomous transportation in an industrial environment and underwater autonomous motion were addressed, in the other an effort towards the systematisation for which it is essential to consider a formal framework was sought.

Given the considerable ambition of this objective, the treatment of the case studies were essential to evaluate the properness of the various formalisms for the representation, analysis and synthesis for the partial problems arising in the design of the multiple instances associated with the addressed autonomous systems.

In the pursuit of the main aim of this dissertation, several contributions were obtained in addressing the partial objectives in which the main goal was decomposed. They are as follows:

- specification of performance requirements for the system, environment description and characterisation of the functional system to environment interaction;
- analysis of viable system strategies fulfilling of the specified requirements;
- specification of structural properties underlying the behaviours of the autonomous system operation;
- definition of a control architecture for the of the autonomous system activities enabling an implementable solution;
- demonstration of the approach in the context of the two addressed concrete applications;
- extension of the approach to control problems involving multiple interacting autonomous systems;
- refinement of the systems engineering process for the design of autonomous systems in order to take into account the specificities embedded systems.

In parallel, a scenario for the intervention of autonomous underwater vehicles in environmental monitoring was identified. A novel solution is

presented for the addressed scenario, which involves requirements of active search. The adopted approach is based on the defined autonomous systems design methodology. This approach permitted the development of contexts enabling the integration of concepts and methods of different areas, which contribute to the emergence of a solution.

The work described in this dissertation encompasses the developments under the Po-Robot project carried out by the activities of Underwater Systems and Technologies Laboratory (LSTS) of FEUP, and, more recently in cooperation with Autonomous Systems Laboratory (LSA) of ISEP.

RESUMÉ

Cette thèse adresse la conception, le développement et l'exécution des systèmes autonomes avec l'accent sur la perspective du contrôle. Deux directions ont été considérées: Tandis que dans un d'elles, deux applications différentes, le transport autonome dans un environnement industriel et le mouvement autonome sous-marin étaient adressés, dans l'autre, ont été cherché un effort pour la systématisation sur le qui il est essentiel considère un cadre formel. Etant donné l'ambition considérable de cet objectif, le traitement des études de cas étaient essentiel pour évaluer si les divers formalismes sont ajustés pour la représentation, l'analyse et la synthèse pour les problèmes partiels surgissant dans la conception des exemples de multiple instances aux systèmes autonomes adressés.

À la poursuite du but principal de cette dissertation, plusieurs contributions ont été obtenues en adressant les objectifs partiels dans lesquels le but principal a été décomposé. Elles sont comme suit:

spécifications des conditions d'exécution pour le système, la description d'environnement et la caractérisation du système fonctionnel à l'interaction d'environnement;

analyse d'accomplir viable de stratégies de système des conditions définies;

spécifications des propriétés structurales sous-tendantes les comportements de l'opération de l' autonomous system;

définition d'une architecture de commande pour des activités de autonomous system permettant une solution implantable;

la démonstration de l'approche dans le contexte des deux applications concrètes a été adressé;

Extension de l'approche du problèmes de contrôle au le cadre de les multiples systems;

En parallèle, un scénario pour l'utilisation des véhicules sous-marins autonomes dans le contrôle de l'environnement a été identifié. Une novel solution est présentée pour le scénario adressé qui implique des conditions de recherche active. L'approche adoptée est basée sur la méthodologie de

conception autonome définie de systèmes. Cette approche a permis le développement des contextes qui permettant l'intégration des concepts et des méthodes de différents secteurs qui contribuent à l'apparition d'une solution.

Le travail décrit dans cette dissertation entoure les développements sous le projet de PO-Robot mis à exécution par les activités du laboratoire sous-marins de technologies (LSTS) de FEUP, et, plus récemment en coopération avec le laboratoire autonome de systèmes (LSA) d'ISEP.

RESUMO

Esta tese endereça a concepção, o desenvolvimento e a implementação de sistemas autónomos com ênfase na perspectiva de controlo. Foram privilegiadas duas vertentes: Enquanto que numa, foram endereçadas duas aplicações concretas respeitantes ao transporte autónomo num ambiente industrial semi-estruturado e à movimentação submarina autónoma, na outra, procurou-se levar a cabo um esforço de sistematização para o qual se torna essencial a consideração de um quadro formal. Dada a extrema ambição deste objectivo foi procurado, com base no esforço de tratamento dos casos estudo, avaliar a adequabilidade de vários formalismos para a representação, análise e síntese de problemas parciais para o projecto das instâncias de sistemas autónomos endereçadas.

Na prossecução do objectivo fundamental desta dissertação, foram realizadas contribuições num conjunto de objectivos parciais integrantes de um quadro para a análise e para a síntese de sistemas autónomos em que destacamos, a:

especificação das necessidades de desempenho do sistema, a descrição do ambiente, a caracterização da interacção funcional do sistema com o meio;

análise das estratégias viáveis do sistema tendo em vista o preenchimento dos requisitos especificados;

especificação das propriedades estruturais subjacente aos comportamentos a obter na operação do sistema autónomo;

definição de uma arquitectura para o controlo das actividades do sistema autónomo tendo em vista a viabilização de uma solução implementável;

comprovação da abordagem através do endereçamento de sistemas para duas aplicações concretas referidas na secção anterior;

extensão da abordagem para problemas de controlo envolvendo o interacção de múltiplos sistemas autónomos;

refinamento do processo de engenharia de sistemas para o projecto de sistemas autónomos tendo em vista dos as especificidades dos sistemas "embedded".

Paralelamente identificamos um cenário para intervenção de veículos submarinos autónomos na monitorização e estudos ambientais que carecem de pesquisa activa, para os quais apresentamos uma inovadora solução.

A abordagem adoptada tem como base a análise do desenvolvimento do projecto de sistemas autónomos, pois este constitui um núcleo privilegiado para o desenvolvimento conceptual essencial à definição de uma metodologia de projecto baseada num quadro formal. Esta procura desenvolver contextos em que se torna possível a integração de conceitos e métodos de diferentes áreas científicas concorrentes para a definição de soluções, contribuindo assim para a procura de um quadro de projecto unificador desde a especificação sistémica até à implementação da solução.

O trabalho descrito nesta dissertação foi consubstanciado no desenvolvimento do projecto Po-Robot, nas actividades de investigação do Laboratório de Sistemas e Tecnologia Subaquática da FEUP e mais recentemente em conjugação com as actividades do Laboratório de Sistemas Autónomos do ISEP.

INDICE

AGRADECIMENTOS		III
ABSTRACT		V
RESUMÉ		VII
RESUMO		IX
INDICE		XI
1	INTRODUÇÃO	1
1.1	O ÂMBITO DA DISSERTAÇÃO	2
1.2	OBJECTIVOS E CONTRIBUIÇÕES	4
1.3	A ABORDAGEM	5
1.4	ORGANIZAÇÃO DA DISSERTAÇÃO	6
PARTE I		
ANÁLISE DE SISTEMAS AUTÓNOMOS		9
2	CARACTERIZAÇÃO DO AMBIENTE INDUSTRIAL PARA O TRANSPORTE AUTÓNOMO	11
2.1	DESCRIÇÃO DO PROJECTO AMSIA – AUTONOMOUS MOBILE SYSTEM FOR INDUSTRIAL APPLICATIONS	12
2.2	ESPECIFICAÇÃO DE NECESSIDADES DE DESEMPENHO DO SISTEMA	13
2.3	CARACTERIZAÇÃO DO AMBIENTE	14
2.3.1	Classes de padrões de restrições geométricas	16
2.3.2	Classes de elementos característicos associados ao ambiente	18
2.4	CARACTERIZAÇÃO DAS RELAÇÕES DO SISTEMA AUTÓNOMO COM O AMBIENTE	19
2.5	CARACTERIZAÇÃO DAS FUNCIONALIDADES REQUERIDAS	20
2.6	CARACTERIZAÇÃO DOS TIPOS DE MOVIMENTO	23
2.7	CARACTERIZAÇÃO DOS TIPOS DE MOVIMENTOS EM GRUPO	26
2.8	COMENTÁRIOS À CARACTERIZAÇÃO DO MOVIMENTO	26
2.9	QUESTÕES SOBRE A CAPACIDADE DE DECISÃO	27
2.10	REQUISITOS DO MOVIMENTO	28
2.11	REQUISITOS ASSOCIADOS À PERCEPÇÃO. LOCALIZAÇÃO E NAVEGAÇÃO	30
2.12	REQUISITOS INERENTES A RESTRIÇÕES TECNOLÓGICAS	32
2.13	ESPECIFICAÇÃO DA MISSÃO	36
3	CARACTERIZAÇÃO DE AMBIENTES HÍDRICOS PARA A MOVIMENTAÇÃO SUBMARINA AUTÓNOMA	37
3.1	INTRODUÇÃO	38
3.2	O PROBLEMA	39
3.3	CARACTERIZAÇÃO DO AMBIENTE	41

3.4 ESPECIFICAÇÃO DE NECESSIDADES DE DESEMPENHO DO SISTEMA 43
3.5 ESTRUTURAR O MEIO E O MOVIMENTO 44
3.6 CARACTERIZAÇÃO DE MOVIMENTOS PARA ESTRATÉGIAS ACTIVAS 47
3.6.1 Caracterização do fenómeno versus movimentação 47
3.6.2 Movimentação cooperativa 51
3.6.2.1 Utilização do gradiente 51
3.6.3 Movimentação em formação 55
3.6.3.1 Estrutura para a movimentação em formação 55
3.6.3.2 Detecção de indícios da existência do fenómeno 56
3.6.3.3 Percepção e estado da formação 57
3.6.3.4 Como seguir os indícios 60
3.7 REQUISITOS DO MOVIMENTO 62
3.8 REQUISITOS PARA A PERCEPÇÃO, LOCALIZAÇÃO E NAVEGAÇÃO 63

4 ESPECIFICAÇÃO DOS PROBLEMAS DE CONTROLO 69
4.1 INTRODUÇÃO 70
4.2 COMO ORGANIZAR O MOVIMENTO 71
4.3 CONTROLO DE MOVIMENTO 72
4.3.1 Manobras associadas aos ambientes industriais semi-estruturados 73
4.3.2 Movimentos elementares (Comando de movimento) para manobras de ambientes industriais semi-estruturados 77
4.3.3 Manobras associadas para movimentação ambientes submarinos virtualmente semi estruturados . 78
4.3.4 Manobras associadas à movimentação cooperativa 80
4.3.5 Movimentos elementares para ambientes submarinos com cariz geométrico. 80
4.3.6 Manobras associadas a estratégias de movimento baseadas em formações de veículos. 81
4.3.6.1 Pesquisa indícios 81
4.3.7 Movimentos elementares associados às manobras das formações 89
4.4 ESTIMAÇÃO E CONTROLO DE SUBSISTEMAS 97
4.4.1 Estimação 97
4.4.2 Estimação da estrutura da formação 97
4.4.3 Recursos 97
4.5 COORDENAÇÃO DOS SUBSISTEMAS 98

PARTE II
ESTADO DA ARTE 101

5 TRABALHO RELACIONADO 103
5.1 ARQUITECTURAS DE MODELO FUNCIONAL 104
5.1.1 A arquitectura de modelo de referência 104
5.1.2 A arquitectura ATLANTIS 105
5.1.3 A arquitectura DAMN 106
5.1.4 A arquitectura hierárquica de controlo inteligente 107
5.1.5 A arquitectura desenvolvida no LAAS 109
5.1.6 A arquitectura de "subsumption" 110
5.1.7 A arquitectura Fate e KAMARA 111
5.1.8 A arquitectura AuRA 111

5.1.9 Arquitectura por camadas e controlo híbrido 113
5.1.10 Arquitecturas baseadas em Redes Neuronais 114
5.1.11 Arquitectura hierárquica da NPS 115
5.2 CONTEXTOS DE DESENVOLVIMENTO E LINGUAGENS DE "TEMPO-REAL" 116
5.2.1 ControlShell 116
5.2.2 Chimera 117
5.2.3 Orccad 118
5.2.4 Matlab – Real Time Workshop 119
5.2.5 TEJA 120

PARTE III
ARQUITECTURA PARA UM SISTEMA AUTÓNOMO **123**

6 ABORDAGEM AO PROJECTO DE SISTEMAS AUTÓNOMOS **125**
6.1 INTRODUÇÃO 126
6.2 SISTEMA AUTÓNOMO 127
6.3 O PROJECTO DE SISTEMAS AUTÓNOMOS 129
6.3.1 Metodologias de Projecto 129
6.3.2 Cenário para o desenvolvimento de uma metodologia 131
6.4 FORMALIZAÇÃO DA METODOLOGIA 132
6.5 QUADRO METODOLÓGICO PROPOSTO 133
6.6 PRINCÍPIOS DE PROJECTO DE SISTEMAS 136
6.7 ARQUITECTURA CONCEPTUAL DE CONTROLO 139
6.7.1 Modelização 142
6.8 ALGUMAS QUESTÕES SOBRE ARQUITECTURAS DE CONTROLO 145
6.9 CONTRIBUIÇÃO PARA ESTABELECIMENTO DE UM QUADRO SEMÂNTICO 149
6.9.1 Problema 149
6.9.2 Como compatibilizar estruturas infinitas com estruturas finitas? 150
6.9.3 Sistemas dinâmicos congruentes 152
6.9.4 Agregação Hierárquica 155
6.9.5 Como utilizar este quadro semântico 157

7 ARQUITECTURA **161**
7.1 ARQUITECTURA IMPLEMENTÁVEL DE CONTROLO 162
7.2 ALGUMAS QUESTÕES SOBRE A IMPLEMENTAÇÃO DE SISTEMAS DE CONTROLO 165
7.3 INTRODUÇÃO AO NÍVEL FUNCIONAL 166
7.3.1 Componente da camada funcional 168
7.1.2 Funcionalidades Elementares 169
7.1.3 Funcionalidades Complexas. 172
7.1.4 Controladores de recursos 176
7.4 ESTÁGIO DE COORDENAÇÃO 185
7.4.1 Modelização de Tarefas 185
7.4.2 Outros aspectos caracterizadores 187
7.4.3 Modelização dos Operadores Composição de Tarefas 188
7.4.4 Paralelismo Forte 189
7.4.5 Paralelismo Fraco 192

7.4.6 Sequenciamento Forte 193
7.4.7 Sequenciamento Fraco 195
7.4.8 Sequenciamento condicional 197
7.4.9 Composição cíclica condicional 199

PARTE IV
FERRAMENTAS PARA A SÍNTESE DE PROBLEMAS DE CONTROLO **203**

8 FORMULAÇÃO DOS PROBLEMAS DE CONTROLO **205**
8.1 INTRODUÇÃO 206
8.1.1 Estratégias de controlo 207
8.1.2 Caracterização de acções de controlo 208
8.2 CONTROLADORES PARA A EXECUÇÃO DE MANOBRAS 209
8.3 CONTROLADORES HÍBRIDOS PARA A DEFINIÇÃO DE UMA MANOBRA 211
8.3.1 Mecanismos de síntese 212
8.3.2 Atingibilidade 213
8.3.3 Invariância 215
8.3.4 Controlo por realimentação 217
8.3.5 Controlo óptimo 219
8.3.6 Coordenação de sistemas dinâmicos 221
8.4 EXEMPLO DE UTILIZAÇÃO DA ESTRUTURA OFERECIDA PELA ARQUITECTURA PARA A IMPLEMENTAÇÃO DE UMA MANOBRA 223
8.4.1 Formulação do Problema – Uma Curva em 'L' 223
8.4.2 Síntese do controlador híbrido 224
8.4.3 Organização do controlador híbrido 225
8.4.4 Implementação no enquadramento do controlador de recurso de movimento da classe 1C1SnA 227

9 CONCLUSÕES **231**

REFERÊNCIAS **235**

ANEXO: PRELIMINARES ALGÉBRICOS **247**
A.1 *POSETS* 249
A.2 DIAGRAMAS DE HASSE 250
A.3 LATTICES 250
A.4 RELAÇÕES DE EQUIVALÊNCIA 252
A.5 NÚCLEO DE EQUIVALÊNCIA E FACTORIZAÇÃO CANÓNICA 255
A.6 "SPRAYS" E "COVERS" 259

1 INTRODUÇÃO

1.1 O âmbito da dissertação

Esta tese endereça a concepção, o desenvolvimento e a implementação de sistemas autónomos com ênfase na perspectiva de controlo. Dado o estádio do desenvolvimento dos conhecimentos e dos métodos científicos e de engenharia nesta vasta e complexa área, foi efectuada uma opção clara de contribuir em duas vertentes distintas e complementares.

Por um lado, foram abordados dois casos estudo endereçando duas aplicações concretas, nomeadamente, o transporte autónomo num ambiente industrial semi-estruturado, e a movimentação submarina autónoma. Considerando a relativa insipiência das metodologias actualmente propostas, qualquer esforço de sistematização deverá proceder de forma cautelosa no que respeita à consideração das especificidades do domínio do problema. Este esforço revelou-se instrumental na isenção necessária para investigar as questões na perspectiva multidisciplinar fundamental para engenharia e, em particular, a acuidade no desenvolvimento da metodologia de projecto de sistemas.

Por outro lado, a sistematização tão natural à abordagem científica ao esforço de concepção requer um quadro formal que, dada a extrema ambição do objectivo proposto, se encontra ainda longe de se estabelecer na prática de engenharia. Assim, procuramos, com base no esforço de endereçamento dos casos estudo, avaliar a adequabilidade de vários formalismos para a representação, análise e síntese de problemas parciais para o projecto das instâncias de sistemas autónomos endereçadas.

Contexto externo

Sendo o projecto de sistemas autónomos e, mais especificamente, a organização e o controlo do movimento, o tema central desta tese, obviamente que se torna fundamental contribuir para a sistematização do processo que se inicia na especificação e se completa com a implementação de um sistema-solução que preenche os requisitos definidos.

A definição de um método de projecto que permita endereçar a complexidade desta realidade, preencher os requisitos pretendidos, bem como garantir as funcionalidades enunciados com níveis de desempenho competitivos obriga à:

a) Especificação de processos de modelização envolvendo uma estrutura de representações, abrangendo diferentes níveis de abstracção que capture as diferentes vertentes necessárias a considerar.

b) Definição de métodos de síntese de estratégias de controlo logicamente verificáveis que integrem comportamentos fisicamente viáveis, satisfazendo os requisitos definidos para cada uma das missões subjacentes à concepção do sistema.

Contexto formal

A conceptualização dos problemas de controlo enunciados envolve os seguintes tipos de ingredientes: Mapeamento da realidade física numa representação que consiste numa estrutura endereçando vários níveis de abstracção, definição de métodos de síntese logicamente verificados para cada contexto abstracto considerado, correspondência entre cada estado abstracto e as possíveis realizações físicas.

Assim, para cada contexto constituinte da representação da evolução do sistema, importa identificar um conjunto finito e completo de situações. Uma missão viável pode ser expressa como uma sequência de pares (situação, transição) de forma a ligar o estado final ao inicial satisfazendo todas as restrições. Exemplos de restrições são: Características dinâmicas, obstruções espaciais (estáticas e topológicas), temporais, lógicas, etc.

Assim numa perspectiva lógica do projecto de controlo para um sistema autónomo, torna-se necessário sintetizar sistemas capazes de:

a) perceber as situações, ou seja o estado do sistema no contexto da execução da missão,

b) gerar decisões lógicas que permitam o estado do sistema transitar da melhor maneira, ou seja, viabilizando um conjunto de requisitos ou optimizando um critério de desempenho subjacente à concepção do sistema.

Observe que o primeiro passo consiste numa redução de informação orientada, uma vez que, o conjunto de dados recolhidos da realidade física é extremamente vasto e dos quais se torna necessário extrair uma interpretação numa base abstracta, relativamente reduzida, constituída pelos elementos de definição dos estados possíveis do sistema.

O segundo passo constitui num movimento inverso, uma vez que há que seleccionar o estado na representação abstracta do sistema ao qual

corresponde um conjunto de realizações físicas mais vantajoso para o sucesso da execução da missão. Note-se que se refere um conjunto, uma vez que, para além da incerteza na execução física, poderá verificar-se o não determinismo (no sentido, de classes de equivalência) no mapeamento entre a representação abstracta e a realização concreta.

1.2 Objectivos e contribuições

Na prossecução do objectivo fundamental desta dissertação, foram considerados um conjunto de objectivos parciais integrantes de um quadro para a análise e para síntese de sistemas autónomos abaixo listados. A cada um destes foram produzidas contribuições:

a) Levantamento de requisitos que inclui, a especificação das necessidades de desempenho do sistema, a descrição do ambiente, a caracterização da interacção funcional do sistema com o meio;

b) Análise das estratégias viáveis do sistema tendo em vista o preenchimento dos requisitos especificados. Esta é efectuada de forma a incorporar as características dos mecanismos de percepção e de actuação e no contexto emergente da execução cujo sucesso é medido no grau de satisfação dos requisitos;

c) Especificação das propriedades estruturais subjacente aos comportamentos a obter na operação do sistema autónomo de forma a satisfazer os requisitos considerados;

d) Definição de uma arquitectura para o controlo das actividades do sistema autónomo tendo como base, por um lado a sistematização das actividades de projecto tendo em vista os resultados de c) e, por outro, a viabilização de uma solução implementável;

e) Comprovação da abordagem através do endereçamento de sistemas para duas aplicações concretas referidas na secção anterior;

f) Perspectivação e esboço da extensão da abordagem para problemas de controlo envolvendo o interacção de múltiplos sistemas autónomos.

g) Refinamento de princípios metodológicos associados ao processo de projecto de sistemas autónomos envolvendo o processo de engenharia de sistemas, com particular ênfase para os sistemas "embedded".

1.3 A abordagem

A abordagem adoptada tem como base a análise do desenvolvimento do projecto de sistemas autónomos, pois este constitui um núcleo privilegiado para o desenvolvimento conceptual essencial para a definição de uma metodologia de projecto baseada num quadro formal.

É de realçar o peso do carácter da observação experimental dos resultados de soluções encontradas nas diferentes fases dos projectos considerados, paralelamente à busca de um quadro teórico que formalize procedimentos de projecto e síntese de soluções.

Se bem que seja um desafio motivador, a interdisciplinaridade associada também constitui um factor de dificuldade na definição de formalismos e conceitos unificadores. Assim, procuramos desenvolver contextos em que se torna possível a integração de conceitos e métodos de diferentes áreas científicas concorrentes para a definição de soluções, contribuindo assim para a procura de um quadro de projecto unificador desde a especificação sistémica até à implementação da solução.

A metodologia que se amadureceu no decorrer das actividades de engenharia relatadas nesta tese tem como base o processo de engenharia de sistemas [IEEE 98] – consistindo num conjunto de tarefas que transforma os objectivos, requisitos e restrições descritos pelo utilizador num sistema-solução – que foi sendo modificado tendo em conta tirar o melhor partido possível dos pontos fortes e fracos no que respeita à classe de sistemas endereçados.

No que respeita ao quadro formal, a nossa abordagem consistiu em identificar o contexto que melhor serve a articulação dos mecanismos de síntese de estratégias de controlo envolvendo ambos os aspectos lógicos e quantitativos essenciais para tratar a classes de sistemas endereçados.

O trabalho descrito nesta dissertação foi consubstanciado no desenvolvimento do projecto Po-Robot [Po-Robot 95] e das actividades de investigação do Laboratório de Sistemas e Tecnologia Subaquática da FEUP [LSTS 00]. Paralelamente alguns destes resultados já foram utilizados na plataforma Utopia desenvolvida no Laboratório de Sistemas Autónomos do ISEP [LSA 01].

1.4 Organização da dissertação

A apresentação do trabalho aqui exposto far-se-á em quatro partes. A primeira endereça os casos de estudo sobre os quais realizamos as nossas reflexões, e fundamentou a especificação do problema de controlo e de coordenação de sistemas autónomos. A segunda parte descreve as principais abordagens e soluções para o quadro de controlo desenvolvida pela comunidade científica nos últimos anos. A terceira apresenta a metodologia e a arquitectura de referência proposta, bem como a discussão para a sua fundamentação. Finalmente, na quarta parte apresentamos organizadamente alguns métodos e estratégias de síntese para a implementação e projecto de controladores.

As alternativas possíveis para leitura desta dissertação, tendo em vista a sustentação do esforço do leitor, estão graficamente representadas na figura seguinte.

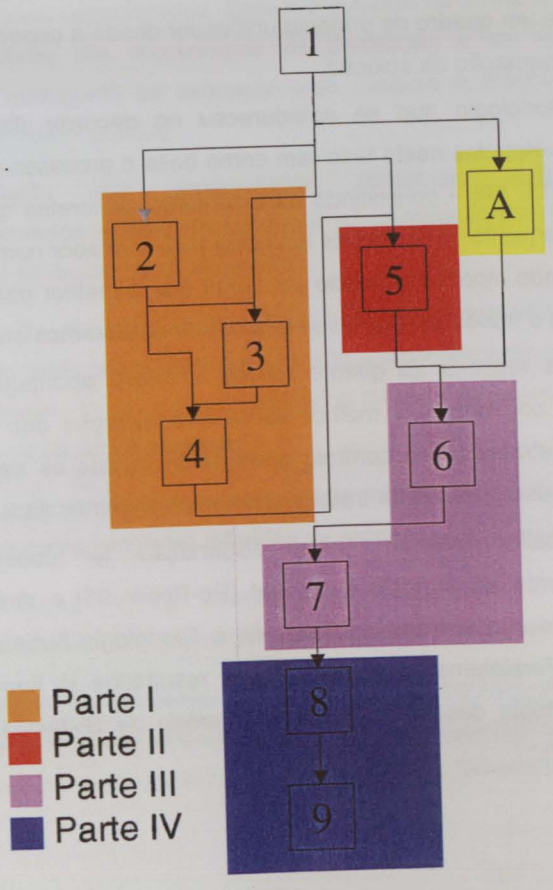


Figura 1.4-1

Percursos alternativos para a leitura desta dissertação

PARTE I – Análise de sistemas autónomos

Apresentamos os seguintes casos de estudo:

- Sistema móvel autónomo para aplicações industriais, no capítulo 2;
- Movimentação submarina autónoma, no capítulo 3.

A forma como estes casos de estudo são abordados não é independente da metodologia proposta para o projecto de sistemas de controlo, pois estes servem simultaneamente como base de reflexão e como "test bed" experimental das conjecturas. Consequentemente, está presente ao longo destes capítulos um certo grau de ambiguidade na precedência entre a metodologia e a análise dos casos de estudo.

Iniciamos os capítulos 2 e 3, com uma caracterização do ambiente e das relações do sistema com este, para, na parte final, descrevermos os requisitos que se colocam à síntese de sistemas autónomos.

Por fim, no capítulo 4, apresentamos um elenco de especificações para problemas de controlo de movimento classificado segundo a abordagem metodológica adoptada, ou seja, um conjunto de manobras e um conjunto de leis de controlo elementares.

PARTE II – Estado da arte

No quinto capítulo apresentamos uma sucinta revisão do domínio das arquitecturas dos sistemas robóticos, com especial atenção à natureza metodológica do desenvolvimento e projecto. No que respeita às outras áreas disciplinares, optou-se pela distribuição ao longo do texto de referências às relações que o nosso trabalho apresenta com trabalhos de outros grupos de investigação. Tal opção deve-se à grande heterogeneidade disciplinar subjacente à classe de problemas endereçadas na dissertação.

PARTE III – Arquitectura para um sistema autónomo

Nesta parte do trabalho, foram apresentadas considerações sobre as linhas gerais de uma metodologia de projecto, bem como a caracterização em detalhe uma arquitectura desenvolvida para um quadro de implementação de sistemas autónomos.

A questão essencial subjacente às considerações do binómio metodologia – arquitectura consiste em adoptar uma estrutura que permita a gestão adequado do compromisso expressividade – complexidade.

PARTE IV – Ferramentas para a síntese de problemas de controlo

Dentro do quadro definido para o projecto, apresentamos nesta parte uma panóplia de resultados e métodos de teoria do controlo de sistemas dinâmicos - cuja evolução poderá ser referida quer a acontecimentos discretos quer ao tempo - que constituem a melhor opção para incorporar os elementos de carácter quantitativo e lógico essenciais à classe de problemas abordadas nesta dissertação. Estes têm em conta a estrutura metodológica de projecto proposta que se centra no conceito de manobra. Dessa forma, são apresentadas diferentes ferramentas matemáticas a utilizar na formulação e síntese de diferentes leis de controlo e de acontecimentos relevantes. Finalmente, concluímos esta dissertação com um capítulo onde realizamos algumas observações e comentários ao trabalho desenvolvido, bem como uma perspectiva de trabalho futuro.

PARTE I

Análise de Sistemas Autónomos

"... o método de análise que empreguei ... torna bastante árdua a leitura dos primeiros capítulos, e é de temer que o público, sempre impaciente por concluir, ávido de conhecer a relação dos princípios gerais com as questões imediatas que o apaixonam, desanime, por não poder passar logo adiante. É essa uma desvantagem contra a qual nada posso fazer senão prevenir e premunir os leitores preocupados com a verdade. Não há estrada real para a ciência e só têm possibilidade de chegar aos seus cumes luminosos aqueles que não temem fatigar-se a escalar as suas veredas escarpadas."

extracto da carta de Karl Marx ao cidadão francês Lachâtre editor da primeira edição em língua francesa do volume I do "O Capital".

Esta página foi intencionalmente deixada em branco

2 CARACTERIZAÇÃO DO AMBIENTE INDUSTRIAL PARA O TRANSPORTE AUTÓNOMO

2.1 Descrição do Projecto AMSIA – Autonomous Mobile System for Industrial Applications

O objectivo principal deste projecto consiste em conceber e desenvolver sistemas móveis para ambientes industriais. Um exemplo típico consiste no transporte autónomo num ambiente industrial. Este problema foi endereçado inicialmente no âmbito do projecto PO-ROBOT [Po-Robot 95] que consistia em desenvolver uma plataforma móvel para uso industrial que sem a intervenção humana fosse capaz de:

- deslocar-se a um conjunto de pontos especificados evitando obstáculos;
- efectuar um conjunto de operações tais como carga e descarga de materiais;
- accionar dispositivos de carga ou descarga, ou;
- accionar ferramentas.

As capacidades referentes à autonomia focavam a problemática da movimentação em ambientes semi-estruturados e variantes no tempo, bem como a capacidade de estabelecer comunicação com os sistemas com os quais irá cooperar [Po-Robot95].

A extensão deste problema a mais que um veículo endereça questões como a gestão de frota, a coordenação de veículos, e o estabelecimento de regras "sociais" associados ao comportamento dos veículos.

O projecto AMSIA é a continuação do desenvolvimento dos resultados promissores do Po-Robot, e resulta em vários objectivos de trabalho em que há a destacar:

- Desenvolvimento de uma abordagem ao projecto que, assentando num quadro formal apropriado, tire o máximo partido da necessária integração disciplinar.
- Concepção de sistemas de percepção avançados que, para além do processamento usual de dados sensoriais de apoio à navegação, permita reconhecer características do ambiente mais ou menos estruturado.
- Modelização híbrida do sistema autónomo no contexto de um ambiente semi-estruturado e desenvolvimento de mecanismos para a manutenção do conhecimento do estado do sistema.

- Geração de estratégias de controlo do modelo referido que sejam consistentes para a prossecução da missão.
- Demonstração da abordagem num caso concreto de transporte industrial autónomo.

2.2 Especificação de necessidades de desempenho do sistema

A resposta tecnológica ao desafio de automatizar o transporte industrial têm se revelado fraca e pouco promissora (vejamos por comparação aos sistemas de comunicação). Se pelo lado conceptual a ideia parece ser óbvia, pela lado da implementação parece ser questionável.

Paralelamente ao desenvolvimento do transporte industrial autónomo, no mesmo perfil de desenvolvimento tem surgido com alguma eficácia o transporte automático de passageiros em sistemas de "Metro", "Eléctricos Urbanos", e Caminhos de Ferro Suburbanos. Estes sistemas são claramente caracterizados pela forte estruturação do meio, o que significa um domínio semântico limitado, logo, sistema de decisão mais simples.

O transporte industrial autónomo sofre o primeiro impulso com os AGV's, que vêm trazer uma nova perspectiva para a automatização industrial, mas cedo se verificou, que a relação "funcionalidade/custos de capital/operação", não permitia a sua aplicação em qualquer situação. Complementarmente algumas das vantagens que eram apresentadas, (como por exemplo em relação aos "convoyer") de possuir um "re-routing" mais fácil não se verificou, dada a dependência do movimento do sistema com o "fio condutor". Ainda se verifica que a operação com estes sistemas está muito dependente do sistema a integrar, pelo que a aquisição "off-the-shelf" está totalmente inviabilizada. Ou se compra a solução transporte/armazenagem, ou é necessário adaptar o sistema para a utilização de um AGV específico. Outro factor que "perturba" a sua utilização é normalização do sistema de transporte e transferência de material (soluções tipo "paletes" vs empilhadores, ou equivalentes aos contentores vs navios na carga e descarga portuária, têm de ser fortemente aumentadas). Com a "popularização" do laser e os sistemas de localização baseados no laser, surgiu um novo impulso ao transporte autónomo pois o permitiu abandonar a questão do "fio". No entanto, também não foi com este importante avanço tecnológico que generalizou a utilização do transporte autónomo. Estas pequenas observações servem para ilustrar a ideia que no actual estado as situações em que a aplicação do transporte autónomo é um sucesso, ainda é limitado e muito específico e, mais do que um problema de tecnologia, é um problema de conceitos e mecanismos que desenvolvam um grau de autonomia permitindo ao sistema lidar com a imprecisão, o erro, a adaptabilidade mecânica e a sua utilização num ambiente heterogéneo

Apresentam-se como vantagens da utilização de sistemas automático de transporte, a diminuição do número de acidentes no ambiente industrial, (pois a aplicação das normas de segurança não dependem do operador mas da especificação de sistema), o cumprimento dos limites de velocidade para sistemas móveis, ou operação com ferramentas na presença de humanos. Outra vantagem que parece surgir consiste no menor desgaste que o material sofre por comparação com equivalentes operações por humanos.

Quando não é colocada a questão da versatilidade verifica-se um aumento da carga transportada por unidade de tempo.

Para além dos requisitos lógicos associados à definição de autonomia, é necessário caracterizar um conjunto de atributos necessários ao projecto de sistemas de transporte autónomo industrial. Assumem um papel importante os seguintes atributos:

carga por unidade de tempo envolve todo o processo e condiciona o valor da potência mínima do sistema, afecta directamente os sistema de tracção e de energia;

precisão na docagem vs limites para a imprecisão na localização das estações afecta o sistema de controlo de movimento, a selecção dos sistemas sensoriais a utilizar no sistema e o processamento sensorial;

autonomia energética afecta na definição de numero de unidades autónomas, necessárias para desempenhar um número determinado de unidades de tempo de serviço;

face às restrições geométricas do ambiente industrial, a definição da ***velocidade média requerida*** afecta o projecto do sistema de travagem e a caracterização das referências para o sistema de controlo do movimento.

2.3 Caracterização do ambiente

Para definir com mais rigor o objecto do nosso estudo, vamos no caso do transporte industrial autónomo omitir referências ao sistema de gestão da produção e assumir que o AMSIA funciona como prestador de serviços. Consequentemente os elementos fundamentais a caracterizar são os grupos funcionais do sistema: armazenamento de material pré e pós processado; estações de trabalho; e sistema de transporte.

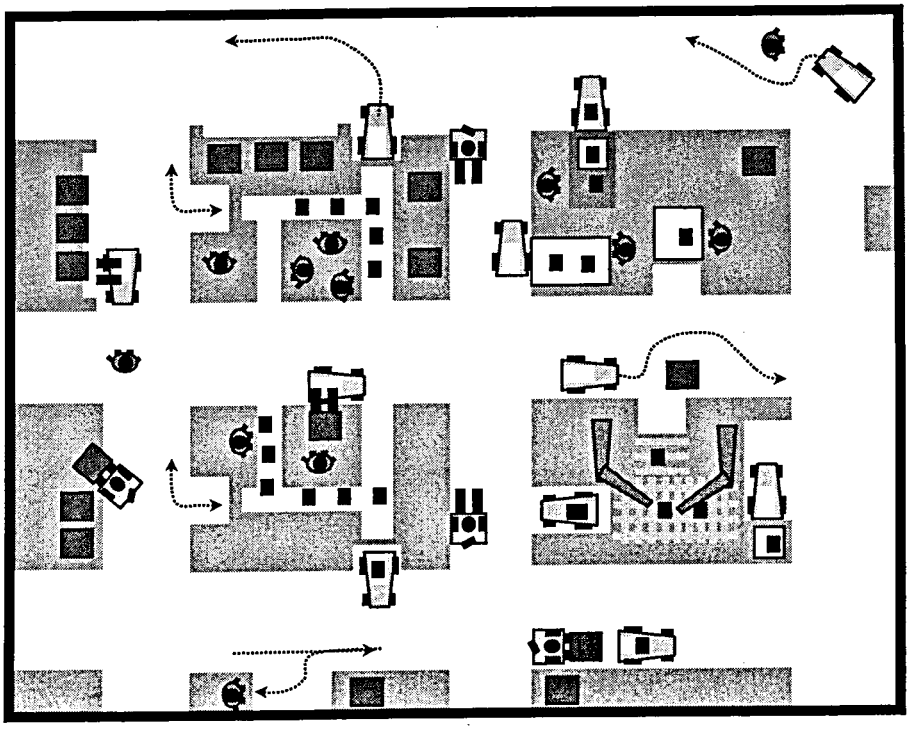
No limite da análise, podemos considerar que existe operações de carga e descarga, que necessitam de coordenar "esforços" com o sistema de locomoção, consequência do efeito dinâmico da movimentação do material.

As características das estações de trabalho e os armazéns são fundamentalmente dependentes da aplicação (da industria) e, se bem que as questões como a carga de materiais e o seu manuseamento sejam factores pertinentes para o estabelecimento da autonomia dada a sua particular natureza, vamos fazer a abstracção destas duas funções, sendo a fronteira para o nosso objecto de estudo o problema da "acostagem" às estações.

O ambiente industrial para o qual endereçamos a questão do transporte e operação autónoma caracteriza-se fisicamente (quanto ao movimento), por uma estrutura de circulações baseada em corredores, cruzamentos, áreas de estacionamento livre, e áreas de operação ou de "docagem" (padrões geométricos de restrição ao movimento). A dinâmica desta estrutura é lenta face à operacionalidade requerida, pelo que consideramos as restrições físicas impostas como estáticas. A estrutura pode ser alterada em consequência de um arranjo do "layout" produtivo. Contudo, será em torno dos padrões de restrição já mencionados. Além das restrições geométricas, os sistemas de transporte interactuam com outros agentes moveis tais como peões (humanos que participam no processo produtivo) e empilhadores (conduzidos por humanos).

Figura 2.3-1

A figura ilustra alguns aspectos do ambiente industrial, em que se podem observar alguns corredores: cruzamentos, curvas em "L" e diferentes tipo de docagem.



A situação proposta e ilustrada na figura anterior, leva-nos a facilmente a considerar a forte dependência entre as restrições geométricas impostas pelo ambiente e o problema associado ao movimento, pelo que vamos realizar uma

análise mais cuidada para o desenvolvimento de um modelo abstracto de restrições geométricas. No entanto esta análise orienta-se fundamentalmente para a especificação de uma estratégia de síntese do movimento do veículo, e não fundamenta a totalidade da analise para a especificação de uma estratégia de percepção. Pelo que, seguidamente iremos desenvolver uma analise que permita complementar o modelo de restrições ao movimento, quer através do aumento dos atributos associados aos elementos de modelação, ou através de novos elementos caracterizadores do ambiente que não estabelecem um relação directa com o movimento.

2.3.1 Classes de padrões de restrições geométricas

Como consequência da análise definimos as seguintes classes de padrões de restrições geométricas:

Classe de corredores.

Um corredor é um padrão de restrições geométricas caracterizada por possuir quatro lados, dois dos quais funcionam como ligações (iremos denotar como fronteira) a outros padrões, e os outros dois como limites físicos ao movimento. Na tabela seguinte podemos observar diferentes formas de corredor, sendo que o último representa a abstracção da classe corredor. Um corredor caracteriza-se igualmente pela linearidade associada ao desenvolvimento da restrição física.

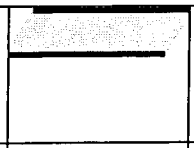
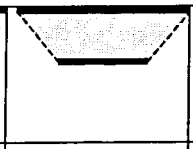
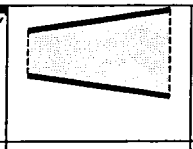
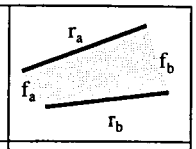
				
restrições físicas paralelas e ortogonais	Restrições físicas paralelas	só limites físicos ao movimento são paralelos	fronteiras paralelas, mas em que as restrições físicas evoluem de forma diferente	Padrão abstracto

Figura 2.3-2

Alguns exemplos da nossa definição de corredor.

Notar que a definição associada ao senso comum têm espectro mais largo.

As linhas fronteira (denotadas no desenho por ***f***) são as linhas definidas pelos limites dos segmentos de recta das restrições mais próximas.

Classe de Curvas

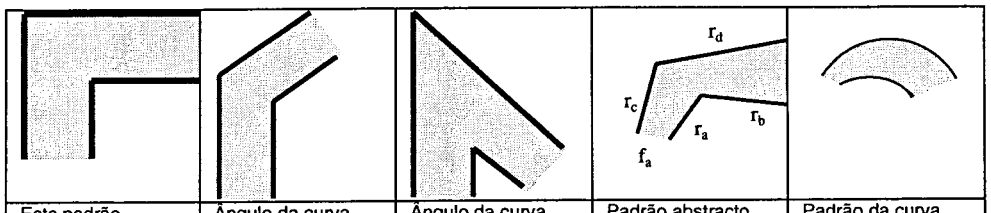
Fundamentalmente uma curva é um padrão de restrições geométricas que provoca um alteração significativa na direcção do movimento. Normalmente nos "layouts" industriais a curvas assumem a forma "L", em todo o caso também consideremos a situação em que a curva possui uma variação

de curvatura distribuída. A restrição geométrica tipo "L" é caracterizada por possuir seis elementos, sendo que dois funcionam como elementos de ligação a outros padrões de movimento e os restantes são limites físicos ao movimento. Claramente nesta restrição geométrica existe uma não linearidade associada à restrição física do movimento. No caso de uma curva com variação de curvatura distribuída, vamos caracterizá-la por dois arcos de circunferência. Na tabela seguinte podemos observar diferentes topologias de curva com variações do ângulo de inflexão, sendo que o quarto elemento apresenta a abstracção da classe curva do tipo "L".

Figura 2.3-3

Alguns exemplos da nossa definição de curva.

Notar que a definição associada ao senso comum têm espectro mais largo.



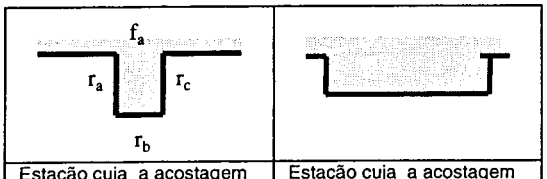
Este padrão representa a forma inicial do nosso problema, sendo os seguintes sucessivas generalizações.	Ângulo da curva superior a 90°. No exemplo curva à "direita"	Ângulo da curva inferior a 90°.	Padrão abstracto	Padrão da curva com curvatura distribuída

Classe de áreas de acostagem ou "docas".

A restrição geométrica denotada por estação de acostagem caracteriza-se por ser uma área de geométrica em que só possui um lado de ligação a outras áreas. Possui um caracter particular quanto à movimentação, pois é a área onde se vai desenvolver fase final do movimento de acostagem.

Figura 2.3-4

Alguns exemplos da nossa definição docas.



Estação cuja a acostagem se efectua no sentido longitudinal do veículo.	Estação cuja a acostagem se efectua no sentido transversal do veículo.

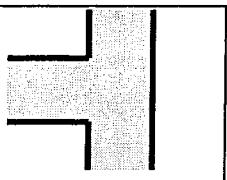
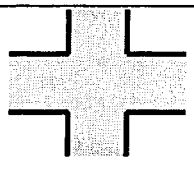
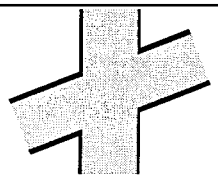
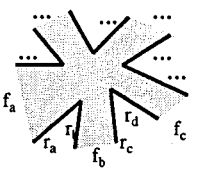
Classe de Cruzamentos

O cruzamento é a restrição geométrica que efectua a ligação entre três ou mais elementos dos padrões do tipo curvas, corredores ou estações de acostagem. Interpretamos um cruzamento como um elemento padrão em que, sobre o mesmo espaço físico em função das fronteiras dos elementos que confluem, se definem varias curvas e corredores. Cruzamento é pois um

padrão de restrição geométrica que associa padrões que partilham o mesmo espaço físico e lógico, e que estabelecem a ligação a mais que dois padrões.

Figura 2.3-5

Alguns exemplos da nossa definição de cruzamento.

			
Cruzamento composição de duas curvas 'L' e um corredor.	Cruzamento composição de quatro curvas 'L' e dois corredores.	Caso anterior mas com ângulos diferentes.	Padrão abstracto. Podem ser definidas varias curva e vários corredores.

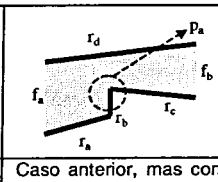
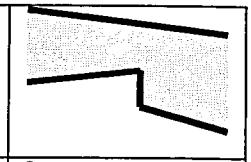
No caso dos cruzamentos as restrições ao movimento são de natureza lógica e dependem dos limites físicos das fronteiras.

Topologias particulares

São identificadas ainda outras formas de restrição geométrica ao movimento que dada a sua particular configuração sugerem que se adopte estratégias de movimentação mais cuidadas. Na tabela seguinte exemplificamos com a não simetria do movimento relativamente ao sentido, e com a definição de espaços que perturbam o movimento com descontinuidades modeláveis.

Figura 2.3-6

Notar que o ponto P é que define as topologias do movimento.

			
Padrão com características de um corredor, mas que possuí num dos sentidos um estreitamento.	Padrão com características de um corredor, mas que possuí num dos sentidos um alargamento.	Caso anterior, mas com uma forma genérica	Caso anterior, mas com uma forma genérica

A titulo exemplificativo podemos observar que na tabela anterior que o ponto P_a é um ponto crítico na definição do movimento.

2.3.2 Classes de elementos característicos associados ao ambiente

Para além dos padrões de restrições geométricas anteriormente apresentados existem outros elementos fundamentais na caracterização do ambiente industrial semi estruturado. Os padrões de restrições geométricas são elementos caracterizadores do plano geométrico da movimentação, e não

são suficientes para a caracterização topológica do ambiente. È necessário identificar outros elementos que permitam ao sistema identificar locais, quer duma forma concreta para utilizar na localização, quer de uma forma abstracta para usar na movimentação. Identificam-se como elementos "naturais" importantes, colunas, paredes, esquinas. Quando estes não possuem uma topologia adequada que permita um uso eficaz na navegação e localização é necessário enriquecer o ambiente com elementos caracterizadores artificiais, como marcas ópticas ou acústicas entre as quais incluímos, a pintura de paredes e pavimentos com cores ou padrões e alteração da textura de uma parede . Complementarmente, pode enriquecer-se o ambiente com marcas para a reflexão de sinais laser ou de Infra-vermelhos.

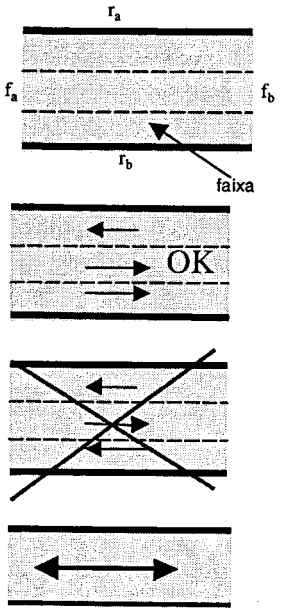
Estes elementos possuem um carácter passivo, e não definem totalmente o ambiente. Este possui normalmente elementos autónomos que estão largos períodos de tempo (ou sempre) fora do controlo do sistema de gestão, ou são controlados noutra dimensão do estado, pelo que o veículo autónomo deve relacionar-se pela capacidade perceptiva e não pela distribuição da informação por parte dos sistema central. Dessa forma, é necessário indentificar outros veículos, autónomos ou não, operadores humanos, caixas e "paletes" (não são autónomos, mas é melhor pensar que sim), e elementos artificiais para a coordenação de sistemas como semáforos.

2.4 Caracterização das relações do sistema autónomo com o ambiente

No problema do transporte autónomo os elementos que são partilhados autonomamente são os padrões de movimento. Consequentemente, quando é solicitado a um sistema que execute uma determinada tarefa, por desenvolvimento não determinístico do estado, não existe a garantia que a execução acompanhe a referência em "malha aberta". Pelo que o ambiente necessita, por um lado, de criar regras construtivas e regras de resolução de conflitos para que parte da coordenação do sistema ocorra de forma distribuída. Consequentemente, resultam algumas regras de movimentação quer para veículos autónomos quer para veículos conduzidos por humanos. Os principais vectores que regulam a relação entre o veículo autónomo e o ambiente no que diz respeito à circulação são:

- um padrão de restrições geométricas pode ter mais que uma faixa de circulação;
- os sentidos nas faixas são independentes. No entanto, em cada restrição só um dos lados de cada faixa possui um outro com circulação em sentido oposto;
- a movimentação realiza-se pelo lado direito das restrições geométricas e das faixas;
- em casos especiais uma faixa pode ser percorrido temporariamente em sentido de circulação oposta ao definido.
- Existem padrões de circulação que possuindo apenas uma faixa, esta pode ser percorrida em sentidos opostos, pelo que é necessário coordenar o acesso a este padrão.

Faixa é um limite virtual ao movimento e funciona como uma subestrutura da restrição geométrica.



Complementarmente, são definidas regras que estabelecem protocolos de comportamento entre sistemas autónomos e outros agentes móveis, incluindo sistemas autónomos.

Quando as actividades de dois agentes interferem de uma forma não cooperativa, o agente não autónomo têm prioridade para prosseguir com a mesma em detrimento do agente autónomo. Particularizando no caso de um cruzamento, o agente não autónomo têm, relativamente ao autónomo, sempre prioridade para o atravessar.

2.5 Caracterização das funcionalidades requeridas

Se bem que a maioria das funcionalidades de um sistema autónomo possam ser generalizáveis num conjunto vasto de aplicações, a definição do quadro completo é consequência directa da aplicação objecto de estudo. No presente exemplo o sistema autónomo é para ser utilizado no transporte de material entre as diversas estações de trabalho e, entre estas e as diferentes zonas de armazenagem. Neste contexto, iremos definir a autonomia (especificar o conjunto de funcionalidades às quais gostaríamos de responder) no que concerne à preparação para a transferência do material e no transporte entre a origem e o destino. Não cabe aqui a discussão do sistema de decisão de toda unidade industrial (aliás neste contexto funciona como restrição operativa). A caracterização neste ponto vai seguir uma filosofia "top-down" ,

pelo que vamos apresentar as funcionalidades do ponto de vista do "cliente" do recurso.

Então o porquê da sua utilização?, quais as vantagens? Ver ideias que justificam no ponto 2.1

Para caracterizarmos as funcionalidades que um sistema autónomo de transporte deve possuir, vamos considerar que este é "intermutável" em termos funcionais com o binómio "operador humano/sistema de transporte", fundamentalmente por ser nossa convicção que este será o passo seguinte na introdução dos sistemas de transporte autónomo num ambiente industrial.

Consequentemente, o "interface" entre o sistema de decisão da unidade industrial e o sistema autónomo possuí um carácter imperativo, fundamentalmente caracterizado por uma relação cliente/servidor.

Os "serviços" (básicos) que neste quadro o sistema autónomo de transporte disponibiliza são do tipo (tenham uma semântica equivalente):

- "carregar o material na estação A e descarrega-lo na estação C";
- "carregar o material no armazém C e levar até à estação de trabalho B";
- "carregar o material na estação C e descarregá-lo no armazém D";
- "accionar o mecanismo A";
- Quando não existe procura dos serviços deve estacionar ou melhora o nível energético.

Uma vez que estamos perante um sistema em que o único recurso que partilham é os padrões de restrição topológica. A cooperação ocorre através do estabelecimento das relações de ordem.

Esta caracterização é limitada face às capacidades que o binómio "operador humano/sistema de transporte" apresenta. Pelo que é necessário expandir as funcionalidades para que o desempenho dependa não só do sistema autónomo mas também dos comportamentos dos outros sistemas. Estas funcionalidades têm de permitir existência "coordenação/cooperação" com o outras entidades. Para que tal aconteça, a complexidade dos conceitos têm que crescer, e necessitamos de colocar na definição do serviço restrições de natureza temporal/espacial ou relação de ordem face a outros acontecimentos do sistema.

Notar que no perfil anterior era irrelevante, que o sistema autónomo "soubesse" que existiam outros sistemas autónomos. Neste é obrigatório o conhecimento da sua existência bem o como em algumas situações acompanhamento do seu comportamento.

Deste modo, um nível mais evoluído para o perfil de serviços que o nosso sistema pode ser caracterizado por:

"acciona o mecanismo C depois do sistema A accionar o mecanismo D"

"carrega material na estação D após o sistema A descarregar na estação C"

"carrega o material no armazém D e espera o accionamento B estar desligado para descarregar na estação C"

Estes exemplos já ilustram um quadro bastante complexo no que respeita a autonomia, no entanto, o senso comum de autonomia apela à necessidade deste quadro ser enriquecido com a noção de ciclo e a noção de condição. Desta forma, o perfil de serviços são enriquecidos com as seguintes formas:

> "enquanto o accionamento A estiver activo, carrega o material na estação A e descarrega na estação D"
>
> "carrega no armazém A, e descarrega na estação C, se esta estiver ocupada descarrega na A".

Esta pequena "discussão" mostra-nos que a autonomia do sistema (resulta da composição) desenvolve-se em dois vectores: um na vertente da relação física do sistema com o mundo envolvente que cresce com o aumento das "habilidades motoras" do sistema (com implicações directas nas capacidades sensoriais e actuadoras e nas estratégia de controlo - como lidar com obstáculos e a "falta" de espaço para a movimentação); um vector na vertente lógica (lidar com regras de circulação, a noção de frota) que cresce com o aumento das entidades sintácticas e semânticas e da relação entre elas. A título exemplificativo, podemos ilustrar um sistema que, em comparação com um outro, tem menores recursos motores pode exibir a mesma performance (relativamente à autonomia) caso possua um quadro lógico mais evoluído.

Deste modo, a caracterização envolve a determinação das funcionalidades do sistema autónomo para que a semântica da execução do sistema inclua a semântica associada ao serviço requerido.

No seguimento da nossa abordagem "top-down", vamos enumerar algumas das funcionalidades que caracterizam o sistema de transporte autónomo.

Agente no contexto deste texto, é qualquer entidade que pertença ao ambiente industrial e cuja a causalidade do comportamento é independente do sistema autónomo actor.

O sistema autónomo têm de possuir a capacidade de:

(i) qualquer ponto do espaço físico disponível para a circulação, deslocar-se até uma estação de trabalho ou armazém;

(ii) acostar às estações de trabalho de forma que a transferência seja possível;

(iii) transferir o material e accionar equipamentos;

(iv) identificar o estado dos outros agentes, com particular interesse a sua localização com diversos níveis de precisão;

(v) estabelecer comunicação com os outros agentes;

(vi) identificar os agentes do processo;

(vii) alterar o comportamento em função do estado de outros agentes (noção de frota);

(viii) identificar acontecimentos relevantes do sistema e ambiente; estabelecer comunicação com os outros agentes;

Os requisitos numerados anteriormente, vão formatar a especificação dos componentes do sistema e a especificação do seu comportamento, e vão contribuir em diversas fase do projecto de forma variada para a definição dos componentes físicos e lógicos.

No capítulo 3, revisitaremos os requisitos para sistemas autónomos de no contexto de um ambiente totalmente diferente – o meio subaquático.

Como este caso de estudo orienta a questão da autonomia para o movimento, vamos seguidamente realizar um aprofundamento na sua caracterização.

2.6 Caracterização dos tipos de movimento

Definamos que um movimento pode ser realizado em: progressão "forward" (significa marcha à frente); ou progressão "backward" (vulgar marcha a trás). Esta característica depende da forma como o veículo executa o movimento, e não do movimento em si.

O sentido não deve ser entendido no seu significado geométrico mas sim como uma ordenação, de e para que fronteira da restrição geométrica o veículo se dirige.

O movimento mais simples de identificar é o "***atravessar***" um padrão de restrições geométricas (estamos apenas a considerar corredores, curvas e cruzamentos). "***Atravessar***" define-se como: O veículo autónomo inicia e termina o movimento noutros padrões, ou seja, o veículo circula entre as fronteiras de transição restringido pelos limites físicos do padrão. É um caso particular quando o veículo inicia o movimento dentro do padrão. Este movimento pode ser feito de uma forma simples, mantendo a progressão e mantendo a faixa de rodagem, ou recorrendo a sequências mais complexas de movimentos (vulgarmente associamos a manobras), alterar a forma de progressão, mudar de faixa, ou realizar uma qualquer combinação entre a forma de progressão, a escolha da faixa de circulação.

Exemplos adicionais poderiam ser apresentados, mas generalizando a situação anterior, podemos englobar neste tipo de movimento o cenário em que o veículo por qualquer motivo (avaria, funciona mal, ou está sujeito a alterações de plano) vai iniciar a marcha de qualquer ponto do padrão e abandonar o padrão em movimento por qualquer fronteira.

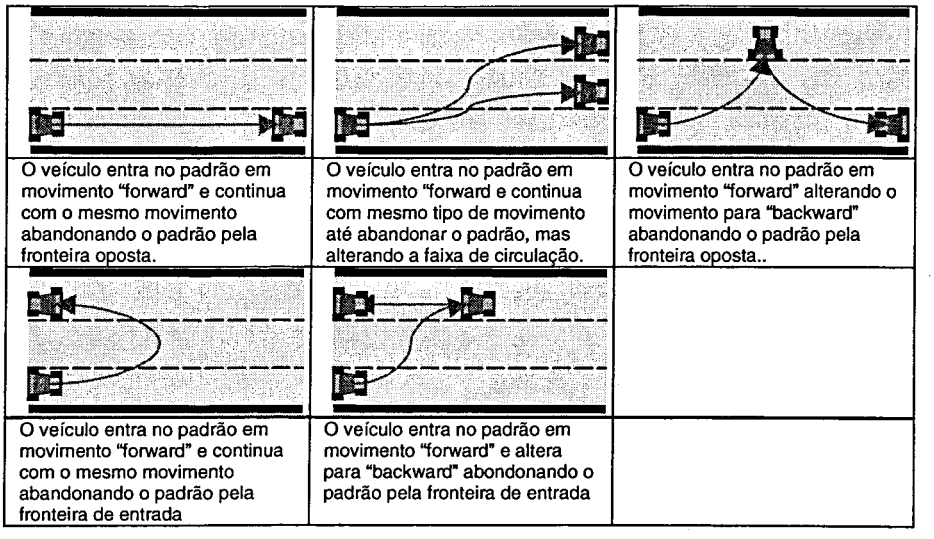


Figura 2.6-1

Exemplos de movimentos que caracterizam o "atravessar" um padrão

Se bem que os exemplos são ilustrados com corredores os conceitos não perdem validade para as curvas.

É de notar que, sendo o cruzamento num sentido abstracto como um conjunto das conexões que é possível estabelecer entre vários corredores, curvas ou estações (é uma entidade hierárquica que sobre o mesmo espaço físico define vários e diferentes padrões), conclui-se que, do ponto de vista do movimento, ***atravessar*** um cruzamento é o equivalente a ***atravessar*** um corredor, ou uma curva definida que está a fronteira de saída.

Outra característica dos movimentos associadas à ideia de ***atravessar*** não sugerem soluções de movimento muito precisas, mas sim um conceito de família de soluções. Dito de outra forma, o conceito de atravessar pode ser visto como um operador que,

> *Uma vez acolhido o veículo na restrição de entrada, coloca o veículo "fora" do padrão pela fronteira de saída determinada pela progressão pretendida, e em "condições" que viabilizam a possibilidade deste atravessar o próximo padrão ou "acostar" na estação destino.*

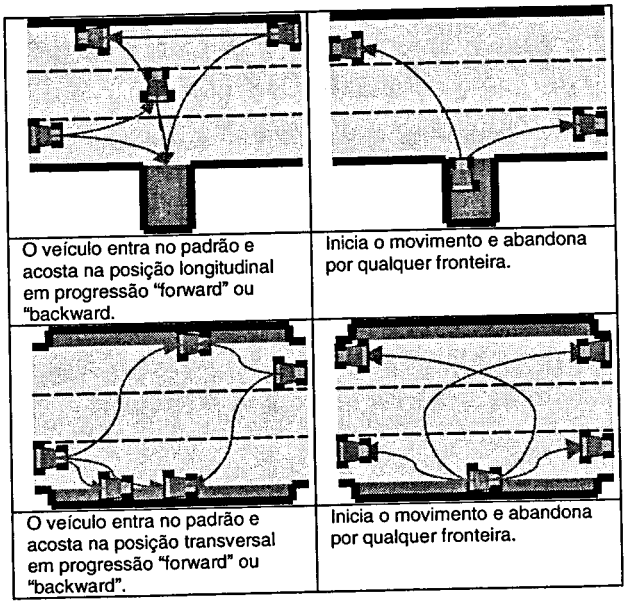
Outro tipo de movimento que se identifica é a "***Acostagem***".

"***Acostagem***" define-se como: "a sequência de movimentos que irá permitir ao veículo posicionar-se na estação de trabalho ou estação de armazenagem com velocidade nula". A complexidade deste tipo de movimento depende muito da aproximação e da forma como o veículo estaciona. Claramente que o último movimento - o que determina as condições de finalização - tem de permitir a precisão suficiente para preencher as condições de terminação. Enquanto que na situação anterior (***atravessar***), o desempenho pode ser medido essencialmente pela "simplicidade" com o que sistema ultrapassa o padrão geométrico, neste caso o factor da precisão afecta

fortemente a medida do desempenho. Por outro lado, o movimento inverso - a "***desacostagem***" - possui características comportamentais menos restritivas.

Figura 2.6-2

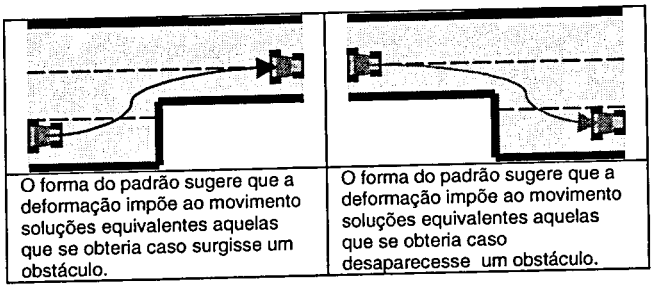
Exemplos de movimentos que caracterizam a acostagem.



Apesar de "***atravessar***" e "***acostar***" serem as duas grandes famílias de movimentos, importa atender cuidadosamente à forma como são formalizadas uma vez que a irregularidade dos espaços físicos poderão implicar implementações com alterações atípicas nas configurações. Chamamos atenção para os exemplos seguintes, em que as "deformações físicas permanentes" do padrão sugerem que esta "deformação" funciona como um "obstáculo" ao movimento.

Figura 2.6-3

Exemplos de movimentos que caracterizam a resolução de obstáculos.



Estes exemplos configuram a caracterização de movimento num padrão em que a forma deste se deve a uma das duas situações: irregularidades permanentes e temporárias.

A uniformização do tratamento destas duas situações, reflectidas no estabelecimento de padres únicos para ambas as situações, conduz a ignorar do ponto de vista funcional a irregularidade. Ou seja, a irregularidade é

transparente para o sistema de controlo, consequência da "regularização semântica": Isto poderia ser conseguido utilizando a "irregularidade" como um "landmark" intra-movimento para a necessária adaptação geométrica deste.

2.7 Caracterização dos tipos de movimentos em grupo

Até agora endereçamos o problema do transporte autónomo num ambiente industrial, focando a atenção para o problema do movimento do veículo. Mas a solução do problema de forma global também passa pela a definição de estratégias gestão do movimento da frota.

Se perspectivarmos que o sistema de transporte pode funcionar como um recurso a ser gerido de forma integrada, então a sua utilização deverá ter em conta índices de desempenho, tais como o número de elementos transportados por unidade de tempo, o tempo que cada estação demora a ser servida, etc.,,,. Noutro contexto pode ser perspectivada a utilização de formações para responder a requisitos especiais que exijam a cooperação física dos veículos.

As soluções poderão passar em estratégias baseadas de cada um por si, formação em comboio, gestão conjunto do movimento.

A gestão pode ser descentralizada, ou distribuída. A gestão distribuída ocorre quando o controlo deriva directamente de regras estruturais do ambiente ou "sociais". Estes mecanismos são implícitos ao veículo. A gestão descentralizada ocorre quando uma entidade supervisiona o comportamento dos veículos. Um problema que ocorre quando existe a coordenação de veículos, é se os sistemas dispõe de toda a informação, ou se as decisões que vão tomar é baseada em informação parcial. A gestão de conflitos é uma boa área de intervenção da coordenação de veículos.

2.8 Comentários à caracterização do movimento

A análise e caracterização dos pontos anteriores foi orientada de forma a considerarmos entidades com diversas granulemetrias, estabelecermos relações isomorfas entre as entidades hierarquicamente distintas. Consequentemente nossa preocupação fundamental de nessa análise se manter a consistência das hierarquias. Recorda-se o pormenor do tratamento da irregularidade dos padrões geométricos, (quer esta aconteça de forma

permanente ou temporária) resultando dessa forma os pormenores transparentes para a consideração da síntese do movimento. Esta forma de análise, permite isolar os diferentes patamares semânticos do sistema, simplificando os problemas pois diminui a complexidade do estado em cada nível. Como veremos mais tarde esta perspectiva de análise, irá por um lado estabelecer padrões (é sempre mais fácil lidar com padrões), tornar "finito" os problemas a resolver, isola-los e permitindo o endereçamento modulado dos mesmos.

2.9 Questões sobre a capacidade de decisão

Não confundir este mecanismo com reactividade.

O que se espera fundamentalmente de um sistema autónomo é que este seja capaz de decidir. Num sistema tempo real existem limites temporais para a decisão, logo esta não pode ser consequência de processos muito "complexos". O projecto dos componentes que intervêm na decisão deve privilegiar o desenvolvimento de sistemas em que a decisão resulta da interpretação do estado do ambiente e do veículo.

Um quadro de decisão assim facilita a racionalização sistémica

O exemplo referido no ponto anterior motiva a reflexão sobre a necessidade do sistema possuir percepção (complementarmente acção) em diferentes hierarquias, permitindo ao sistema lidar com objectos complexos de forma simples. O processo de percepção resulta da classificação da informação, e a hierarquia resulta do desenvolvimento relações de equivalência no espaço de estados. Consequentemente, o processo de avaliação, decisão e controlo em níveis hierárquicos superiores ocorre sobre relações (classes) de equivalência. É sempre mais fácil decidir num quadro lógico por oposição a um quadro quantitativo. Não é novidade que a hierarquia é um meio de gerir a complexidade, mas o que queremos salientar é que desta forma é possível ao sistema perspectivar a sequência de acontecimentos relevantes ao comportamento de uma forma temporal e espacial - as transições nas classes de equivalência de um dado nível, não são "visíveis" no nível mais baixo.

Outro vector a ter em conta em sistemas que se pretende autónomos, é que o sistema não decide e actua simplesmente em função do ambiente que envolve mas também em função do seu próprio estado. Neste caso, o sistema e ambiente "confundem-se".

Algumas questões que concorrem para a capacidade de decisão dos sistema no ambiente caracterizado anteriormente:

O sistema têm de ser capaz de identificar o que é um obstáculo ao movimento. Caracteriza-lo em termos de dimensão e dinâmica, responder às questões de: como se movimenta? é um veículo? onde vai estar nos próximos instantes ? parou?

O sistema têm de ter "conhecimento" das sua capacidades, do que sabe fazer, localizar-se (veremos em mais pormenor no ponto seguinte).

Relativamente a outras entidades ter noção que elas existem, responder às seguintes questões: Quem é esta entidade? Está no sítio que eu previa? Consigo determinar a sua posição com a precisão requerida? Que relações posso estabelecer com esta entidade? Em particular, no nosso caso de estudo, é um armazém ou uma estação de trabalho?

Noutro domínio de decisão, o sistema têm responder às seguintes questões: Estou apto a desempenhar as funções esperadas? Se a energia não é suficiente para cumprir a missão, preciso de aumentar o meu nível de energia, sei como fazê-lo? Se o sistema avaria, este percebe se esta é fundamental para o cumprimento do desempenho esperado? Já terminei a missão solicitada? Estou pronto a aceitar novas solicitações? Sei aquilo que posso oferecer?

2.10 Requisitos do movimento

Na caracterização que desenvolvemos sobre a movimentação do presente caso de estudo, verificamos que o movimento é sugerido por uma estrutura topológica (atravessar corredor, estacionar, etc ...), ou seja, simbólica. Esta caracterização sugere que a execução e o controlo do movimento tenha um correspondente nível simbólico. Por este nível, referimo-nos a acções que ocorrem em função do estado do sistema que, tipicamente, é caracterizado por uma semântica equivalente a: "entrei no corredor A", "saí do corredor A", "entrei na curva D". As referências ao sistema de controlo do mesmo nível serão do tipo: "atravessar para o corredor B", "atravessar o cruzamento para o Corredor C", "estacionar na estação F".

Esta caracterização define um quadro lógico que é independente do veículo e do seu estado, pelo que é necessário introduzir um quadro lógico que aproxime (vai transformando) as acções lógicas das acções físicas. Este

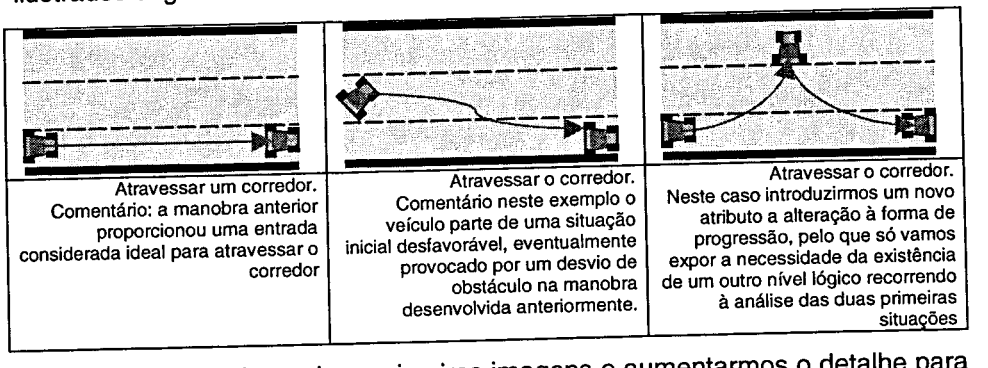
Por exemplo a questão do veículo ser ou não holonómico, diferencia o quadro dos requisitos do movimento.

quadro intermédio já vai ter em conta o veículo e as suas características, bem como a relação que este estabelece com o ambiente.

Analisemos um pequeno exemplo que ilustra a necessidade deste novo nível. No nível lógico com que descrevemos o movimento no ponto 2.6, a execução tinha de ter em conta as faixas de rodagem a restrições físicas e a regra "deve circular sobre a direita". Nesse nível a semântica dos exemplos ilustrados seguidamente é a mesma, ou seja, "***atravessar***".

Figura 2.10-1

Que requisitos para a organização do movimento?.



Atravessar um corredor. Comentário: a manobra anterior proporcionou uma entrada considerada ideal para atravessar o corredor	Atravessar o corredor. Comentário neste exemplo o veículo parte de uma situação inicial desfavorável, eventualmente provocado por um desvio de obstáculo na manobra desenvolvida anteriormente.	Atravessar o corredor. Neste caso introduzirmos um novo atributo a alteração à forma de progressão, pelo que só vamos expor a necessidade da existência de um outro nível lógico recorrendo à análise das duas primeiras situações

Observando as duas primeiras imagens e aumentarmos o detalhe para descrever o comportamento referido ao interior do padrão facilmente que os comportamentos não possuem qualitativamente a mesma semântica. Exemplifiquemos informalmente: Enquanto que, no primeiro cenário, a semântica se poderá eventualmente traduzir pela sequência "estou no corredor", "sigo faixa até ao fim do corredor seguindo afastado da parede 70 cm", "fim do corredor", no segundo caso, será "estou no corredor", "alinhar veículo com a faixa", "veículo alinhado com faixa", "sigo faixa até ao fim do corredor seguindo afastado da parede 70 cm", e "fim do corredor".

Reparar que, enquanto que na caracterização inicial do movimento, o veículo e a relação deste com o ambiente é perfeitamente transparente, nesta estabelece-se uma relação directa entre o veículo e o padrão, quer quanto à localização do veículo, quer quanto à relação deste com a referência. Este quadro ainda se apresenta num nível lógico pois a semântica pode ser aplicada a qualquer padrão de restrições geométricas.

Verifica-se pois necessidade do sistema ser capaz de observar acontecimentos a diversos níveis semânticos tais como "acabei de sair do corredor B", ou "passei a estar alinhado com a restrição à minha direita". São os acontecimentos reveladores desta semântica que vão ser os elementos de realimentação da execução do sistema e a referência concreta que o veículo possuí do padrão.

Definido um quadro lógico em que a relação do veículo e o ambiente está caracterizada, é necessário determinar para cada família de padrões geométricos as estratégias de controlo estáveis, cuja a composição resulta a manobra pretendida. Podemos enumerar uma série estratégias de controlo do tipo:

- evoluir numa dada direcção,
- evoluir seguindo uma recta (notar que é diferente da anterior),
- evoluir realizando uma curva com curvatura contínua,
- contornar um obstáculo pela direita,
- fazer convergir para uma recta o movimento de um dos pontos do veículo.

Notar que estas leis de controlo já possuem uma semântica física e requerem uma realimentação contínua.

A granulametria funcional do movimento fica assim definida em três níveis:

- o primeiro que estabelece que as linhas para a movimentação é apenas função dos padrões de restrições e de outras relações lógicas que se estabelecem com o ambiente,
- o segundo que define as relações lógicos do veículo face aos padrões, sugerindo portanto a composição de manobras,
- o terceiro que estabelece as leis que regulam a relação física do veículo face aos padrões e referências.

2.11 Requisitos Associados à Percepção. Localização e Navegação

A caracterização que fizemos do ambiente, do veículo e do perfil de comportamento esperado, sugere que o quadro de decisão ocorra é de múltiplos níveis e, consequentemente, temos de endereçar os requisitos que se colocam à percepção tendo em vista a execução da missão

Analisando o esquema semântico associado ao ambiente, constatamos uma variedade de planos semânticos. Para "acompanhar" este quadro o sistema autónomo necessita de uma base alargada de percepções que se obtêm através dos diferentes elementos produzidos pelo processamento sensorial.

Da mesma forma que classificamos a estrutura associada aos Padrões de restrição ao movimento, temos que classificar os espectros sensoriais. Criando uma estrutura hierárquica de "landmark's".

A base de percepções desenvolve-se nos diferentes níveis da hierarquia e estes devem emergir em adequada dependência dos cenários de actuação. Importa que o sistema de percepção esteja dotado dos seguintes mecanismos:

- capacidade de acumular informação de forma integrada para construir percepções que, na execução da missão de acordo com os requisitos desta, permita cobrir a base de percepções da missão em questão.
- orientar o esforço de recolha de dados sensoriais e respectivo processamento por forma a ir de encontro às percepções esperadas conforme o plano da missão ou esperáveis de acordo com a caracterização ambiental.

O foco da percepção é dirigido à observação da dinâmica do sistema e do mundo, nomeadamente no que respeita à problemática do movimento. Esta obriga a uma definição das relações entre o sistema e o ambiente. Algumas dessas relações passam por entidades que normalmente associamos à localização e/ou navegação, assumindo características relativas ou absolutas e simbólicas ou numéricas

Da mesma forma que a estruturação do espaço livre permite a definição de movimento caracterizado topologicamente, os "landmarks" são a estruturação do complemento ao espaço livre e permitem uma definição topológica da navegação. Da mesma forma que definimos formas de movimento simbólico, também encontramos formas de localização e navegação simbólica.

Seguidamente podemos observar nas duas tabelas seguintes um pequena exercício de organização dos atributos a considerar como fundamentais para a localização e navegação do sistema.

Tabela 2.11-1

Elementos associados à localização

Localização	***Relativa***	***absoluta***
Numérica	distancia e direcção a um objecto ou restrição	x y θ
Simbólica	dentro / for a direita / esquerda próximo / afastado alinhado / não alinhado inicio/fim frente / traz	sala divisão corredor cruzamento "doca"

Assume particular importância a precisão com que estes atributos são calculados. Essa precisão não é constante para cada atributo mas é dependente da missão e do contexto onde é usado.

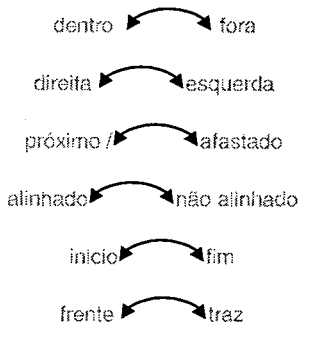
Navegação	***Relativa***	***absoluta***
Numérica	taxa de aproximação a um objecto ou restrição	x' y' θ'
Simbólica		no exemplo do ambiente industrial podemos definir como a restrição geométrica (divisão, corredor, cruzamento, "doca") para a qual é mais "económico" transitar. ou seja, com o movimento corrente qual é a restrição geométrica com mais probabilidade de ser atingida. è uma avaliação que depende da proximidade e do movimento activo

Tabela 2.11-2

Elementos associados à navegação

2.12 Requisitos inerentes a restrições tecnológicas

A solução de projecto de um sistema autónomo encontra um forte condicionalismo que é interpretado pelas restrições tecnológicas. Quando mais o processo de projecto se aproxima da implementação, mais estas se fazem notar no mesmo. Os objectivos da aplicação e as características do ambiente onde esta se vai desenvolver começam por definir o domínio de actuação do sistema. O projecto necessita de realizar o "rendez-vous" entre as necessidades conceptuais e os elementos tecnológicos que é passível de utilizar.

As linhas do "rendez-vouz" são:

Aspectos mecânicos associados ao projecto da estrutura física, fortemente dependentes da aplicação onde incluímos o esquema de tracção e outros componente de actuação no ambiente. Afecta directamente os modelos dinâmicos e cinemáticos do projecto.

- Elementos sensoriais de: caracter "externo" associados à localização/navegação do veículo, e à percepção das alterações que ocorrem no mundo; de carácter interno associados à percepção endógena do comportamento do sistema, à detecção de falhas e à

alocação de recursos. È necessário especificar um mapa das grandezas físicas ou lógicas associadas a cada sensor, definir como interpretamos as grandezas medidas e como as utilizamos no esquema de percepção do sistema.

- Elementos associados ao suporte computacional em que é necessário equacionar o poder de cálculo e as comunicações. São os sistemas computacionais que vão acolher a maioria dos componentes desenvolvidos no projecto e onde ocorrem a maioria das comunicações. O sistema operativo e os mecanismos de controlo associados afectam fortemente a estrutura dos processos a desenvolver pois os sistemas computacionais transformam grandezas físicas em lógicas. O problema principal ocorre com a grandeza "tempo" (relógio), pois por condicionalismo que veremos mais tarde que o crescimento a monótono desta grandeza na realidade física, por vezes não corresponde a um crescimento monótono na realidade lógica.
- Elementos associados ao fornecimento e acumulação de energia, bem como a sua distribuição aos subsistemas. As fontes energéticas e as necessidades energéticas são um factor importante na caracterização da autonomia.

A título exemplificativo caracterizamos seguidamente alguns dos elementos tecnológicos associados ao veículo autónomo do projecto Po-Robot [Po-Robot 99a].

COMPONENTES DO SISTEMA FÍSICO

Actuadores

Os sistemas de actuação do veículo são:

(i) dois motores da propulsão responsáveis pela movimentação do veículo configurado para locomoção diferencial;

(i) um freio electromagnético associado a cada motor;

(ii) um motor do tapete transportador usado para carregar e descarregar o material na estação do trabalho;

(iii) quatro motores passo-a-passo utilizados para rodar quatro pequenas torres equipadas com os sensores ultra-sónicos de TOF e sensores IRs;

(iv) três interruptores de potência para ligar/desligar alguns subsistemas;

Sensores

No que diz respeito aos sistemas sensoriais, a plataforma está equipada com:

(i) dois codificadores ópticos montados no eixo de cada motor para medir a velocidade/posição dos motores;

(i) sensores ultra-sónicos de tempo-de-vôo usados para determinar as distâncias aos objectos próximos da plataforma;

(ii) sensores IR para desempenhar o papel de pára-choques ópticos;

(iii) uma bússola magnética para medir a orientação relativamente ao Norte magnético;

(iv) três acelerómetros tri-axiais para monitorar o movimento do veículo;

(v) dois pára-choques, dianteiro e traseiro, para detectar colisões com objectos do mundo;

(vi) cinco botões para seleccionar o destino da plataforma e resolver situações de emergência;

(vii) um "joystick" para conduzir directamente o veículo em situações de mau funcionamento ou de emergência;

(viii) uma estação do telecontrolo composta por: volante para orientação da movimentação, um acelerador e um travão; e

(ix) um sistema de navegação laser.

Sistema Computacional

O sistema computacional é distribuído por todos os dispositivos da plataforma. Todos os sensores e actuadores são encapsulados com processos lógicos que funcionam localmente em relação ao sensor ou actuador. Deste modo, poupamos o sistema computacional principal para os processos que executam as funções de nível mais elevado. Actualmente, o sistema computacional é composto por:

(i) Um processador central PEP VM62 com um MC68060 que suporta o sistema operativo VxWorks onde são executados os módulos funcionais e os processos lógicos da coordenação;

(ii) Um cartão de input/output que realiza o interface entre os dispositivos físicos e os processos que implementam as algumas actividades funcionais elementares; e

(iii) uma rede sensorial ExMIO-multi-input/output, com a respectiva carta para ligar a rede ao CPU via memória partilhada. Os dispositivos são ligados à rede através de um pequeno cartão que realiza o interface entre o sensor e a rede.

Figura 2.12-1

Configuração básica de hardware do sistema Po-robot.

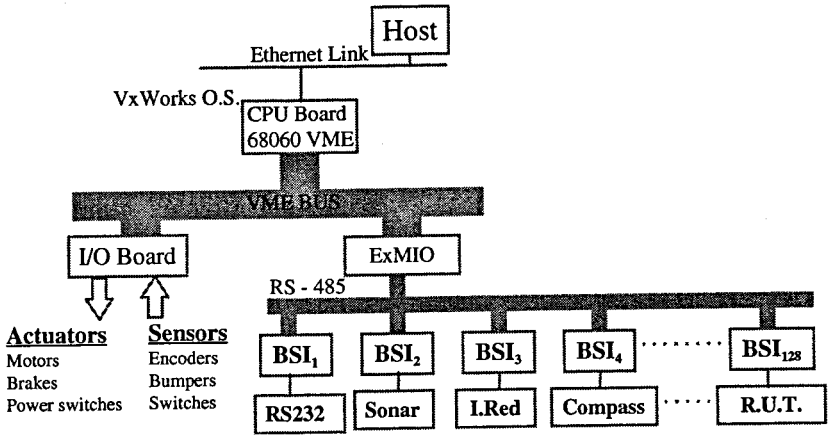
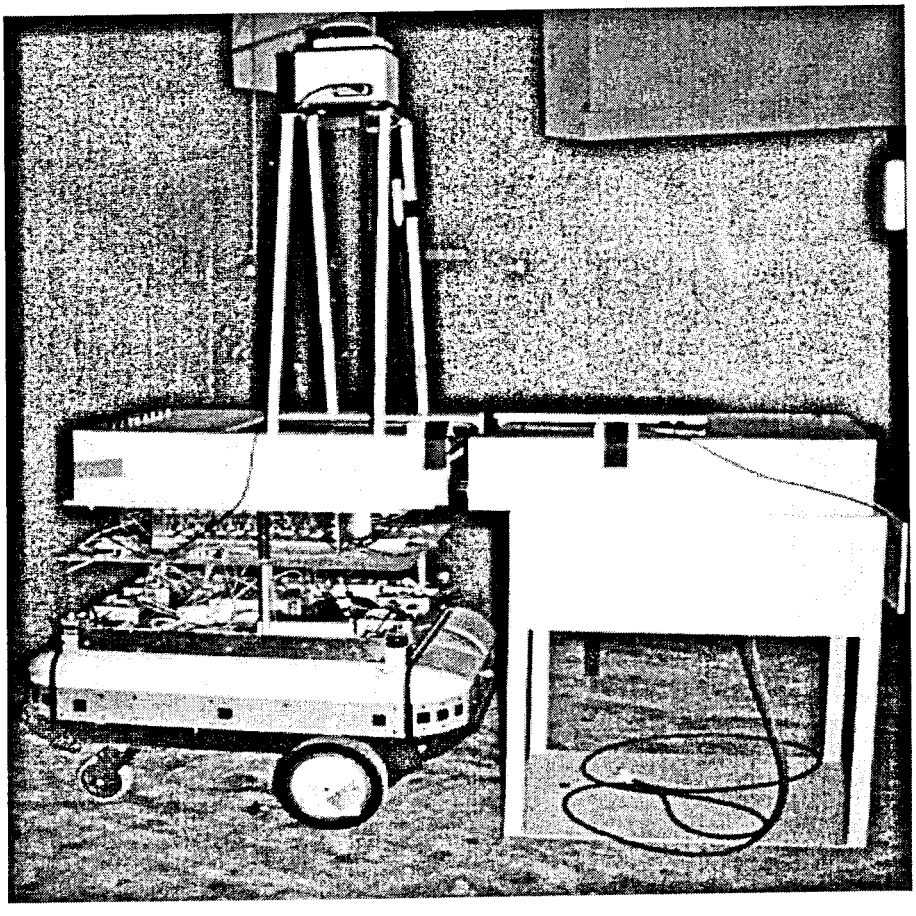


Figura 2.12-2

Configuração actual do Sistema Po-Robot



Sistema de potência

O sistema de potência é baseado em quatro baterias de tracção, fornecendo cada bateria 12 volts a 60Ah. Dependendo das características da missão, o sistema de potência permite de duas a seis horas de autonomia.

2.13 Especificação da missão

Num ambiente industrial, a missão tanto pode ser especificada de um forma procedimental como declarativa.

A forma procedimental de especificar pré-define a execução, ou seja, é um processo de caracterizar de uma forma explicita a lógica da sequência das tarefas. O carácter imperativo da relação cliente/servidor, ajusta-se facilmente à especificação na forma procedimental

Com o aumento da capacidade da autonomia a especificação das missões começam a ganhar um carácter declarativo pois a missão é definida por um objectivo que diríamos global ou, de outra maneira, a missão é especificada por uma propriedade (ou conjunto de propriedades) que pretendemos que seja uma característica do sistema.

3 CARACTERIZAÇÃO DE AMBIENTES HÍDRICOS PARA A MOVIMENTAÇÃO SUBMARINA AUTÓNOMA

3.1 Introdução

A necessidade premente de garantir que as consequências da actividade humana não conduzem a alterações com impacto negativo na sustentabilidade da vida na terra, leva por um lado à necessidade de um aumento do conhecimento do meio onde esta ocorre, e paralelamente à continua observação do efeito dessa mesma actividade. Os sistemas hídricos pela sua dimensão e importância na estrutura do desenvolvimento humano e sustentabilidade da terra têm sido objecto de monitorização e estudo ao longo das últimas décadas. No entanto, o conhecimento que temos destes sistemas têm demonstrado não ser suficiente, principalmente [Manheim 01] devido a sua dimensão e à dificuldade técnica de realizar o seu estudo quando o comparamos com a execução de estudos similares realizados em zonas terrestres. O efeito da mecânica dos rios e estuários já é bem conhecida. No entanto, os efeitos do aumento dos nutrientes e fosfatos (por exemplo) que esses rios transportam ainda carecem de estudos profundos e sistemáticos. Ainda são bastante preliminares e recentes os estudos sobre as descargas submarinas de água doce nos oceanos [Report 00], [Manheim 01], [Smith 00]. Estes estudos são urgentes, pois se, por um lado, são fundamentais para o total entendimento da interface nas zonas costeiras entre água doce/água salgada, por outro lado adquirem particular relevância para a inventariação e preservação das fontes de água doce da terra. Com o aumento exponencial da população na Terra, torna-se cada vez mais importante preservar as reservas de água doce e desenvolver soluções alternativas para a sua captação.

O quadro com a complexidade descrita carece de um aprofundamento do estudo, do controlo activo e da exploração dos fenómenos que se desenvolvem nesta zona tão importante do globo. No entanto, os custos económicos associados ao incremento da actividade de observação baseada nas tecnologias actualmente usadas tornam-se incomportáveis. Deste modo, é fundamental promover os desenvolvimentos no sentido de determinar uma solução sistémica que permita de forma ligeira e a custos reduzidos realizar missões de detecção e de observação de fenómenos dos meios subaquáticos, ou nas estruturas confinantes.

3.2 O problema

O grupo de investigação do LSTS têm investido grande parte das suas capacidades no desenvolvimento de veículos para operar em ambientes hídricos e que permitam o estudo e observação de fenómenos ambientais. Actualmente o grupo desenvolve e opera com dois veículos, o Veículo Submarino Autónomo (VSA) Isurus [Cruz et al. 99] e um Veículo Operado Remotamente (ROV) [Martins 99]. Consequência dos bons resultados obtidos, já se encontra em desenvolvimento um terceiro veículo. O exemplo que aqui vamos abordar centra-se no VSA Isurus e na concepção de sistemas de monitorização baseados em plataformas móveis em ambientes não estruturados.

A introdução de veículos móveis no processo de observação e monitorização ambiental [Ramos 00], [Fletcher 00] vão permitir a massificação da observação (pois os custos de operar com um VSA é manifestamente mais reduzido), o acesso a nova informação que antes não era "possível" recolher, a recolha paralela de informação temporalmente relacionada permitindo o estudo de fenómenos até antes dificilmente abordados e, mais importante ainda, a implementação do conceito da observação activa. Nas metodologias de observação classicamente usadas o tratamento e uso da informação recolhida é realizado posteriormente à missão. Todavia, a dinâmica de alguns fenómenos de interesse, não é compatível com este processo lento e "pesado" (na pesquisa de fontes de fenómenos oceanográficos de grande relevo).

Sem querer desenvolver metodologias de pesquisa subaquática, vamos ilustrar com pequenos exemplos as potencialidades que estes veículos possuem e as mais valias que podem introduzir nas metodologias de observação. Até agora a maioria dos processos de monitorização recorrem ao uso de embarcações oceanográficas na recolha dos dados, com custos de operação elevados (ver [Report 00], [Smith 01]). Normalmente, a resolução espacial desta recolha de informação é grosseira, chegando muitas vezes muito aquém dos requisitos necessários. A recolha simultânea e sincronizada de informação é praticamente impossível pelos custos envolvidos. Em águas pouco profundas, o estudo fica por vezes comprometido pela dificuldade do acesso das embarcações oceanográficas, perdendo-se informação fundamental para o desenvolvimento da pesquisa. Nas zonas costeiras a utilização de veículos móveis autónomos de pequena dimensão poderá ser a solução para os estudos que necessitam de uma resolução espacial muito

pequena e de informação temporalmente relacionada. Basta pensar que um VSA como o ISURUS pode operar facilmente com dois a quatros investigadores, utilizando uma embarcação semi-rígida de pequena dimensão, ou mesmo a partir de terra.

Para além das vantagens de natureza operacional (acesso a zonas com profundidades baixas ou de relevo complexo), facilmente se compreende que o custo de multiplicar as equipas de operação é baixo, quando comparado com os de multiplicar embarcações oceanográficas. Portanto, a utilização de VSA's pode

- diminuir significativamente o período de obtenção de informação através da paralelização e coordenação das actividades de recolha de dados, e
- aumentar significativamente o impacto de um conjunto de dados pelo incremento da resolução espacial e temporal, bem como pela possível correlação da informação.

O estudo da influência das descargas de água doce no oceano [Report 00], [Smith 00], [Swarzenski 02], [Franzemeier 00], [Charrete 01] ainda se encontra num patamar insatisfatório. Enquanto que o interface "água doce/água salgada" na zona de influência das descargas dos rios e das zonas estuarinas está razoavelmente bem conhecida, ainda é bastante difícil avaliar o impacto das descargas directas de água doce quer por "fugas" dos lençóis freáticos na zona de interface terra-oceano, quer através de fontes que descarregam directamente desses lençóis.

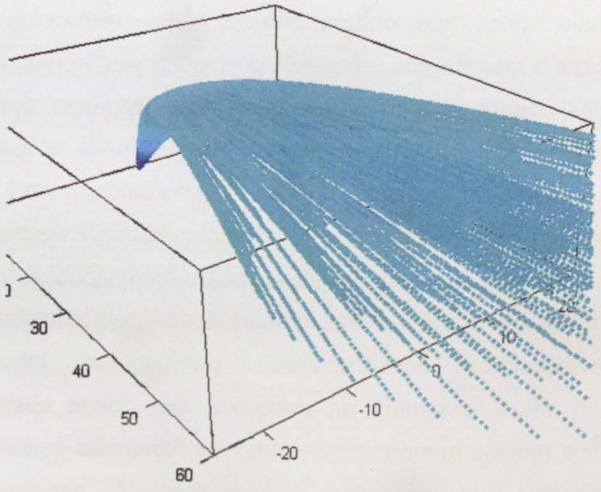


Figura 3.2-1

Forma aproximada da difusão de uma fonte de água doce numa massa de água salgada

É nossa convicção de que o desenvolvimento de metodologias de pesquisa que utilizem recursos como o VSA Isurus parece ser muito promissor para a pesquisa activa de fontes de descarga submarina de água doce no oceano [Fletcher 00], [Smith 00]. As metodologias de pesquisa podem desenvolver-se ao longo de duas linhas complementares. A primeira endereça o estudo da influência das descargas (e aqui estamos a falar de todo o tipo, rios, exutores ou fontes) e o peso da sua importância na biodiversidade. A segunda trata a identificação do local e caudal das fontes submarinas de água doce. A identificação do local e do caudal, são essenciais para a caracterização deste fenómeno, e por consequência, para a avaliação do potencial de exploração deste importante recurso. Até ao presente, o estudo destas fontes de água baseia-se fundamentalmente na recolha de informação em que se dá particular atenção à resistividade da água [Report 00]. Tipicamente, são realizadas diversas leituras a diferentes profundidades e em diferentes locais e, através da correlação (posterior, após a missão off-line) desta informação com outra como as correntes, a temperatura e oxigénio dissolvido, se tenta identificar e caracterizar a pluma. Como veremos adiante, a introdução de VSAs poderá alterar por completo o quadro das metodologias clássicas. Estas fontes são caracterizadas pela sua baixa salinidade e pela riqueza em nutrientes.

3.3 Caracterização do ambiente

Um dos pontos fundamentais na caracterização destes cenários de operação é o aparecimento de uma nova dimensão – a profundidade - relativamente ao movimento no ambiente industrial. Por esta razão, características universais como a força da gravidade ou pressão a que um corpo é sujeito assumem um papel importante no relacionamento do veículo com o ambiente.

Ao analisarmos o ambiente dos cenários possíveis de actuação, verificamos que estes vão desde quadros totalmente não estruturados, ou seja desprovidos de estruturas artificiais criadas pelo homem, até ambientes que confinam ou acolhem essas estruturas passíveis de tipificação e que marcam fortemente o cenário de operação.

Na primeira situação o carácter não estruturado do ambiente (rios e oceanos) onde propomos intervir resulta da imprevisibilidade da dinâmica dos

elementos que o caracterizam na vizinhança de qualquer configuração do veículo. Consequentemente, ao contrário do que ocorreu nos cenários do ambiente industrial, não nos é possível caracterizar padrões de movimento com base apenas nas características físicas de elementos espaciais. Mesmo que, por paralelismo, tentássemos definir características constantes numa determinada variável (equivalente ao facto de uma parede estar sempre presente), é extremamente difícil, senão impossível, extrair uma caracterização completa para todos os padrões de movimento.

Pode-se equacionar a caracterização do ambiente como um volume de água, limitado pela superfície, pelo fundo irregular e por obstáculos laterais de origem natural (montes submarinos, rochedos, etc.) ou artificiais (paredões, exutores, cais, etc.).

Dada a natureza irregular e não estruturada dos elementos constituintes dos ambientes subaquáticos (e.g., morfologia do fundo) não intervencionados pelo homem, não é geralmente possível utilizar este elemento do ambiente como factor de caracterização precisa, ou seja podemos medir a distância ao fundo mas não podemos daí inferir características regulares muito rigorosas. As características morfológicas só poderão ser utilizadas para efeitos de navegação em áreas bem delimitadas e com fundos particularmente bem definidos ou ainda em zonas intervencionadas tais como portos ou barragens. Em zonas "abertas" onde o fluxo do campo magnético se mantêm constante, é possível utilizar esta grandeza física para determinação da orientação dos veículos. Uma vez que a movimentação submarina que estamos a considerar se faz relativamente à massa de água (e não ao solo ou a referências submarinos ou terrestres), é necessário ter em conta as correntes submarinas. (Ao contrario da movimentação terrestre em que a informação obtida pela força da gravidade não era muito importante, esta característica em contraponto à força de impulsão vai ser importante na navegação submarina).

A forma como o homem é capaz de referenciar ou indexar fisicamente elementos de forma inequívoca passa pelo o que nós vulgarmente chamamos a posição e atitude (orientação no espaço) relativamente a um referencial "absoluto".

Porque será que em diversas situações os mergulhadores perdem a noção de verticalidade, e consequentemente a sua orientação global?

A ausência de referências de carácter espacial é no entanto compensada com outros elementos. Um desses elementos são as correntes que possuem um carácter mais ou menos estável do qual é possível conhecer o perfil. Se, associado a este facto, se constatar a existência de um campo magnético pouco perturbado, poder-se-á, por conjugação, obter informação estruturante.

Consideremos somente as correntes de baixa frequência, perturbações em torno destas podem ser vistas como ruído.

Daí resulta a possibilidade de realizar "posicionamentos" e consequentemente desenvolver uma navegação relativamente a elementos não espaciais. No cenário das fontes de água doce que ilustramos anteriormente, o modelo da difusão é previsível pelo que este facto poderá também ser utilizado na "navegação" na pesquisa da fonte.

3.4 Especificação de necessidades de desempenho do sistema

Uma das grandes vantagens que se espera obter com a utilização de veículos autónomos de pequena dimensão face às técnicas actuais de recolha de dados no meio subaquático e, em particular o oceânico, consiste na flexibilidade operacional que permite uma frequência elevada das operações de recolha e num considerável aumento da qualidade dos dados, não só em exactidão, mas na correcta e precisa caracterização do local e instante em que a informação foi recolhida.

A análise do cenário anteriormente caracterizado indicia o desenvolvimento de três classes de missões:

Visto de outra forma, a execução de cada um dos veículos, e a informação recolhida, não afecta o desempenho presente, nem futuro no decorrer da presente execução.

A primeira classe resulta num "survey" num dado volume de água [Cruz 99] no qual interessa recolher informação por um ou mais veículos, sendo a actividade de cada um deles definida apriori. Esta classe de missões pode ser desenhada tanto para um como para vários veículos. No caso da utilização de vários veículos, a operação de cada um deles poderá ser definida totalmente na fase de preparação da missão resultando a articulação das actividades do planeamento "off-line".

A segunda classe de missões é caracterizada pelo objectivo de pesquisa de um determinado fenómeno que só se manifesta de forma local (é o caso da deteção de minas militares) relativamente ao conjunto de sensores que o veículo possa transportar. Ou seja, dados recolhidos na vizinhança de um ponto não são suficientes para a detecção do fenómeno. A diferença desta missão para um "survey", é que neste caso existe a possibilidade de esta terminar sem ser necessário patrulhar todo o "volume" de água endereçado.

Na terceira classe de missões, a pesquisa da origem do fenómeno pode ser orientada recorrendo à interpretação de indícios que revelam a

influência do mesmo, ou seja, é possível orientarmos a pesquisa sem a observação directa do mesmo. Neste último caso, dificilmente se poderá desenhar uma missão de forma eficaz recorrendo simplesmente a um veículo. Mesmo recorrendo a vários veículos, é muito difícil levar a cabo a missão de forma eficiente sem que exista alguma coordenação da acção dos diversos veículos.

Sistematizando e organizando os principais requisitos gerais colocados subjacentes às missões a desempenhar no contexto da aplicação considerada, tem-se:

- "Setup" de operação reduzido;
- Capacidade da movimentação autónoma, com elevado grau de precisão e repetibilidade;
- Capacidade para armazenar informação, e a respectiva indexação espacial e temporal;
- Capacidade de estabelecer e comunicar entre veículos e o sistema;
- Interface de operação simples e de "alto nível";
- Vários tipos de dispositivos sensoriais em consequência da variedade de fenómenos a caracterizar;
- Capacidade de coordenação das actividades dos veículos.

3.5 Estruturar o meio e o movimento

A dificuldade de definir referências espaciais no ambiente submarino é um obstáculo à referenciação espacial e temporal da informação recolhida. Consequentemente, torna-se necessário intervir "artificialmente" no cenário de operação para que seja possível ao sistema responder às especificações solicitadas. Normalmente, recorre-se a um sistema de posicionamento exterior. O sistema de navegação "Long Base Line" (LBL) assenta na definição da posição por triangulação uma vez conhecida a distância aos vértices de um simplex. Uma vez conhecida a velocidade de propagação do som da água, estas distâncias são determinadas medindo o tempo de resposta a cada um de um conjunto de faróis acústicos (no mínimo dois) posicionados nos vértices de um simplex escolhido por forma a definir a região de operação dos veículos. A informação que se obtém deste sistema é complementada com a que se obtêm dos sensores de "dead-reckoning", de profundidade e dos sensores inerciais.

Contudo, a interacção dos veículos com os faróis acústicos tem custos energéticos consideráveis, consequência da energia necessária à emissão das ondas acústicas que sofrem forte atenuação na transmissão na água, sendo necessário equacionar o autonomia dos veículos versus configuração da rede de faróis acústicos e a interacção com esta. No que respeita ao consumo de energia, é consideravelmente diferente operar com faróis colocados a uma distância média de 1000 metros ou de 300 metros. Facilmente se compreende que, para uma mesma área, ou se decide no estabelecimento de uma rede rica em faróis estrategicamente bem colocados resultando num baixo consumo energético, ou se decide por um "setup" de operação mais leve colocando pelo menos dois faróis e consumindo bastante energia na interacção com estes. Complementarmente, o sistema de navegação deverá limitar a taxa de utilização da informação dos faróis desde que recorrendo a outros métodos mantenham a qualidade da localização no nível especificado. Por outro lado, as leis de controlo deverão ser capazes de limitar a propagação de erros. Como veremos mais adiante a solução também poderá passar por desenvolver mecanismos e estratégias de controlo que lidam com a incerteza mais facilmente. Algumas questões que se levantam para a navegação e o controlo podem ser endereçadas recorrendo a estratégias de grupo.

Para as duas primeiras classes de missões definidas no ponto anterior (survey, e pesquisa local de fenómenos), em que o resultado das grandezas monitorizadas não afectam o progresso da missão, a abordagem que propomos resulta numa actuação lógica no volume de água onde são geradas "paredes" virtuais. Mais concretamente, são considerados corredores de movimentação virtuais muito semelhantes aos corredores reais definidos para a movimentação autónoma industrial. Neste cenário, é necessário que a localização tenha um carácter absoluto e não dispensa a utilização da navegação LBL baseada na rede de faróis acústicos.

A patrulha sistemática de um dado volume de água poderá ser realizado, recorrendo a definição de subconjuntos deste a percorrer em função do alcance e resolução dos sensores e do objectivo do "survey" ou da extensão do fenómeno a detectar. A definição destes subconjuntos do volume de água induz estratégias de controlo para cada um dos seguintes tipos:

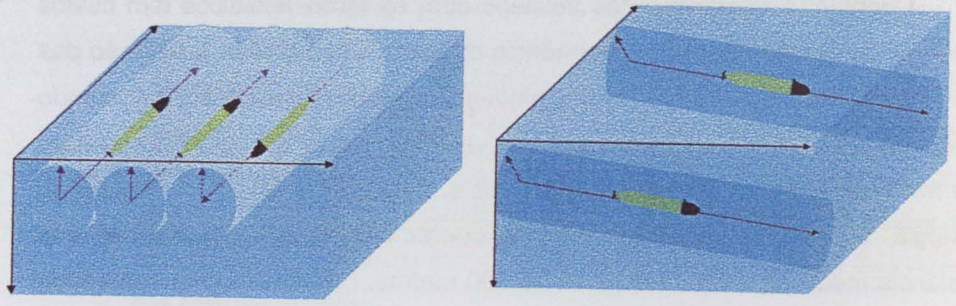
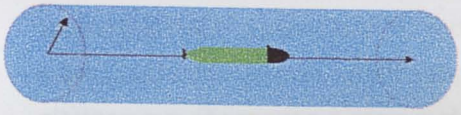


Figura 3.5-1

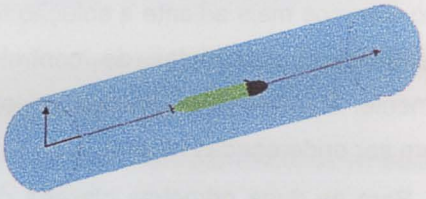
Definição de corredores para a movimentação autónoma submarina

Tubo

Um ***tubo*** é um padrão de restrições geométricas virtual caracterizado por possuir uma direcção e sentido para o movimento, uma fronteira caracterizado por um raio que defina a tolerância do movimento relativamente à direcção, um ponto inicial e um ponto final. Na tabela seguinte podemos observar diferentes formas de ***tubos***.



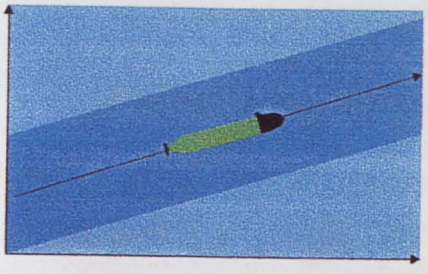
Evolução com variação numa dimensão



Evolução com variação em duas ou três dimensões

Figura 3.5-2

Evoluções alternativas ao movimento



Projecção em duas dimensões

Para a composição de sequências destes tubos é necessário definir uma abordagem para os ligar quer se encontram na mesma profundidade quer se encontrem a profundidades diferentes.

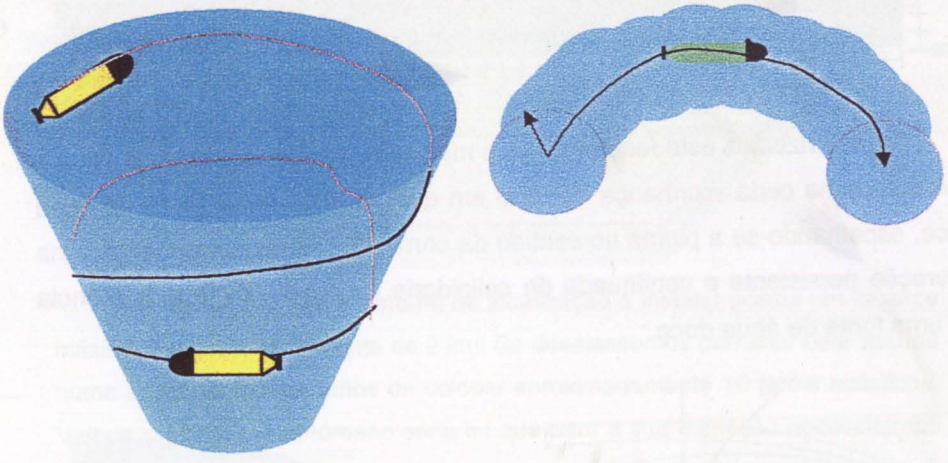
Ligação

Uma ligação é um padrão geométrico "virtual" que pode ser visto como uma curva de raio variável. É caracterizado como uma curva no plano de raio variável e ângulo qualquer, composta com uma taxa de subida na vertical, um

ponto inicial e um ponto final. Na tabela seguinte podemos observar diferentes formas de ligações.

Figura 3.5-3

Configurações de ligações entre tubos



Evolução com variação em todas dimensão

Evolução com taxa de subida nula

Para estas duas tipologias de movimento é possível definir várias estratégias de controlo. Alguns exemplos podem ser analisados em [Estrela 01].

3.6 Caracterização de movimentos para estratégias activas

3.6.1 Caracterização do fenómeno versus movimentação

A abordagem ilustrada no ponto anterior para missões de monitorização ambiental possui um carácter bastante conservador no sentido de que o veículo adopta uma atitude passiva face ao fenómeno em análise. Coloquemos o cenário da detecção de fontes de água doce. As estratégias de pesquisa utilizando quer VSA´s, quer barcos oceanográficos reservam o tratamento da informação recolhida para uma fase posterior ao desenrolar da missão o que torna este método bastante lento.

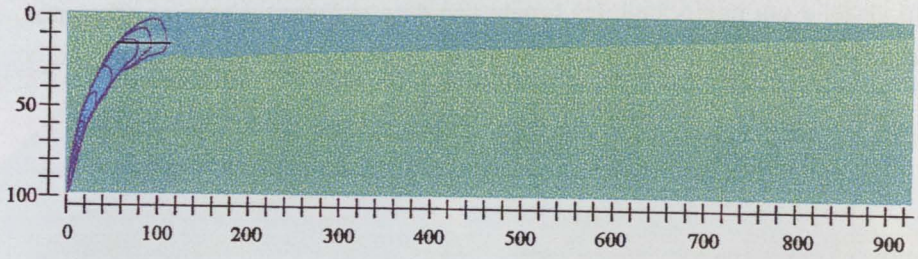
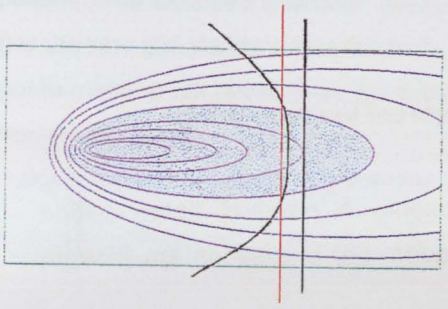


Figura 3.6-1

Corte longitudinal da difusão da pluma de água doce, Caracterização aproximada da difusão da pluma, considerando a corrente constante.

Os tons azulados correspondem a água com menor salinidade.

Caracterizemos este fenómeno com mais detalhe. A salinidade da água é alterada numa certa vizinhança do local em que se encontra a fonte de água doce, espalhando-se a pluma no sentido da corrente. Consequentemente, uma alteração persistente e continuada da salinidade da água indicia a existência de uma fonte de água doce.



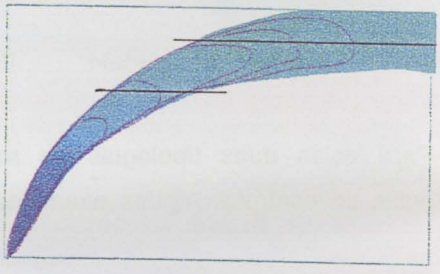


Figura 3.6-2

Corte horizontal paralelo ao nível do mar.

A zona a azul indica que o valor da salinidade está dentro de parâmetros de detecção.

Linhas equi-salinas. Na prática, estas curvas sofrem distorções devidas à influência de outras fontes de água doce e de correntes

O caudal da fonte irá determinar a profundidade a que esta será passível de ser observada. A amplitude da corrente dominante irá determinar em parte a forma da pluma. Esta evolui com forma estreita com correntes de maior amplitude e alarga-se quando a amplitude da corrente diminui. A salinidade é mais baixa no interior da pluma e mais alta no exterior e diminui à medida que a profundidade aumenta.

Se quiséssemos observar este fenómeno utilizando uma abordagem convencional, teríamos que patrulhar com alguma "resolução" um volume de água considerável na vizinhança da fonte. Suponhamos que as informações de carácter hidrogeológico indicam que era possível encontrar uma ou mais fontes de água numa área de trinta quilómetros quadrados (6x5 km^2) com uma profundidade média de 80 metros.

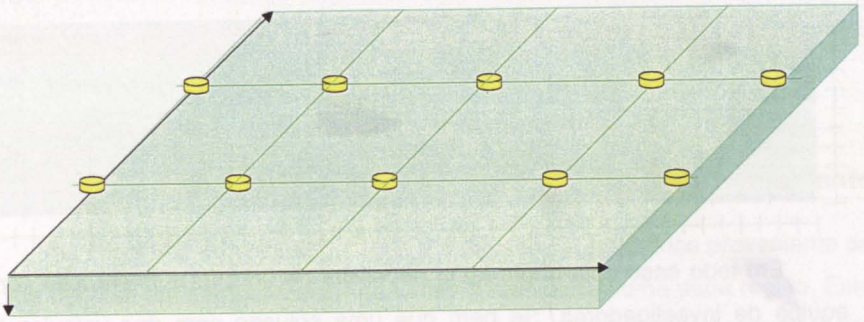


Figura 3.6-3

Plano para a distribuição de faróis acústicos.

Vamos admitir que o sistema de localização a instalar possui um alcance máximo de aproximadamente de 2 km. Se desejássemos patrulhar este volume numa única missão teríamos de colocar aproximadamente 10 faróis acústicos. Vamos supor que o fenómeno seria tal que para a sua detecção necessitamos de dados recolhidos com a seguinte resolução espacial: profundidade 5 metros, longitudinal 20 metros e horizontal 5 metros. Deste modo, precisamos de "patrulhar" 15 planos a diferentes profundidades, desde os 5 m até aos 75 m, e para cada plano, (ou seja 300 segmentos para cobrir 6 quilómetros) realizar um segmento de 5 quilómetros de 20 em 20 m. Estas especificações permitem detectar uma fonte de água doce com um erro máximo de 20 m. Neste cenário razoável, seria necessário recolher 4500000 registos de salinidade e percorrer 22 500 km. Se a velocidade nominal para a missão for de $2ms^{-1}$, seriam necessárias 3125 horas para o VSA realizar a missão. Mesmo operando com dez veículos em simultâneo precisávamos de 312,5 horas de operação. Admitindo que um veículo opera com uma autonomia de 12 horas (e necessita de outras 12 horas para carregar as baterias e para manutenção) por dia, necessitaríamos de 10 veículos durante 26 dias para patrulhar este pequeno volume de água.

É óbvio que, mesmo seguindo um perfil de missão semelhante, seria possível realizar uma missão mais económica, se seguíssemos uma abordagem hierárquica. Primeiro, realizar-se-ia uma missão de patrulhamento com uma resolução mais grosseira nas dimensões profundidade e longitudinal. Depois de analisados os dados, seleccionar-se-ia área mais promissora e repetíamos a operação com maior resolução num volume menor (aproximadamente 10% do original). É obvio que as resoluções espaciais das duas missões dependem da intensidade do fenómeno.

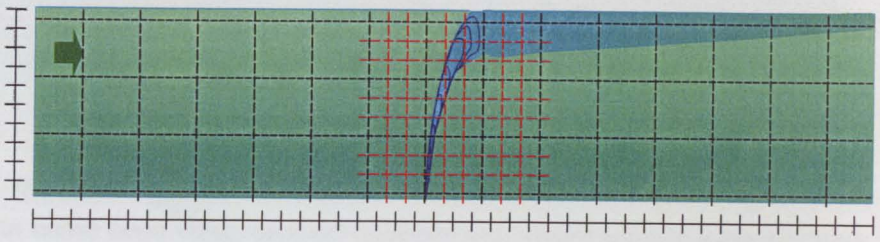


Figura 3.6-4

Patrulhamento hierárquico

1ª fase (velocidade nominal $2ms^{-1}$)

resolução em profundidade: 15 metros

resolução longitudinal: 150 metros

resolução horizontal: 5 metros (não afecta a distância a percorrer)

nº de medidas: 160 000

quilómetros percorridos: 824 kms

10 veículos simultâneos: 1 dia

2ª fase (velocidade nominal $2ms^{-1}$)

resolução em profundidade: 5 metros

resolução longitudinal: 10 metros

resolução horizontal: 5 metros

nº de medidas: 61 750

quilómetros percorridos: 320 kms

10 veículos simultâneos: 1/3 dia

Em todo caso esta abordagem necessita de uma forte interacção com a equipa de investigadores. Se bem que uma solução com dez veículos tipo Remus, é operacionalmente complexa e requer uma infra-estrutura relativamente pesada.

Alternativamente a esta abordagem, podemos equacionar uma outra para a concepção de missões em que a pesquisa é activa e autónoma. Enquanto que no perfil de missões anteriores a determinação do local da fonte era realizado à posteriori através da análise dos dados recolhidos, numa abordagem de pesquisa activa, a determinação da posição da fonte é realizada durante o curso da missão, para tal sendo necessário que as variáveis que caracterizam o fenómeno entrem na malha de controlo.

Após uma breve análise, facilmente se verifica que o caminho para a fonte se encontra na perpendicular às superfícies de equi-salinidade. Apesar de não possuirmos à priori essa informação uma vez que o fenómeno é desconhecido, é possível aproximar estas superfícies através da aquisição de valores de salinidade em tempo real e assim determinar o movimento adequado.

O número de veículos necessários para viabilizar a implementação desta abordagem depende da variabilidade dos fenómenos que afectam a salinidade do volume de água a observar. Voltando ao exemplo atrás em que são necessários 26 dias de missão para patrulhar um volume de água de $30km^2x80m$ usando uma abordagem convencional, rapidamente se conclui a inviabilidade desta pelo facto de ser irrealista esperar que as condições da corrente se mantivessem constantes durante este período.

3.6.2 Movimentação cooperativa

3.6.2.1 Utilização do gradiente

As correntes no volume de água e a difusão da água doce proveniente da fonte definem um campo escalar da variável salinidade numa dada região. Este campo é utilizado para se definir o caminho até ao local da fonte, consistindo uma estratégia possível em seguir o vector gradiente do campo. Apesar desta abordagem deparar com a dificuldade do campo não ser conhecido a priori, poder-se-á admitir que o campo se mantém constante num dado período de tempo e conceber uma abordagem que utilize o método simplex [Pereira 00] para determinar as direcções de "descida" que nos leve até ao ponto de salinidade mínima.

Do ponto de vista conceptual, a utilização adaptada do método passaria pelos seguintes etapas (a que chamamos SimplexV):

1. inicialmente escolher quatro pontos para efectuarmos a leitura da salinidade (estamos a evoluir num cenário de três dimensões);
2. ordenar os pontos por valores crescentes de salinidade;
3. definir o plano (que dignaremos por ß) que contêm os três pontos de menor salinidade;
4. projectar o ponto de maior salinidade relativamente ao ponto no plano ß equidistante aos restantes três;
5. testar um critério de paragem. Se este for positivo, o mínimo encontra-se no polítopo que tem como vértices os pontos do simplex. Doutro modo, prosseguir para o ponto 6.;
6. reordenar os pontos e definir um novo plano ß' que contenha o novo ponto e os dois pontos com menor salinidade que definiam o plano ß anterior;
7. Fazer ß:= ß' e voltar ao ponto 4.

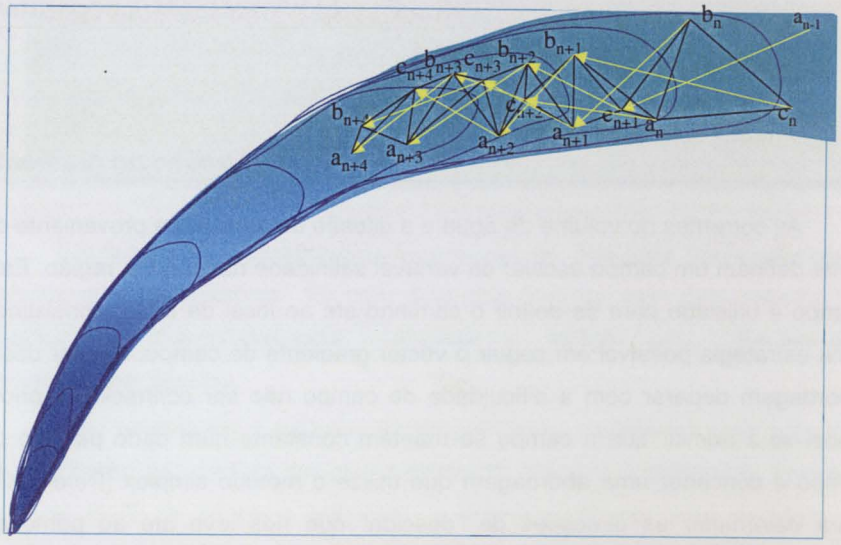


Figura 3.6-5

Evolução do algoritmo ao longo da pluma.

Uma questão que se coloca é a de como utilizar este conceito recorrendo a veículos submarinos autónomos. Os requisitos para a utilização de múltiplos veículos no cenário descrito anteriormente passam por dotar o sistema das capacidades de:

- posicionamento absoluto que poderá ser assegurada por uma rede de faróis acústicos tal como as estratégias de patrulhamento ou por dois barcos com posicionamento absoluto (GPS) que sigam os veículos funcionando como faróis acústicos móveis;
- comunicação de informação para que todos possam conhecer os valores de salinidade do campo que vão sendo sucessivamente observados pelos outros elementos da equipa;
- decisão distribuída para assegurar a adaptatividade através da selecção do veículo que se irá deslocar a um determinada ponto.

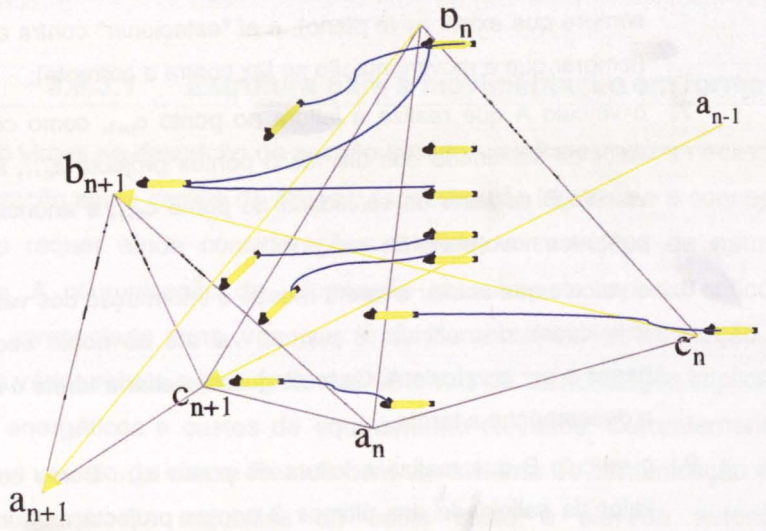
Enquanto que os dois primeiros requisitos são fundamentalmente de natureza tecnológica, o terceiro implica o desenvolvimento de uma estratégia de controlo e decisão distribuída.

Conceptualmente, a abordagem aqui descrita assenta na noção de "cardume". Uma observação rápida dos movimentos dos cardumes revela que estes são caracterizados por um movimento global associado ao cardume como um todo e por movimentos parciais de "sub-cardumes". Há elementos do cardume que tanto se associam a um sub-cardume como, seguidamente, se associam a outro. Parecem movimentos indecisos mas que globalmente resultam num movimento colectivo bem definido.

Figura 3.6-6

Evolução do algoritmo ao longo da pluma

Detalhe de uma iteração.



Consideremos agora os seguintes fundamentos subjacentes à estratégia de implementação:

- admitamos que podemos executar paralela e concorrentemente e de forma distribuída várias instâncias de algoritmos SimplexV;
- estas execuções (ou parte delas) podem ocorrer em todos os veículos em simultâneo;
- qualquer veículo pode ser actor de qualquer uma das instâncias dos algoritmos que executam paralelamente.
- a solução para a atribuição da tarefa "leitura da salinidade" baseia-se na distância do veículo ao ponto de leitura.
- Para todas as instâncias dos algoritmos SimplexV que estão a correr em paralelo temos:
  1. inicia-se com a leitura em quatro pontos iniciais (x,s(x)), onde, no exemplo da figura anterior, $x = a_{n-1}, a_n, b_n, c_n$;
  2. por projecção de a_{n-1} define-se c_{n+1};
  3. por projecção de c_n define-se b_{n+1};
  4. o veículo (que designaremos por A) mais próximo desse ponto, auto selecciona-se para realizar a leitura;
  5. o veículo (que designaremos por B) sem tarefa atribuída que se encontre mais próximo desse ponto auto selecciona-se para realizar a leitura;
  6. todos os outros veículos devem dirigir-se para o último plano ß definido na instância do último algoritmo em que participaram (e

sempre que existir novo plano), e aí "estacionar" contra a corrente (lembrar que a movimentação se faz contra a corrente);

7. o veículo A que realiza a leitura no ponto c_{n+1}, como conhece o valor da salinidade dos últimos 3 pontos projecta a_{n+1}, informa o veículo B do valor da salinidade do ponto c_{n+1}, e anuncia o novo ponto e o novo plano ß';
8. o veículo que aceitar a tarefa recebe a informação dos valores das salinidades dos últimos 3 pontos, vai até ao ponto escolhido e passa a ser o veículo A. Caso ninguém aceitar a tarefa é o próprio a desempenha a tarefa.
9. o veículo B que realiza a leitura no ponto b_{n+1}. Como conhece o valor da salinidade dos últimos 3 pontos projecta c_{n+2}, informa o veículo A do valor da salinidade, e anuncia o novo ponto e o novo plano ß'.
10. veículo que "aceitar a tarefa recebe a informação dos valores das salinidades dos últimos 3 pontos, vai até ao ponto escolhido e passa a ser o veículo B. Caso ninguém aceite a tarefa é o próprio que desempenha a tarefa.

Como se pode observar qualquer veículo pode executar uma dada tarefa numa instância do algoritmo e, após terminar essa tarefa, aceitar uma outra referente a outra instância. Notar que não há veículos com capacidades e funções especiais e todos podem executar as mesmas tarefas. Esta estratégia precisa no mínimo de dois veículos para executar uma instância do SimplexV. Por questões de redundância esta estratégia beneficia da utilização de vários veículos, pois, desta forma, a perda de um veículo não inviabiliza o funcionamento do sistema e quantos mais veículos existirem mais instâncias podem ser lançadas. Esta solução é caracterizada por uma grande utilização de comunicações e, por consequência, um elevado consumo de energia. Contudo este problema pode ser atenuado limitando o alcance das comunicações. Notar que este facto não inviabiliza o funcionamento do sistema, todavia poderá implicar a divisão do cardume em grupos que executam instâncias diferentes.

3.6.3 Movimentação em formação

3.6.3.1 Estrutura para a movimentação em formação

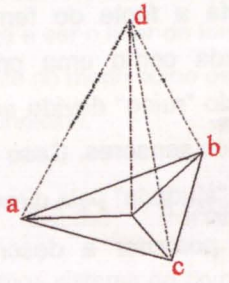
Como vimos na descrição da secção anterior, a infra-estrutura necessária para a operação ainda carece de alguma especificação logística e a concepção da solução requer ainda considerações relativas a restrições de natureza tecnológica. A comunicação de informação constitui um aspecto crítico na estratégia apresentada uma vez que é fundamental na coordenação das acções dos vários robots e pelo facto dos sistemas de comunicação implicarem consumos energéticos e custos de equipamento elevados. Correntemente, o preço de um veículo da classe Remus com um sistema de comunicação é 2,5 do valor original. Os requisitos de baixo custo e elevada autonomia representam pois desafios importantes na concepção do tipo de solução aqui proposto.

Vamos apresentar um método de pesquisa que utiliza informação espacial do fenómeno numa vizinhança da sua origem para definir indicadores de coordenação subjacentes à estratégia de pesquisa. Em vez de utilizar a comunicação no sentido tradicional, esta abordagem baseia-se na percepção do comportamento dos diversos veículos autónomos cooperantes.

Considere-se uma formação de 4 veículos autónomos cujas posições constituam o seguinte simplex em R^3.

Figura3.6-7

Estrutura da formação



Os veículos ***a***, ***b***, e ***c*** são submarinos autónomos e o veículo ***d*** é um barco autónomo. As distâncias entre os veículos e os ângulos formados pelos segmentos que os unem vão depender das características do fenómeno a detectar e da acção de controlo.

Para além de ter um papel determinante na localização dos veículos pois funciona como farol acústico, o veículo à superfície vai permitir obter a referência absoluta da origem do fenómeno.

3.6.3.2 Detecção de indícios da existência do fenómeno

Dado o conhecimento, *a priori*, da forma geral da pluma, a sua manifestação no volume de água em estudo terá, a menos das perturbações na corrente, uma representação como a da figura seguinte.

A monitorização contínua da resistividade e a definição de um valor limiar (este valor assume um carácter relativo) indicia a presença de uma fonte de água doce vai permitir a sua detecção através do deslocamento dos veículos perpendicularmente às superfícies equi-salinas.

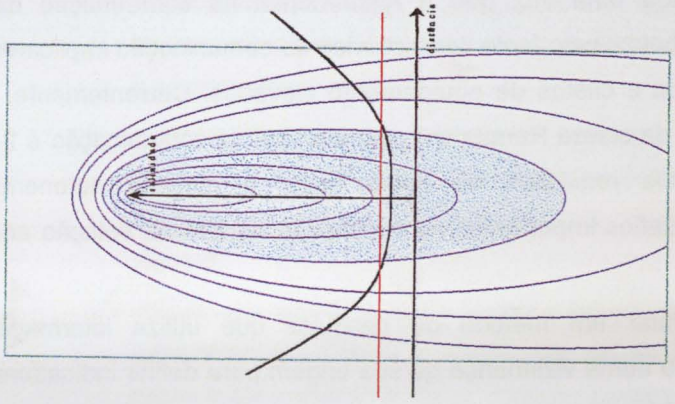


Figura3.6-8

Corte no plano horizontal

Quando um dos veículos entra na área sombreada azul é capaz de detectar o fenómeno

Notar que o sensor é sensível às salinidades mais baixas. No entanto, só consideramos que estamos perante um indício credível caso os valores se encontrem num determinado intervalo.

A estratégia consiste em construir um mapa do deslocamento versus salinidade e, caso exista um mínimo da salinidade abaixo do limiar estipulado para a detecção, então considera-se a como forte a possibilidade de nos encontrarmos perante o fenómeno desejado. O instante em que ocorre o mínimo corresponde a um ponto que pertence a um plano vertical paralelo ao vector da corrente que intercepta a fonte do fenómeno. Na prática, esta direcção só pode ser considerada como uma primeira aproximação pois alguma incerteza é introduzida pelo "ruído" devido ao efeito das correntes, do vento à superfície e da precisão dos sensores. Caso se confirmem os indícios, a estratégia de pesquisa pode ser "ajudada" pela direcção da corrente.

Deixemos para explicação posterior a descrição da estratégia para determinar o caminho até à fonte e vamos analisar como é que a formação aqui proposta detecta os indícios da presença de água doce e os "comunica" aos restantes veículos a funcionar em cooperação.

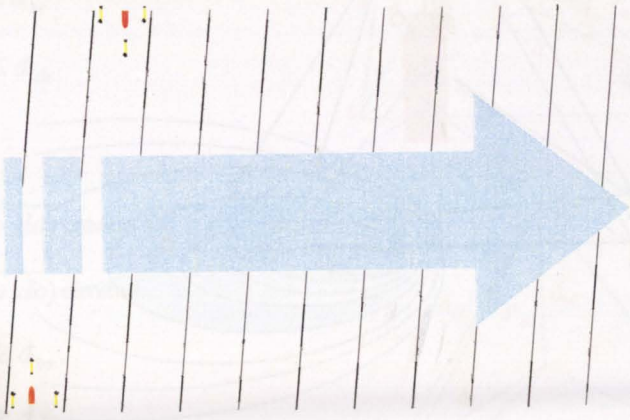
Suponha que a formação vai iniciar a pesquisa num determinado perfil de profundidade e, caso nesse perfil não seja determinado qualquer indício, vai-se aumentado sucessivamente a profundidade até a um valor máximo de operação ou até se considerar que o fenómeno não possui a relevância

necessária para a prossecução da pesquisa. Notar que quanto mais próximo de se estiver da superfície, maior é área onde o fenómeno poderá ser detectado, sendo de referir que a expressão da manifestação diminui à medida que se aproxima da superfície.

Figura3.6-9

Vista no plano horizontal do perfil da missão para determinar os indícios de água doce

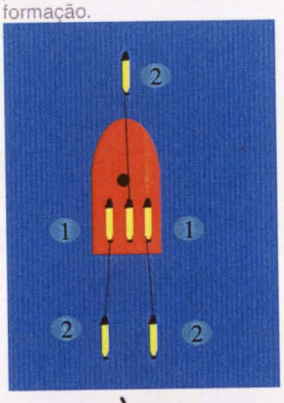
Sombreado a azul a direcção da corrente



Para cada plano que haja necessidade de patrulhar, é preciso percorrer com a formação proposta aproximadamente 106 km (cobrir completamente toda a área, enquanto não definirmos estratégia mais conveniente) e demorar aproximadamente 1 dia de operação, e se algum indício for detectado precisaríamos de percorrer no máximo aproximadamente mais 6 km até encontrar a fonte.

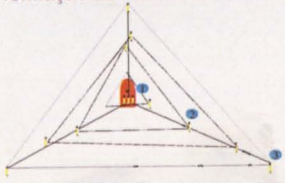
Enquanto não se detectarem vestígios do fenómeno, o líder da formação é o veículo de superfície, que conduz a formação de forma a garantir que o fenómeno possa ser observado. No entanto, quando um dos veículos detecta o fenómeno é esse que passa a ser o líder da formação. Uma questão pertinente que se pode colocar consiste no modo como os outros veículos são informados de que foram encontrados indícios.

1. submarinos autónomos estacionados no barco, para manutenção de informação e carga das baterias.
2. submarinos autónomos abandonam o barco e vão posicionar-se para construir a formação.





Fase de iniciação até atingir a formação de base inicial.



3.6.3.3 Percepção e estado da formação

Uma vez que não temos sistema de comunicação de dados, os veículos vão "perceber" qual o estado da mesma e que acção devem desenvolver através da estrutura e dimensão da formação.

A formação desenhada a azul é a de ***pesquisa***, quando o veículo ***c*** detecta e adopta a estratégia já anteriormente descrita (veículo desloca-se contra corrente), a formação passa a ter a configuração desenhada a preto.

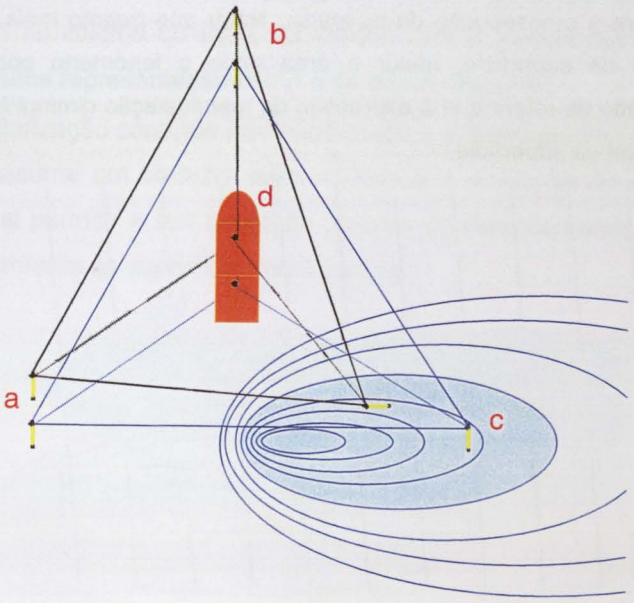


Figura3.6-10

Percepção do fenómeno pelo veículo c

Deste modo, constata-se uma "distorção" consistente na estrutura da formação que, como se pode ver na figura anterior, é facilmente dedutível pelos outros veículos como correspondendo a uma transição de fase. Ou seja, o sistema vai passar da fase de pesquisa para a de confirmação de indícios. Neste exemplo, o veículo ***c*** que detecta o fenómeno e, como se ilustra na figura seguinte, tem-se:

$\angle(\overline{db}, \overline{dc})$ cresce,

$\angle(\overline{da}, \overline{dc})$ diminui;

$d'_{bc} \geq d_{bc}$,

$d'_{dc} \neq d_{dc}$

$d'_{ac} \neq d_{ac}$

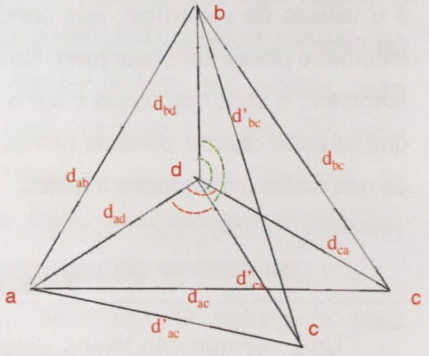


Figura3.6-11

Os diferentes padrões de percepção

De forma similar se tem a detecção pelos veículos ***b*** e ***a***, respectivamente.

Figura3.6-12

Continuação da figura anterior

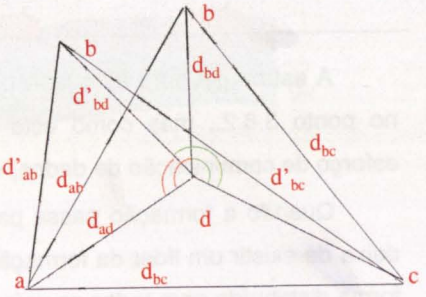
Os diferentes padrões de percepção

$\angle(\overline{db}, \overline{dc})$ cresce

$\angle(\overline{da}, \overline{db})$ diminui,

$d'_{bc} \geq d_{bc}$,

$d'_{ab} \leq d_{ab}$,

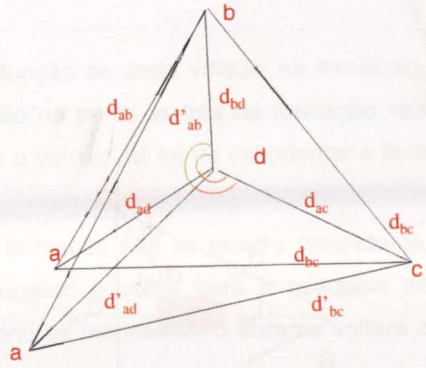


$\angle(\overline{db}, \overline{da})$ cresce

$\angle(\overline{da}, \overline{dc})$ diminui,

$d'_{ba} \geq d_{ba}$,

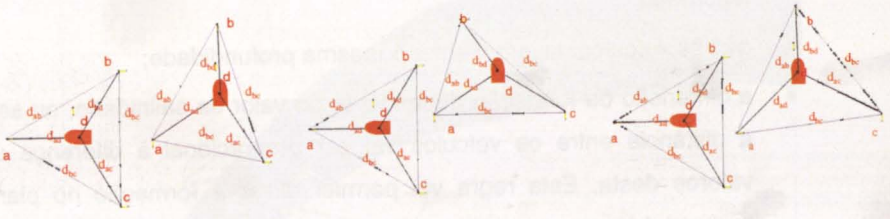
$d'_{ac} \geq d_{ac}$,



Após o sistema constatar que um dos veículos observou indícios do fenómeno – notar que todos os outros sabem quem foi pelo tipo de distorção na formação – uma nova configuração vai ser adoptada. Nesta, o veículo que detectou o indício passa a ser o líder da formação até que esta mude de novo.

Figura3.6-13

Evoluções da formação para a fase seguinte



Os veículos submarinos autónomos assumem uma formação idêntica à anterior em que há um perfil de distâncias e ângulos de referência. No entanto, a frente da formação passa ser o veículo ***a***. O barco autónomo coloca a sua projecção vertical sobre o ponto central do triângulo no plano definido pelos veículos. O veículo líder baixa a velocidade nominal para facilitar a reformulação da formação.

3.6.3.4 Como seguir os indícios

A estratégia para esta fase poderia ser baseada na abordagem seguida no ponto 3.6.2., mas como esta implicava a necessidade de um elevado esforço de comunicação de dados, vamos apresentar uma solução que o evita.

Quando a formação passa para a fase de seguir o fluxo de água doce, deixa de existir um líder da formação e a liderança do processo passa de uma forma distribuída para todos os veículos.

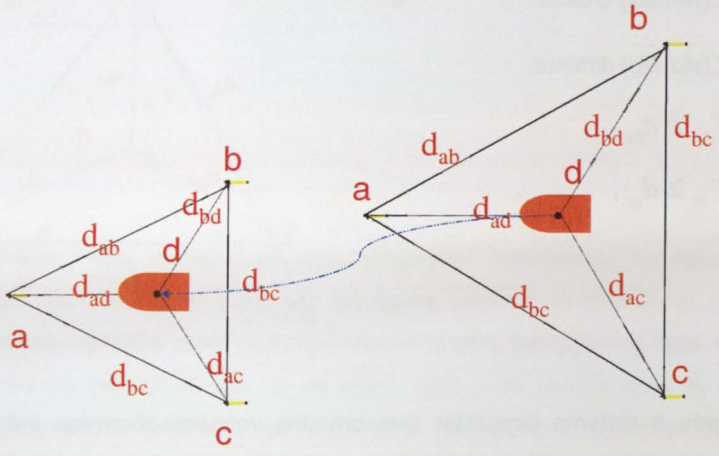


Figura 3.6-14

Após atingir a nova formação procura a melhor direcção

Considere-se uma dada velocidade nominal igual para todos os veículos. A estratégia baseia-se no seguinte:

- a corrente define uma direcção paralela para o caminho da fonte do fenómeno;
- os três veículos devem manter a mesma profundidade;
- a dimensão da formação dependente do valor da salinidade, ou seja a distância entre os veículos vai ser proporcional à diferença de valores desta. Esta regra vai permitir dirigir a formação no plano horizontal (ver na especificação de manobras.

O controlo no plano vertical vai depender do sinal da derivada da salinidade no veículo da frente, uma vez que estamos a evoluir contra a corrente, quando a derivada da salinidade é positiva o veículo deve aumentar a profundidade, e conjugar com a informação da salinidade da parte de trás da formação que o veículo da frente sabe pela distância dos veículos.

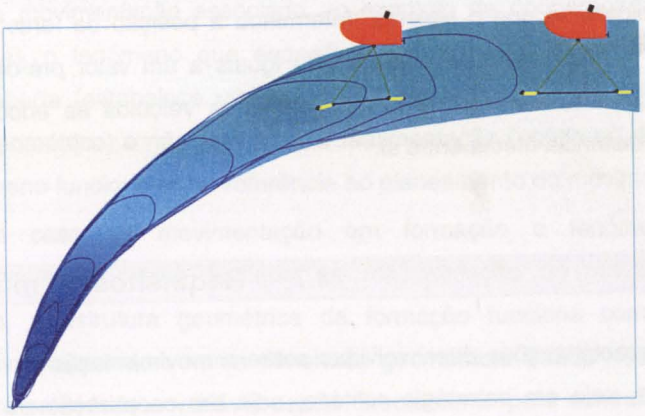


Figura3.6-15

Inicia o mergulho para a fonte

Fazendo um pequeno resumo da função de cada veículo na formação, podemos dizer que os veículos que estão na parte de trás da formação vão orientar a estrutura no plano horizontal, e o veículo da frente vai orientar a taxa de descida da formação.

Caso a variação da dimensão da formação não se mostre consistente, esta deve abandonar esta fase do processo e voltar para a pesquisa de indícios. Seguramente, que se o fenómeno se manifestar, o sistema voltará a detectá-lo numa zona de maior evidência.

Quando o ângulo de descida especificado pelos controladores é superior ao viável sob o ponto de vista físico, a taxa de descida satura e o sistema de controlo faz a compensação através da orientação no plano horizontal, começando a realizar "espirais em torno da pluma"

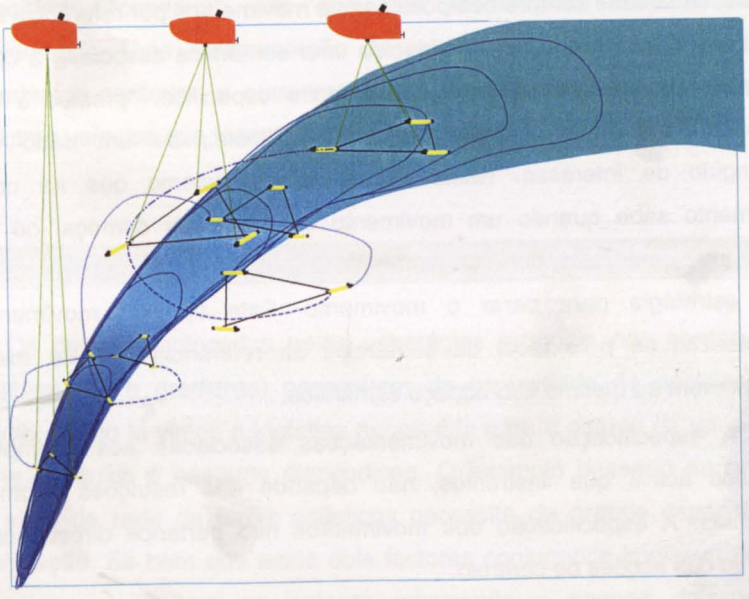


Figura3.6-16

Mergulha em espiral à taxa de máxima descida

Considera-se que o sistema determinou a posição da fonte quando as distâncias médias entre os veículos são iguais a um valor pré-definido. Por exemplo quando, numa certa profundidade, os veículos se encontram a 5 metros de distância média entre si.

3.7 Requisitos do movimento

As caracterizações desenvolvidas sobre a movimentação em ambientes submarinos, seja em estratégia solitária, seja em cooperação ou formação, sugerem uma estrutura topológica (de modo equivalente à robótica terrestre), ou seja simbólica, permitindo que diferentes trajectórias físicas tenham um único representante a nível simbólico.

Esta caracterização define um quadro lógico que é independente do veículo e do seu estado. Quando caracterizamos as fases possíveis de uma pesquisa activa nos exemplos apresentados, não foi necessário ter em conta todos os detalhes do modelo físico real do veículo. Consequentemente, torna-se necessário complementar este quadro com outro que permita incorporar as restrições físicas, estáticas e dinâmicas, que tenham em conta as características específicas do veículo e do ambiente.

As movimentações associadas aos espaços delimitados definidos geometricamente pelas restrições do ambiente, mesmo que estas sejam virtuais, são quase sempre composições de movimentos por referência a essas restrições. Desta forma, é estabelecida uma semântica associada a cada um desses espaços restringidos e que permite especificar "prestes a entrar", "evoluir", "abandonar", "quase na fronteira", etc., de um dado espaço restringido de interesse. Nestas situações, o sistema que irá controlar movimento sabe quando um movimento em particular começa, ou acaba, consegue quantificar a perturbação do movimento, consegue facilmente definir uma estratégia para parar o movimento. Este tipo de movimentações caracterizam-se pelo facto da semântica da referência e a da execução pertencerem ao mesmo sub-espaço semântico.

A especificação das movimentações associadas aos exemplos da pesquisa activa que ilustramos, não depende das restrições geométricas presentes. A especificação dos movimentos não pertence directamente ao domínio das acções de controlo.

Na movimentação associada ao exemplo da cooperação (exemplo do gradiente), o fenómeno que especifica o movimento é uma referência em malha aberta (estabelece uma relação entre um vector de salinidade e um ponto geométrico) e não participa na realimentação "contínua" do movimento. O fenómeno funciona como referência ao planeamento do movimento.

No caso da movimentação em formação o fenómeno participa directamente na malha fechada de realimentação do movimento. Neste exemplo, a estrutura geométrica da formação funciona como sensor do fenómeno mapeando-o numa dimensão geométrica. Sendo neste exemplo a grandeza salinidade a variável usada directamente para controlar a distância entre os veículos da formação.

Esta classe de abordagens estabelece a necessidade de sintetizar um conjunto de estratégias de controlo. No entanto na maioria das situações, as estratégias de controlo passaram pela composição de estratégias elementares, em estratégias mais complexas definidas noutro quadro lógico. Grande parte das estratégias elementares necessárias já foram desenhadas e podem ser particularmente observadas [Estrela 00], contudo estas estratégias e respectivos controladores, não são suficientes para a situação da formação. Enquanto nas situações anteriores o sucesso da estratégia de controlo depende unicamente do veículo que a executa, no caso da formação, o sucesso das estratégias de controlo depende de terceiros veículos. Esta situação leva-nos a uma situação particularmente nova, que é uma estratégia de controlo "contínuo", em que são identificadas duas malhas de realimentação. Uma realimenteção de estado (clássica) das variáveis que pretendemos controlar, e outra realimentação que reporta o comportamento dos outros veículos que fazem parte da formação.

3.8 Requisitos para a percepção, localização e navegação

Os desafios colocados pelas estratégias proposta nos pontos iniciais (patrulhamento e gradiente) necessitam da capacidade de posicionamento absoluto. Como já vimos a logística necessária para o estudo de um pequeno volume oceânico é bastante dispendiosa. O exemplo baseado no gradiente para além da rede de faróis acústicos necessita de grande capacidade de comunicação. Se bem que estes dois factores conjugados implicam soluções dispendiosas, permitem no entanto recorrendo a apenas dois veículos

submarinos determinar o fenómeno. A rede de faróis acústicos pode ser substituída por dois barcos autónomos que acompanham os submarinos.

A terceira proposta que fizemos para a pesquisa baseia-se na capacidade dos veículos navegarem em formação. Como vimos, esta por sua vez funciona como sensor distribuído, e qualquer alteração consistente à sua geometria indicia alterações à distribuição espacial da grandeza que estamos a medir. A questão coloca-se pois no recrutamento da informação que permite a cada veículo calcular as distâncias a que se encontra dos parceiros da formação. Como já referimos o custo associado à capacidade de comunicação de dados é elevado em relação ao custo dos veículos, contudo a tecnologia utilizada para a localização através do uso de faróis acústicos, garante equacionar uma solução que permite a cada veículo calcular a distâncias. Vejamos como.

Todos os veículos possuem relógios sincronizados. Uma das tarefas principais no início da missão é os veículos realizarem a sincronização dos relógios. Com a tecnologia actual esta sincronização pode manter-se válida por largos períodos de tempo (garantidamente superior a 240 horas).

Cada veículo possui a capacidade de emitir vários sinais acústicos identificadores. Cada veículo emite de forma pró-activa e em instantes bem determinados um sinal identificador, ou de forma reactiva emite outros sinais identificadores como resposta aos sinais emitidos pelos outros veículos da formação. Como a nossa formação é composta por quatro veículos vamos ter dezasseis sinais. Dependendo da tecnologia disponível podemos utilizar modulação em frequência para a comunicação e multiplexar os identificadores no tempo, ou usar identificadores em variação de frequência, vulgarmente chamados por "shirp". A segunda hipótese do ponto de vista do tratamento do sinal é bem mais dispendiosa, só se justificando a sua utilização em ambientes com forte saturação acústica.

Cada veículo possui um sinal identificador e três sinais acústicos que utiliza como resposta na identificação positiva de um sinal identificador dos restantes veículos.

A estratégia de envios e respostas proposta permite a cada veículo, não só a observação da distância a cada um dos outros elementos da formação, bem como a distância entre os outros elementos da formação (ver tabela seguinte).

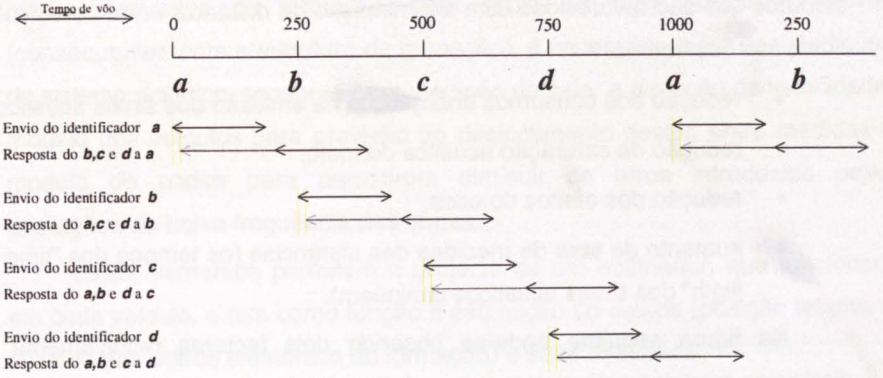


Figura 3.8-1

A escala informa do número de unidades de tempo para multiplexar os sinais acústicos. A frequência a que estes sinais podem ser emitidos depende fundamentalmente da distância a que os veículos se encontram uns dos outros, dos "offset" que é necessário introduzir por questões tecnológicas.

Deste modo, recorrendo a relações trigonométricas simples, é também possível de calcular todos os ângulos entre todos os segmentos que unem os veículos da formação.

Tabela 3.8-1

Distâncias medidas.

Sinais	***Distâncias calculadas pelos veículos***			
	veículo a	***veículo b***	***veículo c***	***veículo d***
a envia identificador		ab	ac	ad
Resposta de ***b*** a ***a***	ab+ba		ab+bc	ab+bd
Resposta de ***c*** a ***a***	ac+ca	ac+cb		ac+cd
Resposta de ***d*** a ***a***	ad+da	ad+db	ad+dc	
b envia identificador	ba		bc	bd
Resposta de ***a*** a ***b***		ba+ab	ba+ac	ba+ad
Resposta de ***c*** a ***b***	bc+ca	bc+cb		bc+cd
Resposta de ***d*** a ***b***	bd+da	bd+db	bd+dc	
c envia identificador	ca	cb		cd
Resposta de ***a*** a ***c***		ca+ab	ca+ac	ca+ad
Resposta de ***b*** a ***c***	cb+ba		cb+bc	cb+bd
Resposta de ***d*** a ***c***	cd+da	cd+db	cd+dc	
d envia identificador	da	db	dc	
Resposta ***a*** a ***d***		da+ab	da+ac	da+ad
Resposta ***b*** a ***d***	db+ba		db+bc	db+bd
Resposta ***c*** a ***d***	dc+ca	dc+cb		dc+cd

Uma vez que nesta abordagem, na maior parte do tempo, a formação mantém-se constante resulta que as derivadas das distancias reais entre veículos assume valores quase nulos, logo a utilização sensores capazes de medir velocidades relativas através de métodos como o efeito de "doppler" não se revela muito interessante.

Um parâmetro importante no desenho do sistema é a distância máxima entre veículos da formação. Este é fortemente dependente resolução dos sensores de salinidade instalados, e, se por um lado quanto maior for a distância máxima entre veículos menor vai ser o tempo de pesquisa necessário efectuar para cobrir um determinado volume, por outro, existe um conjunto de

atributos que são favorecidos com a diminuição da distância entre os veículos, dos quais destacamos:

- redução dos consumos energéticos na emissão dos sinais acústicos;
- redução da saturação acústica do meio;
- redução dos efeitos do ecos;
- aumento da taxa de medidas das distâncias (os tempos dos "time-of-fligth" dos sinais acústicos diminuem).

Na figura seguinte pode-se observar dois factores relativamente às distâncias medidas pelo sistema de acústico, originadas devido à dinâmica da formação.

Primeiro que as medições dos segmentos que unem os veículos vão depender da origem e destino do sinal acústico, ou seja, de uma forma geral, a medição do tempo de voo do sinal acústico do veículo ***a*** para o veículo ***c*** (representada na tabela como ***ac***) não vai ser igual à medição do tempo de voo do veículo ***c*** para o veículo ***a*** (representada na tabela como ***ca***). Estas medidas apenas vão ser idênticas nos casos em que os veículos se deslocam perpendicularmente ao segmento.

Segundo, existe uma discrepância entre a distância entre os veículos e as medidas do sistema acústico. Essa discrepância será maior quanto mais elevada for a velocidade dos veículos e mais longa a distância entre eles.

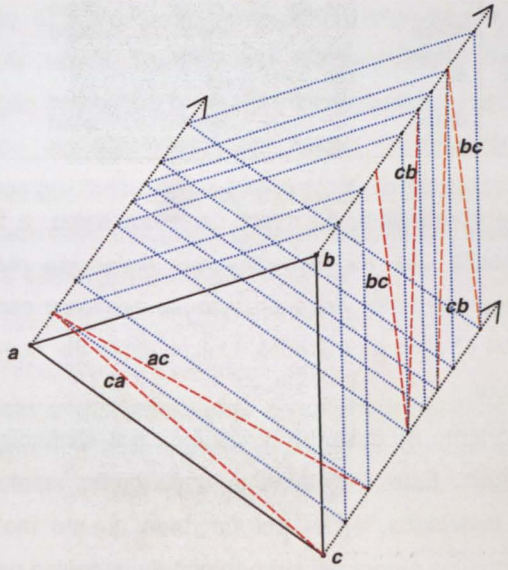


Figura 3.8-2

Mecanismo de avaliação da dinâmica da formação.

Para conseguir estimar descentralizadamente a dinâmica da formação, i.e., cada veículo conseguir estimar as velocidades lineares e angulares dos

outros elementos da formação, bem como as suas posições relativas (consequentemente a estrutura da formação), é necessário além das medições do sistema acústico, sensores para medição do valor e direcção da velocidade, modelo dos veículos para previsão do deslocamento destes entre medidas e modelo de ondas para permitirem diminuir os erros introduzido pelas oscilações de baixa frequência das ondas.

Estes elementos permitem o projecto de um estimador, que funcionará em cada veículo, e tem como função a estimação do estado (posição relativa e atitudes dos outros elementos da formação) e suas derivadas.

Esta página foi intencionalmente deixada em branco

4 ESPECIFICAÇÃO DOS PROBLEMAS DE CONTROLO

4.1 Introdução

Os cenários apresentados nos capítulos anteriores tipificam quatro situações de análise bem distintas e que descrevemos seguidamente de forma resumida.

Da análise que desenvolvemos nos pontos anteriores, ganha particular destaque a necessidade de responder aos desafios pela incerteza da execução.

No cenário do transporte industrial, o desempenho de cada veículo é analisado individualmente, e se existir a necessidade de coordenar esforços, essa coordenação ocorre ao nível da síntese e da atribuição de tarefas a cada veículo. Os veículos podem interagir de forma passiva, ou seja, a decisão de coordenar esforços é exterior aos veículos. Uma vez que a área física de actuação pode ser a mesma, há a necessidade destes identificarem activamente outros veículos na vizinhança da configuração corrente, ou serem informados do comportamento destes.

O cenário do patrulhamento de volumes oceânicos, em que o objectivo é inspeccionar um dado volume, ocorrem problemas equivalentes aos da situação anterior, só que, nesta situação, não existe do ponto de vista da evolução do movimento qualquer interacção no decorrer da missão. Aqui a coordenação é novamente efectuada pela distribuição de objectivos a cada veículo.

O terceiro cenário – o da pesquisa de uma fonte de um determinado fenómeno e a respectiva solução (identificada como estratégia de cooperação) – tipifica uma situação em que o objectivo é comum a todos os veículos. Nesta solução, todos os veículos possuem a mesma função. Não existe nenhum veículo ou sistema com função especial de supervisão, os veículos são caracterizados funcionalmente de forma homogénea e a coordenação emerge da acção de cada um dos veículos. Isto significa que a coordenação ocorre durante a síntese da solução. A solução para a missão (a missão neste cenário é fixa) é do tipo cooperante pois qualquer actividade do veículo contribui para o objectivo. È particularmente interessante verificar que é possível trabalhar neste cenário com um número indeterminado de veículos, desde que este seja maior que o número mínimo. A força da solução surge pela capacidade da interacção dos veículos uns com os outros, ou seja, o conjunto "diminuido" de capacidades de cada veículo é "compensado" com a

riqueza do fluxo informativo e do conhecimento que cada veículo vai "herdando" dos outros.

Por último, o cenário em que usamos a estratégia da formação para a pesquisa de uma fonte de um determinado fenómeno tipifica a situação em que o objectivo é comum a todos os veículos (tal como no caso anterior), mas desta vez cada veículo contribui com uma determinada especificidade. Como, nesta solução, partimos do princípio da necessidade de impor algumas restrições (no exemplo a incapacidade de passar informação relevante entre os veículo) é o conceito de formação que estabelece a coordenação activa. Ou seja, a estrutura da formação por um lado coordena a actividade de cada veículo e paralelamente pelos padrões de perturbação na estrutura da formação constatados por cada veículo funcionam como canal de comunicações entre estes. Neste cenário, a coordenação é intrínseca à solução encontrada para o controlo do movimento.

Independentemente da diversidade dos problemas de controlo enumerados nas quatro situações anteriores, vamos seguir a mesma metodologia para a sua especificação. Vamos iniciar a especificação com uma ligeira abordagem ao planemento da actividade movimento para, seguidamente, referenciarmos as especificação de manobras e, por último, as questões associadas às leis de controlo. Não deixaremos contudo do analisar ainda que de uma forma sumária a problemática associada ao controlo da estrutura de decisão e dos seus subsistemas.

4.2 Como organizar o movimento

A organização do movimento de um sistema autónomo é fortemente dependente do cenário de actuação e dos objectivos propostos para a missão.

A proposta metodológica por nós apresentada no capítulo 6 para o desenvolvimento de sistemas autónomos baseia-se no estabelecimento de diversos níveis semânticos. Enuncia algumas ideias nos permite estabelecer algumas relações entre esses diversos níveis. Transpondo para a especificação do movimento esse resultado, podemos dizer que, quando o problema em análise nos confronta com uma mistura complexa de entidades de natureza contínua e discreta, devemos procurar a solução mais do lado da

observação, da identificação e da partição do espaço de estados, do que do lado do ponto de vista da complexidade do algoritmo de controlo.

Essa linha orientadora leva-nos a especificar a solução para o problema de controlo (movimento) em duas fases.

- A primeira envolve:

  a definição de um conjunto de manobras que são solução para as classes de movimento de acordo com os padrões associados às restrições (especificamos a manobra),

  a definição leis de controlo elementar que suportem a definição das manobras de movimento (especificar o controlador de cada lei de controlo).

- A segunda envolve a especificação do planeador.

Na abordagem por nós proposta, o problema do planeamento do movimento aparece subalternizada, pela estrutura manobra. Nos problemas de busca de fontes com a nossa abordagem o planeamento é automáticamente imposto pela especificação das diferentes fases da missão. Como classificamos as restrições e associamos a cada uma delas as manobras, o problema do planeamento assume nos restantes casos um carácter lógico cujo o resultado é uma lista manobras e as respectivas transições consequência de uma simples pesquisa num grafo.

4.3 Controlo de Movimento

Nos exemplos abordados anteriormente, vimos que os movimentos tinham natureza distinta, tal como: seguir uma "linha", rodar e movimentar em "forward" ou "backward" até atingir uma distância mínima a uma parede. Estes dois movimentos poderiam ser atribuídos a uma manobra para uma acostagem transversal e verificamos que possuem características diferentes, dependendo obviamente as execuções de diferentes leis de controlo. Designaremos estas estratégias elementares de movimento como ***"comandos de movimento"***, ou ***modalidades elementares***.

Consequentemente, necessitamos de estruturar uma entidade que permite a definição de movimentos mais complexos, que resultam da organização da sequência de "comandos de movimento". No exemplo referido, a sequência não está pré-definida, nem tão pouco a parametrização dos

comandos elementares, mas dependem do espaço disponível bem como da distância às paredes.

A esta entidade chamaremos manobra, e organiza-se fundamentalmente em torno de um conjunto de transições que correspondem a determinado conjunto de acontecimentos. Estes acontecimentos estabelecem a relação entre as diferentes modalidades elementares do controlo de movimento.

Define-se então:

Manobra é um conjunto discreto e finito de estados, possuindo cada um destes associado um par: lei contínua de controlo de movimento e conjunto de transições entre os estados.

Comando de movimento é uma estratégia de controlo uniforme, constituindo o elemento básico da manobra.

Observa-se que a definição hierárquica de uma manobra, introduz uma grande flexibilidade aos tipos de movimento executados pelo veículo, e facilmente se compreende da flexibilidade que o sistema adquire com a composição de manobras.

Manobra pode ser um simples movimento de seguir um linha ou uma parede, ou um movimento mais complexo como por exemplo a acostagem.

Uma vez que uma manobra pode enquadrar mais que um comando elementar de movimento, podemos considerar que esta estrutura de movimento define uma lei de controlo híbrido para o veículo.

4.3.1 Manobras associadas aos ambientes industriais semi-estruturados

Definimos duas grandes classes padrões geométricos, os Corredores e as Curvas, e como já vimos uma das manobras mais importante associada a estes padrões é atravessar entrando por uma das fronteiras e abandonando pela outra.

Analisemos e especifiquemos alguns dos casos.

Atravessar corredor

A estratégia a adoptar para passar um corredor depende das características físicas do padrão. Podemos definir um caminho virtual que deve ser seguido definido por um recta. As características dessa recta podem depender da restrição reguladora (parede que se encontra à direita do movimento) e das duas semi-rectas fronteira. No entanto, a transição para essa

estratégia de movimento depende da manobra anteriormente activa e, consequentemente, da estratégia activa no estado anterior à transição. O mesmo acontece na transição para o movimento seguinte. Pode haver necessidade de terminar a manobra "atravessar" de maneira diferente dependendo da manobra seguinte. Na figura abaixo vemos que, na sequência atravessar o corredor (corredor não sombreado), o veículo pode executar uma curva à direita ou atravessar para um novo corredor. Dada a proximidade da curva há necessidade de a preparar mesmo antes do veículo entrar no cruzamento. Como podemos ver na mesma manobra as estratégias associadas à transição para outra manobra são diferentes.

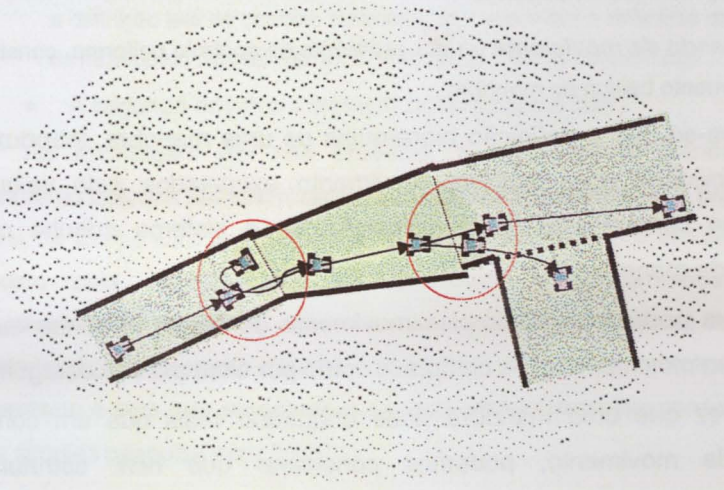


Figura 4.3-1

Manobra atravessar corredor caracterizada por entrada e saída por fronteiras opostas e mantendo a progressão "forward".

Notar que a transição para o comando elementar que vai levar o veículo até à restrição de saída, depende do comando elementar anterior e a transição para o comando da manobra seguinte também deste depende.

No entanto, a definição que apresentamos para manobra endereça situações mais abstractas, tais como entrar e abandonar pela mesma restrição, ou, complementarmente, alterar a forma de progressão.

Portanto, especificar a manobra "atravessar o corredor" consiste em definir os parâmetros associados que condicionam o comportamento da mesma, tais como:

- restrição de entrada , restrição de saida;
- dimensões dos elementos que formam o padrão
- forma de progressão à entrada , forma de progressão à saida;
- transições das manobras que é possível receber o movimento;
- transições para as manobra que é possível entregar o movimento.

Dependendo da estratégia que se irá utilizar para levar o veículo da entrada à saída, existem estratégias para resolver problemas extemporâneos (tais como, ultrapassar obstáculos) que naturalmente têm de estar embebidos

na definição da manobra. Desta forma, a manobra assume um carácter hierárquico, ou seja, é possível definir manobras por composição de outras manobras.

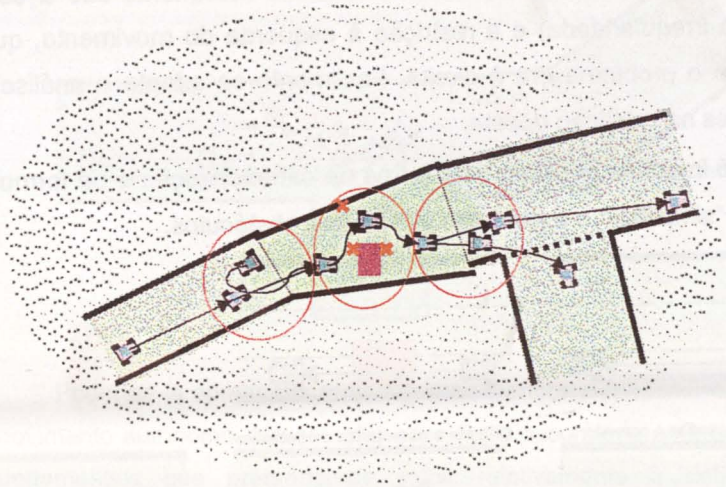
Torna-se portanto necessário formular estratégias de movimento elementares, para as diversas configurações da manobra (ver exemplos da caracterização do movimento no ponto 2.6.)

Figura 4.3-2

Manobra atravessar corredor

caracterizada por entrada e saída por fronteiras opostas e mantendo a progressão "forward". Só que, neste caso, surge um obstáculo, sendo necessário caracterizar a dimensão destes e os pontos críticos relativos ao movimento por forma a definir a manobra para o ultrapassar.

Notar que esta manobra pode alterar a transição para a manobra seguinte.

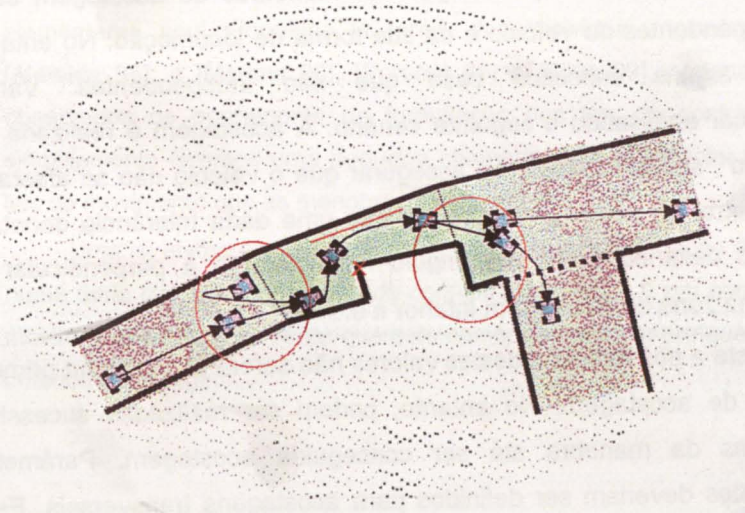


Atravessar curva

A curva é de uma forma abstracta um caso semelhante ao corredor, com a diferença da não linearidade na restrição geométrica. Deste facto, resulta a necessidade de uma estratégia composta que tenha em conta os elementos caracterizadores da não linearidade modelados pela restrição geométrica. Exemplos para a especificação de uma curva do tipo "L" poderá ser vista na em [Almeida 96].

Figura 4.3-3

Elementos caracteristicos da progressão num padrão com uma topologia "especial"



Atravessar padrão especial

Como podemos observar na figura existem "novos" elementos que caracterizam o movimento no padrão. Se realizarmos uma análise por comparação ao corredor verificamos que, após a "detecção" da "irregularidade", os elementos que condicionam o movimento são a esquina (que forma a irregularidade) e a restrição à esquerda do movimento, quando anteriormente o problema era à direita. Facilmente se adapta a análise para irregularidades na restrição oposta.

Como é imediato concluir, esta forma de caracterização e de formulação do problema "simplifica" o tratamento casos com obstáculos.

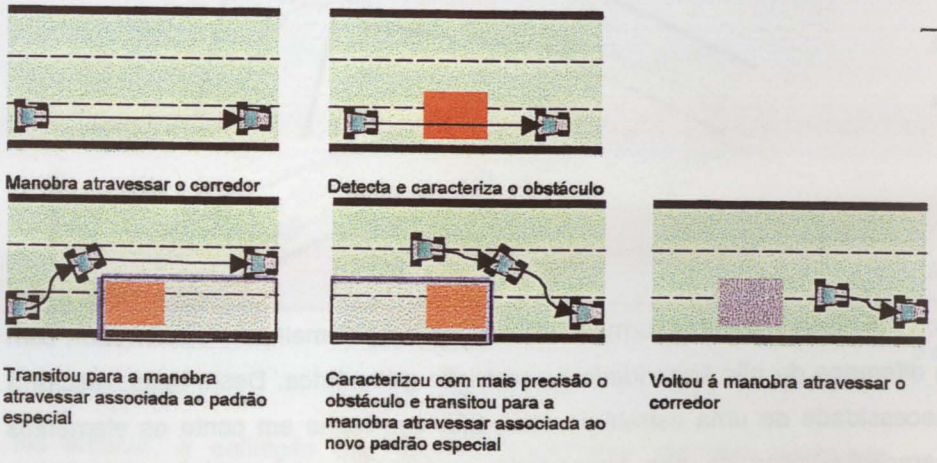


Figura 4.3-4

Resolução do problema do obstáculo por via da agregação semântica

Acostar

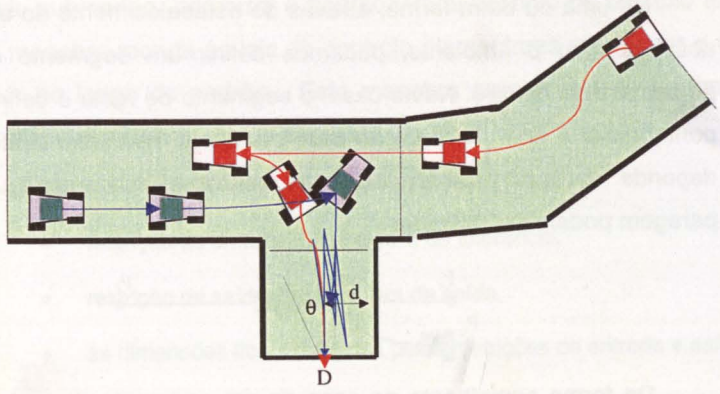
A estratégia e a especificação das manobras de acostagem estão muito dependentes do veículo e da sua forma de locomoção. No entanto, existem alguns conceitos base que são independentes. Vamos exemplificar analisando o seguinte cenário. A acostagem é realizada em "backward", sendo fundamental assegurar que o veículo não se afasta do ponto central de acostagem mais do que uma dada tolerância denotada por ***d*** no desenho e faz um angulo relativamente à perpendicular da estação na zona de acostagem inferior a θ.

Existe a possibilidade destes valores não serem atingidos na primeira tentativa de acostagem. No entanto, podem ser realizadas sucessivas sequências da manobra até ser conseguida acostagem. Parâmetros equivalentes deveriam ser definidos para acostagens transversais. Estes

parâmetros são fundamentais na realimentação das estratégias de controlo.

Figura 4.3-5

Abordagem ao estacionamento



Poder-se-ia apresentar mais algumas manobras fundamentais para o movimento autónomo. No entanto, esse facto esconderia as propriedades fundamentais que pretendemos exibir relativamente à definição de manobra.

4.3.2 Movimentos elementares (Comando de movimento) para manobras de ambientes industriais semi-estruturados

As estratégias de movimentos elementares são, no fundamental, estratégias baseadas numa realimentação uniforme de uma ou mais atributo que relaciona o veículo com o ambiente. Os comandos elementares aqui propostos já foram desenvolvidos originalmente em [Almeida 96], e [Martins 96], [Porobot 98] e [Silva 99], podendo aí ser observados os pormenores de formulação das leis de controlo. Aqui, simplesmente faremos uma pequena chamada para referênciar o tipo de leis de movimento que as manobras propostas vão usar.

Uma lei elementar bastante útil consiste na movimentação do veículo numa dada direcção em função da configuração inicial. É um comando de movimentação relativo. Notar que este comando pode ser enriquecido com critérios de paragem.

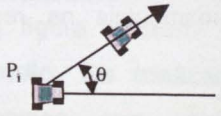


Figura 4.3-6

Comando elementar de movimento: direcção

De uma ou outra forma, através do estabelecimento de uma relação absoluta com o ambiente, podemos definir um segmento de recta a percorrer pelo veículo. Neste caso, o segmento de recta é definido por um ponto inicial e final. A condição para permitir a execução deste comando depende da configuração inicial do veículo. Várias configurações de paragem podem ser definidas.

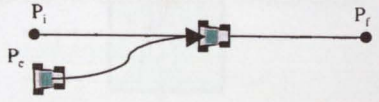


Figura 4.3-7

Comando elementar de movimento: recta

De forma semelhante ao caso de um segmento de recta, pode ser definido um arco de curva através da especificação de um centro de curvatura e um ponto inicial e final. Mais uma vez as condições de paragem e de circulação podem depender da utilização do comando elementar.

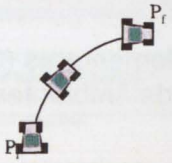


Figura 4.3-8

Comando elementar de movimento: curva

Outra situação muito importante e fundamental para fazer evoluir o veículo em situações especiais - tais como os padrões específicos e, consequentemente, para situações movimento em que existe obstáculos - temos a lei que aproxima o movimento de qualquer ponto do veículo, ao movimento ao longo de uma recta.

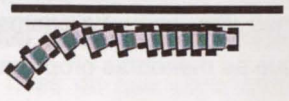


Figura 4.3-9

Comando elementar de movimento: recta tangente

4.3.3 Manobras associadas para movimentação ambientes submarinos virtualmente semi estruturados .

Estas manobras caracterizam-se pela utilização do espaço aberto através de uma estrutura artificial lógica que pemite organizar a movimentação.

Atravessar tubo

A manobra ***atravessar tubo*** é a estrutura lógica que permite ao veículo submarino percorrer o padrão virtual definido no capítulo anterior. Esta manobra recruta as leis de controlo elementares para fazer o veículo evoluir ao longo do padrão. Esta manobra agrega leis de controlo que promovem movimentação no espaço.

A manobra fica definida com os seguintes atributos:

- restrição de entrada, semi-esfera de tolerância,
- restrição de saída, semi-esfera de saída,
- as dimensões ficam definidas pelas restrições de entrada e saída,
- tolerâncias, ou dimensão da secção do tubo,
- transições das manobras das quais é possível receberem o movimento,
- transições para as manobras para as quais é possível entregarem o movimento.

Fazer ligação

A manobra ***ligação*** é a estrutura lógica que permite ao veículo estabelecer a ligação entre dois ***tubos*** cuja a transição da progressão entre os dois não é trivial. É o caso em que a restrição de saída do primeiro ***tubo*** se encontra a diferente profundidade da restrição de entrada do segundo ***tubo*** e não existe um tubo que consiga efectuar a conexão.

A manobra fica definida com os seguintes atributos:

- restrição de entrada, semi-esfera de tolerância sentido para atravessar restrição,
- restrição de saída, semi-esfera de sáida de tolerância sentido para atravessar restrição,
- eixo ou eixos de rotação,
- tolerâncias, ou dimensão da secção do tubo,
- transições das manobras das quais é possível receberem o movimento,

- transições para as manobras para as quais é possível entregarem o movimento.

4.3.4 Manobras associadas à movimentação cooperativa

O tipo de movimentação que resulta da abordagem adoptada para o caso do movimento é bastante simples e pode descrever-se como uma movimentação suave para sucessivos pontos do espaço da movimentação.

Manobra gradiente

Vamos especificar os atributos em função da caracterização realizada no capítulo 3. Enquanto que as manobras nos casos anteriormente definidos podiam possuir um número indeterminado (pelo menos na fase de especificação) de estados, em função das leis elementares de movimento disponíveis, bem como de possibilidades de transição entre essas mesmas leis, neste caso, está bem claro os diferentes estados da manobra. Deixando de fora os promenores de iniciação e de terminação da manobra, identificam-se os seguintes estados: a) dirigir para o plano β (ver caracterização no capítulo 3.6.2) e esperar nova solicitação, b) um dado veículo foi o sistema escolhido e dirige-se para o ponto de leitura indicado, c) realizada a leitura de salinidade selecionar o novo elemento a realizar a nova leitura.

A movimentação vai-se resumir praticamente à sucessiva atribuição de novos pontos para onde o veículo se deve deslocar e aí efectuar as medidas respectivas. Notar que, por comparação com as manobras propostas para a movimentação em formação, o fenómeno causador da busca não "realimenta" directamente o movimento, estabelecendo-se apenas um mapa entre o fenómeno e a referência em malha aberta para a movimentação.

4.3.5 Movimentos elementares para ambientes submarinos com cariz geométrico.

Grande parte de leis de movimento elementares para ambientes submarinos que pretendemos utilizar nas manobras caracterizadas nos pontos anteriores já foram apresentadas em [Estrela 01], pelo que promenores da sua especificaçãoe e síntese podem aí ser encontrados.

Seguidamente apresentamos uma listagem dos referidos movimentos elementares adaptada de [Estrela 01].

Tabela 4.3-1

Comandos elementares para veículos submarinos autónomos com cariz intrinsecamente geométrico

Manobra	Tipo de Trajectória no Plano Horizontal	Tipo de Trajectória em Profundidade	Condição de conclusão
Goto	Segmento de Recta	Constante	Proximidade do extremo do segmento de recta
YoYo	Segmento de Recta	Onda triangular	Proximidade do extremo do segmento de recta
Line	Segmento de Recta	Constante	Tempo
Turn	Referência constante de direcção	Constante	Direcção desejada alcançada
Heading	Referência constante de direcção	Constante	Tempo
Surface	Leme de direcção na posição central	Inclinação constante	Fim de variação de profundidade
Timer1	Leme de direcção em posição constante	Constante	Tempo
Timer2	Leme de direcção na posição central	Leme de inclinação em posição constante	Tempo

4.3.6 Manobras associadas a estratégias de movimento baseadas em formações de veículos.

Uma das soluções que apresentamos para a pesquisa de fontes de água doce foi baseada numa estratégia que estabelece uma formação de veículos com um determinado padrão geométrico. Este padrão geométrico funciona como sensor distribuído e simultaneamente como estratégia de coordenação de veículos. As manobras da formação para as diferentes fases da pesquisa vão resultar da conjugação das manobras do cada um dos veículos, sendo a coordenação garantida pela a estrutura da formação.

Complementarmente, a formação de referência vai permitir a estimação do comportamento dos outros veículos da formação simplificada.

4.3.6.1 Pesquisa indícios

Nesta fase da missão, enquanto não for detectado indícios da presença do fenómeno, o comportamento que a formação irá adoptar será um sequência de manobras, caracterizada pelo "varrimento" de um conjunto de corredores virtuais e as respectivas ligações entre os corredores definidos. Caso a totalidade da área prevista para a pesquisa tenha sido percorrida sem que a detecção do fenómeno em causa tenha

ocorrido, a área será novamente percorrida mas agora com os veículos submarinos noutro patamar de profundidade. Significa isto apenas que a formação muda alguns dos parâmetros geométricos. Assumindo a geometria de uma pirâmide para a formação, significa pois, que esta crescia em altura, mantendo-se o vértice da pirâmide à superfície da água. A manobra da formação resulta das estratégias implementadas pelo líder em conjugação com o objectivo de preservar a formação assegurado pelas manobras de seguir o líder executada pelos outros veículos.

Atravessar corredores e efectuar ligações

Na fase de pesquisa de indícios a formação é liderada pelo barco, sendo pois este a determinar a movimentação nominal da mesma. Consequentemente, a definição de manobras e comandos elementares de movimento para o barco deverão também ter em conta os veículos submarinos principalmente nas transições entre manobras (existem descontinuidades nas transições que precisamos de ter em conta).

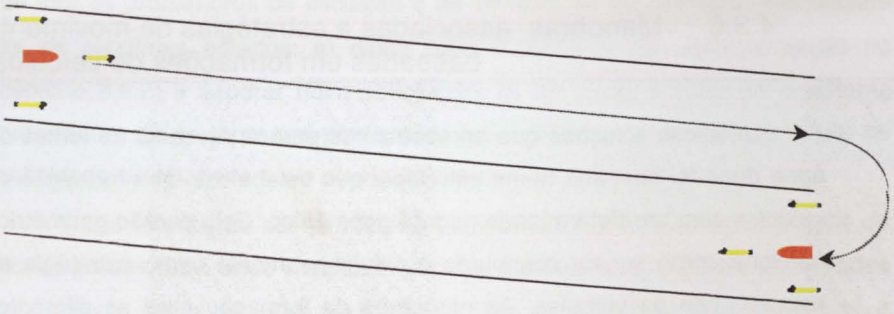


Figura 4.3-10

Plano geral da manobra

Como a geometria destes corredores é bastante regular e o cariz virtual das suas dimensões físicas permite ligeiras violações das suas fronteiras, a manobra associada a atravessar corredor vai recorrer a poucos comandos elementares de movimento (o controlo da manobra é "relaxado", ou seja a violação das restrições poderão não ser totalmente penalizadoras). As manobras praticamente ficam reduzidas e especificadas por comandos elementares do tipo seguir direcções. Eventualmente, na inicialização destas manobras ou após instantes de perturbação, direcções de movimento não colineares com os corredores poderão ser indicadas como referências ou, alternativamente, a utilização de um comando de movimento apropriado para colocar a formação na direcção do corredor pretendido. Se os comandos de movimento forem sintetizados tendo em conta o barco, os submarinos e a estrutura da formação, então é possível

encontrar uma solução para sequência de comando elementares que permite à formação atravessar os corredores.

Barco à superfície

A manobra atravessar corredor, manobra Varrer Corredor, definida para o barco fica caracterizada com os seguintes atributos:

- restrição de entrada, restrição de saída, dimensões dos elementos que formam o padrão,
- transições das manobras das quais é possível receber o movimento,
- transições para as manobras para as quais é possível entregar o movimento.

Nesta manobra, o objectivo é fazer a formação convergir para o ponto médio da restrição de saída P_{1s} (ver na figura seguinte) segundo a direcção que une os pontos médios das restrições que definem o corredor. Quando esta manobra é iniciada com a formação numa posição na vizinhança do eixo central do corredor e com uma direcção de movimento paralelo à direcção, a manobra pode ser realizada recorrendo praticamente a um único comando elementar. No entanto, caso a manobra se inicie após um falso indício, a formação estará em princípio orientada aproximadamente perpendicular à direcção desejada, pelo que será necessário recorrer a uma sequência de comando elementares.

A manobra estabelecer ligações - manobra Ligar Corredor - pode ser virtualmente confinada a um semi-anel que liga dois corredores paralelos, pelo que a manobra definida para o barco fica caracterizada com os seguintes atributos:

restrição de saída do corredor corrente;

restrição de entrada do próximo corredor, sentido de atravessamento.

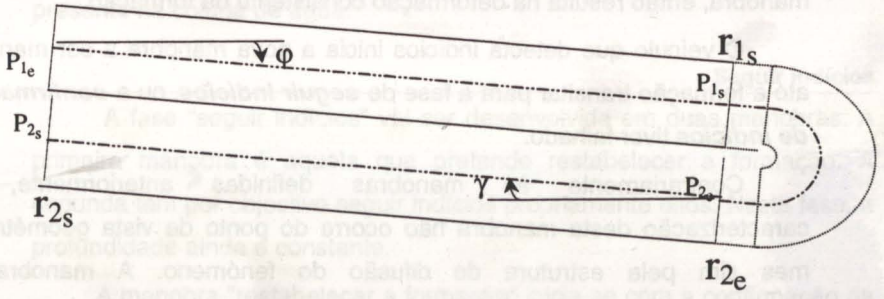


Figura 4.3.11

Caracterização das manobras atravessar e ligar corredores.

Os pontos P são os pontos médios das restrições.

Na manobra ligar corredor, o objectivo é fazer a formação entrar no próximo corredor na vizinhança do ponto médio da restrição de entrada e

segundo a direcção paralela definida pelos pontos médios das restrições que definem o próximo corredor (no caso da figura anterior a direcção). O comando elementar a que esta manobra irá recorrer com mais frequência será ao comando elementar que fará o barco descrever um movimento circular.

Submarinos

Para os restantes membros da formação, neste caso os submarinos, a manobra - manobra Seguir Líder - é do ponto de vista conceptual bastante simples pois a sua definição passa simplesmente pelo conceito de seguir o comportamento do líder e manter a formação. Deste modo, os atributos necessários definir para a caracterização da manobra são:

o veículo da formação a seguir, neste caso o barco,

a estrutura da formação, ou seja, profundidade de pesquisa, distância e ângulos aos outros veículos da formação.

Esta manobra define-se para uma profundidade constante.

Confirmar indícios

A manobra global da formação nesta fase da missão irá resultar numa deformação consistente da mesma no plano.

A confirmação de indícios é iniciada pelo veículo submarino que os detectou. Só este é que vai seleccionar uma nova manobra de controlo. Do ponto de vista da estratégia de controlo, os restantes veículos submarinos vão manter a manobra seguir barco, mesmo sabendo que há um veículo que coloca a hipótese de ter encontrado indícios de um dado fenómeno. O barco vai manter a manobra atravessar corredor só que diminui a velocidade nominal. Se o veículo que detectou os indícios mantiver a nova manobra, então resulta na deformação consistente da formação.

O veículo que detecta indícios inicia a nova manobra a ser mantida até a formação transitar para a fase de ***seguir indícios***, ou a ***confirmação de indícios*** tiver falhado.

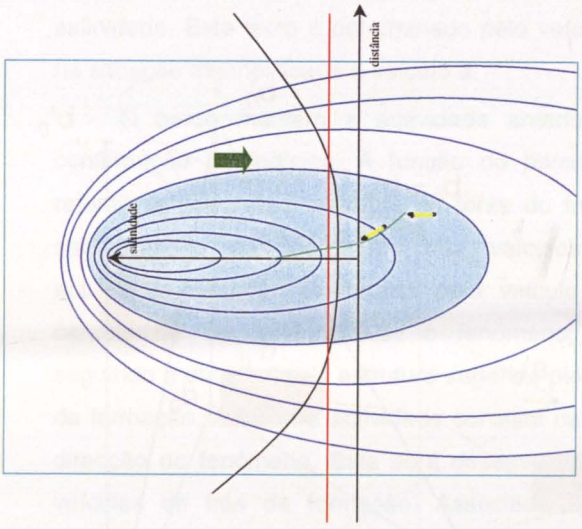
Contrariamente às manobras definidas anteriormente, a caracterização desta manobra não ocorre do ponto de vista geométrico, mas sim pela estrutura de difusão do fenómeno. A manobra é caracterizada pela direcção e sentido da corrente e pela taxa de variação da salinidade da água.

A manobra confirmar indícios é caracterizada com os seguintes atributos:

- direcção da corrente da massa de água;
- salinidade.

Figura 4.3-12

Solução para confirmar indícios



Uma vez que a estratégia para a detecção de indícios (como se pode ver na figura), só é possível após o veículo ultrapassar o ponto que, segundo a direcção da corrente passa pela fonte do fenómeno, a manobra deve iniciar o movimento com sentido contrário ao da corrente corrigida de um certo ângulo. Enquanto o sistema desenvolve esta movimentação, a salinidade deverá decrescer e dessa forma confirmar a existência do fenómeno. A estratégia de movimento está directamente relacionada com a capacidade de confirmação da existência dos indícios. Tal só se confirma se o movimento do veículo convergir para a direção contrária à da corrente presente na coluna de água.

Seguir indícios

A fase "seguir indícios" vai ser desenvolvida em duas manobras. A primeira manobra é aquela que pretende restabelecer a formação. A segunda tem por objectivo seguir indícios propriamente ditos. Nesta fase, a profundidade ainda é constante.

A manobra "restabelecer a formação" inicia-se com a confirmação da fonte de salinidade baixa. Essa confirmação ocorre porque os veículos detectam uma deformação consistente da formação. Para o

restabelecimento da formação, vamos considerar o seguinte. Até ao restabelecimento da formação, o líder da formação passa ser o veículo que detectou os indícios (no exemplo da figura o veículo ***c***) e mantém o movimento da fase anterior. Com base neste veículo, os restantes submarinos redefinem a sua posição na formação e passam a ter como objectivo uma posição numa configuração virtual, que depende do veículo líder.

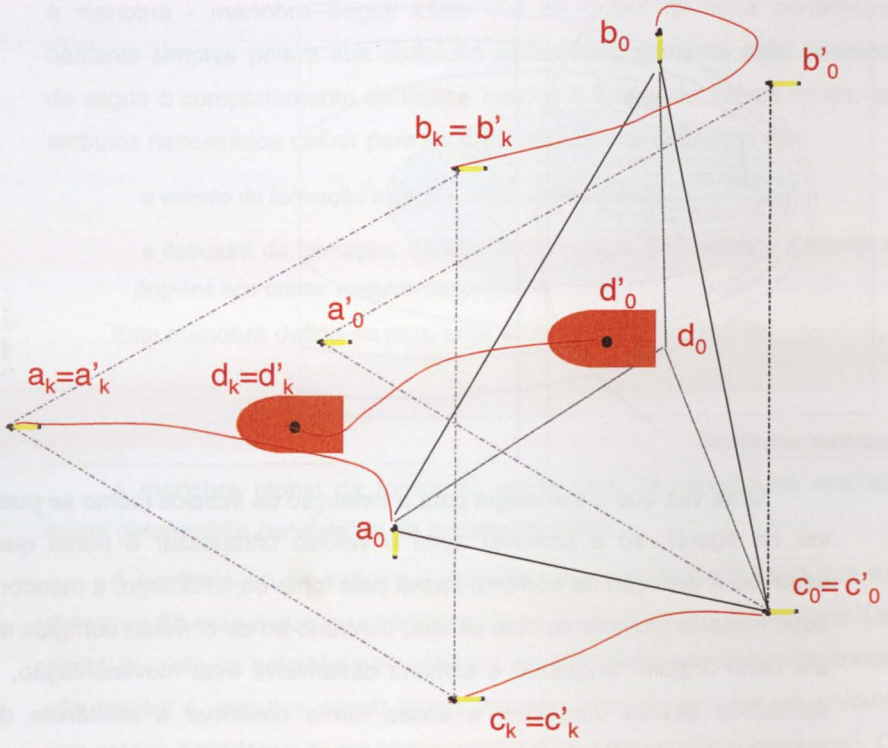


Figura 4.3.13

Exemplo em que

*O índice **0** define o instante em que o sistema começa a reformular a formação, o índice **k** é o instante em que a formação atingiu o padrão estabelecido. A apóstrofe indica a formação virtual.*

*No caso, presente em que é o veículo **c** a confirmar o indício, a frente da formação passa a ser o veículo **a**, o veículo que detecta passa ser o lado esquerdo da formação e o veículo **b** o lado direito.*

Consideramos que, enquanto a formação real não estiver numa vizinhança da configuração virtual, mantemo-nos na manobra para restabelecer a formação. Ou seja,

> o barco tenta que a sua projecção no plano definido pelos submarinos, esteja numa vizinhança do ponto médio do triângulo definido pelos submarinos,
>
> os outros dois submarinos minimizam a distância à sua posição virtual.

A manobra "redefinir formação" é caracterizada - quer para o barco, quer para os submarinos - com os seguintes atributos:

- dinâmica do veículo líder e a respectiva posição relativa,
- estrutura da formação para esta fase da missão.

Restabelecida a formação inicia-se outra manobra, denominada ***seguir indícios***. A estrutura do fenómeno continua a servir metodologicamente para a definição desta manobra. A manobra da formação vai ser desenvolvida por três actividades bem distintas associadas aos diferentes veículos. Esta será utilizada enquanto a evolução no plano de detecção dos indícios permitir uma diminuição da salinidade. Este facto é determinado pelo veículo da frente da formação - na situação exemplificada o veículo ***a***.

O barco mantêm a actividade anterior, aliás desde a fase de confirmação de indícios. A função do barco é a de estabelecer uma referência geográfica absoluta da fonte do fenómeno pelo que segue a sub-formação definida pelos três veículos submarinos. Uma outra actividade - função executada pelo veículo da frente – consiste em caracterizar em profundidade o fenómeno, mantendo a formação e seguindo a dimensão da estrutura sugerida pelos veículos da parte de trás da formação. A terceira actividade consiste na orientação da formação na direcção do fenómeno. Está será desenvolvida de forma simétrica pelos veículos de trás da formação. Associado ao carácter estruturante do fenómeno, existem ainda dois denominadores comuns a estas manobras nesta fase da missão: a profundidade constante e a velocidade linear de referência para a formação.

Assim, a manobra descrita para o veículo submarino da frente, cujo o objectivo é definir o instante em que a formação, deve começar a descer fica complementarmente caracterizada com os seguintes atributos:

dimensão da formação, em particular distância aos veículos de trás e ao ponto médio do segmento que os une - a dinâmica dos veículos de trás é seguida pela manutenção desta característica da formação,

salinidade, ou seja a variação da salinidade.

A manobra para os veículos submarinos de trás, cujo o objectivo é definir um rumo para a formação, fica complementarmente caracterizada com os seguintes atributos:

a dinâmica do veículo oposto na formação,

taxa de variação da dimensão e da velocidade em função da salinidade,

salinidade, ou seja a variação da salinidade.

Mergulhar para a fonte

As manobras associadas a esta fase da missão acrescentam relativamente à fase anterior a dimensão da profundidade às manobras. Este conjunto de manobras é concebido tendo em conta a forma como o fenómeno se manifesta.

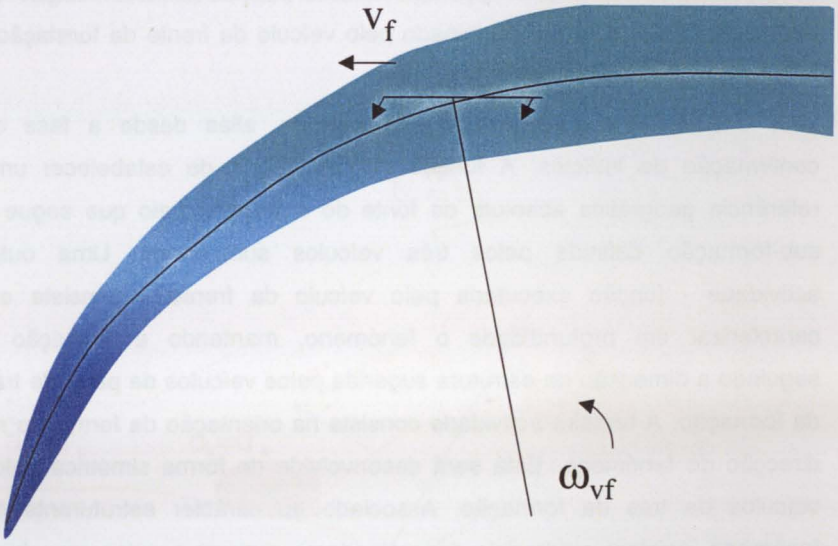


Figura 4.3-14

Formação mergulha para a fonte

Em profundidade, o melhor posicionamento da formação será a situação em que o veículo da frente se encontra acima da linha de máxima descida e os veículos de trás abaixo. Para controlar a taxa de descida da formação vamos utilizar a forma como as superfícies equi-salinas se desenvolvem. A ideia que vamos utilizar é a taxa de descida ser controlada pelo veículo da frente da seguinte forma. O veículo da frente através do mapeamento da salinidade medida em função do seu movimento pode determinar se está a baixo ou acima do caminho de máxima descida. Dessa forma, através da taxa de variação da salinidade medida pode-se determinar se se deve aumentar a taxa de descida de veículo ou não. Por exemplo, se o veículo da frente se encontrar abaixo do caminho de máxima descida e a salinidade aumentar o veículo deve diminuir a profundidade da formação (ou seja subir a formação), caso se encontre a cima do caminho de máxima descida, ocorrendo variação semelhante de salinidade deve aumentar a profundidade da formação. A alteração da dinâmica vai provocar uma variação da velocidade angular da formação no plano vertical, que pode ser observada e seguida pelos veículos de trás da formação

Devido ao limite dinâmico dos veículos a taxa de descida para a formação a determinada altura satura. Consequentemente, a determinado momento a formação afastava-se do fenómeno. Todavia, esta situação não coloca em causa a estratégia pois as manobras no plano horizontal provocam uma rotação da formação em torno do fenómeno, levando a que a formação desça para a fonte num movimento em espiral. Contudo, se a bissectriz da formação se encontrar no plano vertical que contêm o caminho de máxima descida, a formação "perde o fenómeno". Pelo que as manobras no plano horizontal deve possuir um ligeiro "offset" relativamente à direcção.

Marcar alvo

A fase da marcação do alvo ocorre quando a formação satura em profundidade (ou em altura mínima). Desta forma, a formação fica a rodar a uma profundidade fixa em torno da fonte, e o barco regista esse facto dando início ao fim de missão. Consequentemente, as manobras para a fase de marcação alvo são as mesmas da fase anterior

4.3.7 Movimentos elementares associados às manobras das formações

Movimento elementar seguir líder

Esta lei de movimento elementar fundamenta-se na possibilidade de todos os veículos serem capazes de estimarem a dinâmica do movimento ($\omega \quad v$) dos restantes veículos da formação. Para o caso presente, só estamos a considerar o movimento no plano. Esta lei de movimento vai ser composta por duas componentes, uma de "feedforward" $[\omega_{ff}, v_{ff}]$, e uma outra de realimentação $[\omega_r, v_r]$.

A componente de feedforword depende directamente da dinâmica desejada para a formação. No presente caso, o líder é barco e consideramos então ω_d, v_d a velocidade angular e linear estimada pelo veículo seguidor da projecção do barco no plano formado pelos submarinos (ver figura seguinte).

Deste modo, o movimento desejado para a formação será $[\omega_{for}, v_{for}] = [\omega_d, v_d]$, sendo possível determinarmos:

- $[\omega_{ff}, v_{ff}] = f(\omega_f, v_f)$ uma referência em malha aberta para velocidade angular e linear do veículo.
- centro de rotação da formação, c_{rf}.

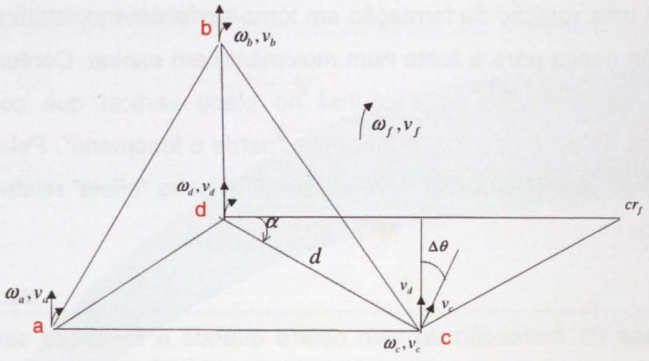


Figura 4.3-15

Comando elementar seguir líder exemplificado com a situação do veículo ***c*** *seguir o barco*

A componente têm por objectivo manter a formação, corrigindo os efeitos dos erros de estimação e das perturbações a que cada veículo é sujeito. Os parâmetros da realimentação são:

O erro na direcção do movimento do veículo relativamente ao veículo líder, determinado pelo ângulo $\theta = \angle(v_d, v_c)$ - na figura está exemplificado com o veículo ***c***.

O erro do ângulo α (ângulo formado pela raio de rotação da formação, e a recta que une o veículo ao veículo líder) denominado por $\Delta\alpha$ em função do ângulo estrutural (que no caso de uma formação com os lados todos iguais será de 30°).

O erro da distância entre os dois veículos e a distância definida pela estrutura da formação Δd.

Então a componente de realimentação será

$$[\omega_r, v_r] = f(\theta, \Delta\alpha, \Delta d)$$

pelo que o comando elementar de movimento será especificado por:

$$[\omega, v] = f(\omega_f, v_f) + f(\theta, \Delta\alpha, \Delta d).$$

Caso o líder fosse qualquer outro dos veículos, a mesma estratégia podia ser seguida, a partir da referência da dinâmica do veículo líder para a dinâmica da formação. Esta lei devidamente adaptada vai ser usada pelo barco quando este passar a seguir a formação por forma a manter sua posição "por cima" do ponto médio da formação.

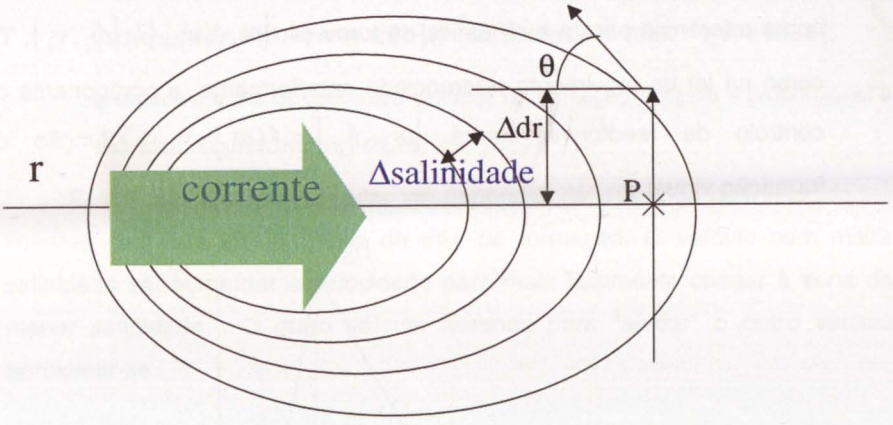
Movimento elementar confirmar indícios

Esta lei de movimento suporta a manobra "confirmar indícios" e só se aplica ao veículo que pretende confirmar indícios.

A dinâmica do veículo vai depender duma referância em malha aberta. Esta referência é definida como a recta (ver na figura seguinte a recta ***r***) paralela à corrente que passa sobre um ponto (o ponto P) que pertence à direcção estimada até à fonte do fenómeno. Este ponto é estimado como o o ponto de mínimo salinidade na trajectória do veículo encontrado na fase anterior do movimento.

Figura 4.3-16

Comando elementar para o veículo que pretende confirmar indícios



A lei de movimento vai então ser a soma das duas seguintes componentes:

$[\omega_{ff}, v_{ff}] = f(r, \Delta dr, \Delta\theta)$, a referência em malha aberta para velocidade angular e linear do veículo.

$[\omega_r, v_r] = f(\Delta salinidade)$, a compensação para fazer progredir no caminho mais rápido até à fonte.

A componente deve convergir para zero $[\omega_r, v_r] \rightarrow [0,0]$, ou ficar limitada. Caso tal não aconteça, então é sinal que os indícios não se confirmam.

Consequentemente, o comando elementar de movimento será especificado por:

$$[\omega, v] = f(r, \Delta dr, \Delta\theta) + f(salinidade)$$

Esta lei elementar também vai ser usada na manobra confirmar indicios, pelo veículo líder até a formação restabelecer a nova estrutura.

Movimento elementar re-formação

Esta lei de movimento elementar utiliza-se nos veículos que seguem o submarino que detectou indícios na fase ***seguir indícios*** na manobra que é reponsável pelo estabelecimento da nova configuração da formação. A componente da lei da malha aberta é em todo semelhante a lei de controlo ***seguir líder***. Nesta situação o veículo a seguir será o que abandonou a formação inicial, no exemplo que estamos a utilizar para ilustrar esta situação será o veículo ***c***, então seja ω_c, v_c a velocidade angular e linear estimada pelos veículos seguidores, que deve ser usada como referência para a a dinâmica da formação $[\omega_{for}, v_{for}] = [\omega_c, v_c]$. Tal como na lei de movimento referenciada anteriormente, a componente de controlo de feedforward será $[\omega_{ff}, v_{ff}] = f(\omega_{for}, v_{for})$ função da formação virtual.

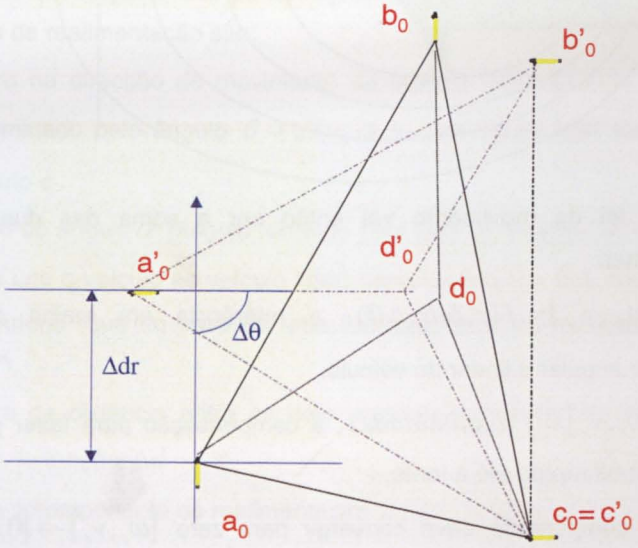


Figura 4.3-17

Comando elementar para os submarinos reconstruírem a formação.

A componente de realimentação têm por objectivo levar o mais rapidamente possível a formação real para a dimensão e estrutura da formação virtual. O objectivo é minimizar $[\Delta dr\, \Delta\theta]$

Então, a componente de realimentação será

$$[\omega_r, v_r] = f(\Delta\theta, \Delta dr)$$

Consequentemente, o comando elementar de movimento será especificado por:

$$[\omega, v] = f(\omega_c, v_c) + f(\Delta\theta, \Delta dr)$$

Movimento elementar, orientar a formação no seguimento de indícios

Esta lei de movimento elementar é utilizada pelos veículos de trás da formação e tem por objectivo orientar a formação na direcção da fonte, no plano de detecção (ou seja à mesma profundidade).

Esta lei baseia-se nos seguintes prossupostos:

a formação deve seguir uma velocidade linear de referência em função da salinidade medida pelos veículos,

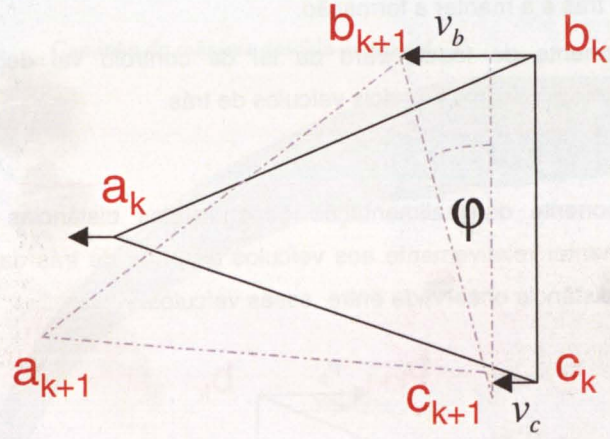
cada veículo vai alterar a velocidade linear de referência em função da salinidade, de forma relacionada com a seguinte proporcionalidade $[salinidade = 0 \rightarrow v = \min]$ e $[salinidade = \max \rightarrow v = \max]$,

a distância entre os diferentes veículos da formação também é proporcional à salinidade ($[distâncias]\ \alpha(salinidade)$);

Então, a velocidade ângular vai depender da diferença das velocidades lineares dos veículos da parte de trás da formação. O veículo com maior salinidade vai aumentar a velocidade para mais facilmente chegar à zona da menor salinidade. O outro veículo abranda para "ajudar" o outro veículo aproximar-se.

Figura 4.3-18

Orientação no plano da formação.



Pelo que a velocidade ângular de referência para a formação será dado por $\omega_f = f(v_c, v_b)$. Então, a componente de feedforward para a lei de controlo para cada veículo (estamos só a falar dos veículos da parte de trás da formação) vai ser $[v_{ff}, \omega_{ff}] = [f(salinidade), f(v_c, v_b)]$.

À componente de realimentação vamos atribuir várias funções. A primeira é reforçar a componente de feedforward através do reforço ou relaxamento das

variações da direcção do movimento em função das variações de salinidade. A segunda é definir a dimensão da estrutura da formação.

Deste modo, a componente de realimentação será

$$[\omega_r, v_r] = f(\Delta salinidade, \omega, v),$$

pelo que o comando elementar de movimento será especificado por:

$$[\omega, v] = [f(salinidade), f(v_c, v_b)] + f(\Delta salinidade, \omega, v).$$

Esta lei foi esboçada no plano paralelo à superficie do volume de água, mas podemos desenvolvê-la no plano definida pela posição definida pelos três veículos submarinos.

Movimento elementar, acompanhar a formação no seguimento de indícios

Esta lei de movimento destina-se ao veículo da frente da formação para ser usado nas manobras da fase seguir indícios.

A função deste veículo consiste em seguir a direcção de pesquisa proposta pelos veículos da parte de trás da formação e detectar o momento em que é necessário iniciar o processo de "mergulho" para a fonte. Esse instante acontece quando o veículo detecta que da progressão no plano corrente não resultam diminuições da salinidade. O seu papel fica então reduzido a seguir os veículos de trás e a manter a formação.

A componente de feedforward da lei de controlo vai depender da velocidade que este observa nos dois veículos de trás.

$$[v_{ff}, \omega_{ff}] = f(v_c, v_b)$$

A componente de realimentação depende das distâncias que este veículo quer manter relativamente aos veículos da parte de trás da formação em função da distância observada entre esses veículos.

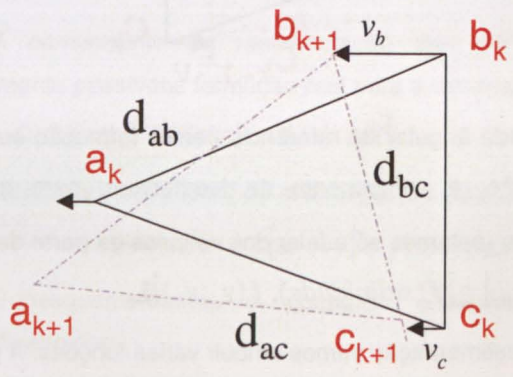


Figura 4.3-19

O veículo da frente segue a proposta de orientação com origem nos veículos de trás da formação..

Portanto d_{ab} e d_{ac}, dependem de d_{bc} observado.

Então a componente de realimentação será

$$[\omega_r, v_r] = f(d_{ab}(d_{bc}), d_{ac}(d_{bc})),$$

pelo que o comando elementar de movimento será especificado por:

$$[\omega, v] = f(v_c, v_b) + f(d_{ab}(d_{bc}), d_{ac}(d_{bc})).$$

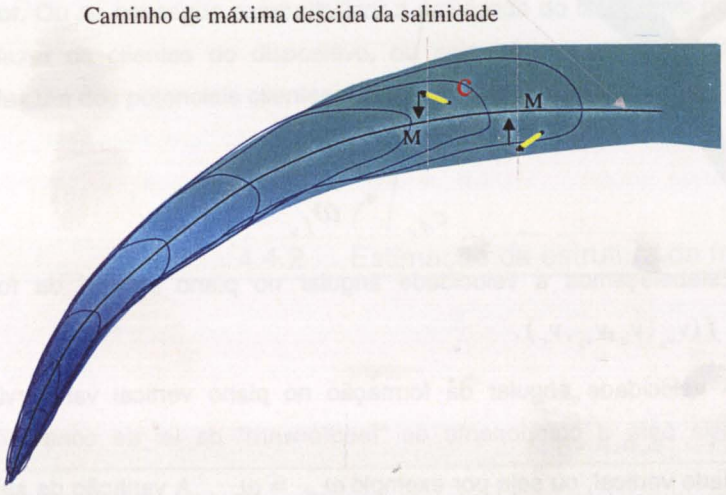
Movimento elementar, referênciar o mergulho da formação no seguimento de indícios

Esta lei de movimento destina-se ao veículo da frente da formação para ser usado na manobra mergulhar para a fonte e destina-se simplesmente a controlar o veículo no plano vertical. Quando o veículo detecta que não resultam diminuições da salinidade da progressão no plano corrente, este inicia o processo de mergulho informando (com o seu comportamento) a formação da necessidade de alterar a taxa de descida. Se, pelo contrário, o veículo verifica que com taxa de descida corrente provoca um aumento da salinidade então deve diminuir a taxa de descida. É fundamental o veículo observar a sua posição relativamente ao caminho de máxima descida da salinidade. A lei de controlo vai depender dessa posição.

Vamos definir como invariante a velocidade linear do veículo sendo esta definido na lei de controlo elementar seguir indícios.

Figura 4.3-20

Manutenção da formação no centro da pluma.



A lei de controlo vai seguir o seguinte princípio. Definamos um plano vertical, prependicular ao plano que contêm o caminho de máxima descida, que passa pelo veículo ***c.*** Se, nesse plano, o veículo está a uma profundidade superior (mais baixo) ao ponto de intercepção (ponto M na figura) do caminho com o plano vertical, então, caso a salinidade aumente, o veículo deve diminuir

a profundidade, senão continua tal como movimento vertical anterior. Se no mesmo plano o veiculo está a uma profundidade inferior (por cima) ao ponto de intercepção (ponto M na figura), então se a salinidade aumentar o veículo deve aumentar a profundidade.

Deste modo, resulta que a velocidade no plano vertical $v_v = f(\Delta salinidade, M, profundidade)$.

Como pretendemos que o veículo da frente da formação se situe numa profundidade inferior ao caminho de máxima descida, então devemos definir um pequeno valor de defeito no posicionamento vertical do veículo.

Movimento elementar, mergulhar a formação no seguimento de indícios

Esta lei de movimento elementar é utilizada pelos veículos de trás da formação e aplica-se ao movimento vertical. A variação da velocidade vertical do veículo da frente provoca uma variação da velocidade ângular no plano vertical da formação definida pelos três veículos submarinos.

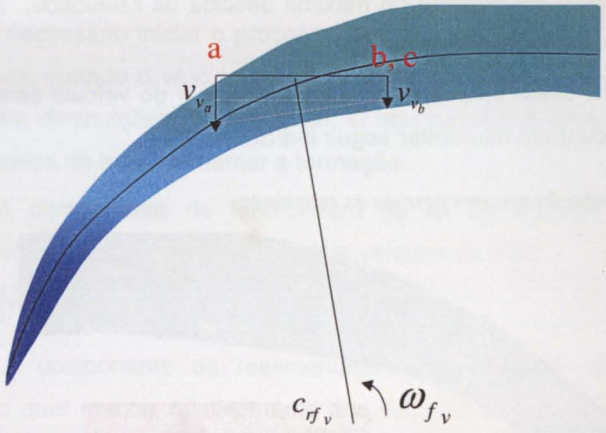


Figura 4.3-21

Determinação da taxa de descida.

Estabeleçamos a velocidade ângular no plano vertical da formação $\omega_{f_v} = f(v_{v_a}, v_a, v_{v_c}, v_c)$.

A velocidade angular da formação no plano vertical vai servir como referência para a componente de "feedforward" da lei de controlo para a velocidade vertical, ou seja por exemplo $\omega_{vff_b} = \omega_{v_f}$. A variação de salinidade vai contribuir para uma componente de realimentação. Esta lei vai seguir o mesma estratégia que foi seguida para a lei de controlo do veículo da frente na fase "mergulhar".

A componente de realimentação será $\omega_{vr_b} = f(\Delta salinidade)$.

4.4 Estimação e Controlo de Subsistemas

4.4.1 Estimação

Da mesma forma que o desempenho e a eficácia das estratégias de controlo variam consoante uma dada configuração veículo/ambiente, da mesma forma varia o desempenho as estratégias de navegação. O uso de múltiplos sistemas sensoriais, e o uso seleccionado da informação é um dos factores que pode contribuir para o maior sucesso da estimação do estado do sistema.

Recorrendo a sistemas sensores passivos e/ou relativos, podemos implementar várias leis de estimação em simultâneo (com/sem fusão) redundantes ou não. Simultaneamente a informação que resulta em diferentes etapas do processamento pode ser usada por outros recursos que não os "recursos estimadores".

Já os sistemas sensoriais activos apresentam alguma exclusividade no acesso da informação ou na parametrização da actividade do dispositivo sensor. Ou se consegue compatibilizar a actividade do dispositivo por forma a satisfazer os clientes do dispositivo, ou este só executa a estratégia que satizfaz um dos potenciais clientes.

4.4.2 Estimação da estrutura da formação

4.4.3 Recursos

Esta multiplicidade de perfis de actividade sugere que cada recurso possua um controlador de recurso, por forma a gerir a sua execução e dar resposta às questões por outros solicitada. Por exemplo um dispositivo como um sonar, um infravermelho ou uma câmara ccd pode ser utilizado como uma fonte de resultados para o sistema de localização/navegação, que por sua vez pode gerer resultados para o sistema de locomoção. Simultaneamente este

dispositivos podem ser um fonte de informação directa para o sistema de locomoção quando estamos a usar uma lei do tipo "sensor based motion".

Os requisitos impostos por cada recurso (recurso mais complexo) a cada um dispositivo (recurso elementar) poderão não ser totalmente compatíveis daí a necessidade de definir um sistema de controlo dos recursos. Se bem que um controlador de recursos, funciona como a entidade que mobiliza os "esforços", internos por forma a que o sub-sistema seja capaz de executar a actividade que lhe foi atribuída.

Então definimos:

> ***recurso*** é uma estrutura hierárquica, organizada dinamicamente, que disponibiliza um conjunto de funcionalidades ao veículo.
>
> ***Controlador de recursos*** é um controlador de estados discretos para um dado recurso, que fundamentalmente assegura a disponibilidade do recurso e a coerência dos conjunto de funcionalidades fornecidas.

Dado que o recurso é uma entidade que tem por objectivo levar a cabo uma ou varias actividades e já o definimos num quadro discreto, o interface com as entidades que assumem um papel "cliente" deverá ser definido, na semântica associada aos seguintes acontecimentos: ***inicio da actividade***, ***fim da actividade***, ***falha da actividade***. No entanto, este quadro semântico não viabiliza processos em que a dependência das actividades de dois ou mais recursos ultrapasse o sequênciamento temporal da actividade ou o sequenciamento condicional em função do sucesso da actividade correntes.

Sempre que exista conjunto de actividades que requeiram um maior sincronismo (interno, implica a necessidade de partir a actividade em segmentos, ou tornar observáveis acontecimentos não observáveis) , haverá necessidade das mesmas desenvolverem as formas de partilharem os acontecimentos sicronizadores.

4.5 Coordenação dos subsistemas

O sucesso de uma missão não está somente dependente, do sucesso de cada sistema por si. Para além da qualidade com que as manobras desembaraçam (responsabilidade da locomoção) o sistema dos sucessivos objectivos, este necessita de organizar os seus recursos por forma que estes colaborem de forma eficaz, na tomada de decisões.

A forma como os recursos do sistema desenvolvem a sua actividade, decorre num quadro de objectivos concurrentes, e muitos destes recorrem aos mesmos dispositivos para desenvolver a parte da missão que lhes está confiada.

A vista dos recursos como um autómato de estados discretos permite especificar a coordenação global do sistema como o controlo discreto do estado que resulta da composição do estado dos recursos do sistema.

Ou seja a coordenação global do sistema tem por objectivo especificar o mecanismo que permite que o domínio do espaços de estado tenha incluso no domínio que traduz a semântica da missão.

Esta página foi intencionalmente deixada em branco

PARTE II

Estado da Arte

Esta página foi intencionalmente deixada em branco

5 TRABALHO RELACIONADO

Não é objectivo deste capitulo a cobertura substancial do estado da arte em arquitecturas de controlo para robots móveis autónomos. Apenas consideramos útil apresentar alguns desenvolvimentos que, de um ponto de vista ou de outro, estão relacionadas com o nosso trabalho, permitindo não só situar a nossa abordagem ao projecto de arquitecturas de controlo, mas também ajudar a identificar a direcção de progresso do trabalho corrente.

Quando se pretende desenvolver uma monografia, somos muitas vezes tentados a classificar as abordagens em estudo. No entanto dada a juventude e dimensão da área, tal não nos parece fácil e possível, pois não existem ainda definidos e consagrados princípios metodológicos que nos permitam avaliar uma dada arquitectura, nem possuímos os mecanismos de avaliação que nos permitam clarificar se uma dada implementação corresponde a uma dada concepção. Scott Harmon, [Harmon 92], faz esta discussão em torno das diferenças que resultam das abordagens dos implementadores e dos projectistas. Contudo, neste trabalho, optou-se por não classificar as abordagens, e apresentar complementarmente a estas, ferramentas e contextos de desenvolvimento de sistemas em "tempo-real".

5.1 Arquitecturas de modelo funcional

Estas abordagens têm como denominador comum o facto de reclamarem o desenvolvimento de uma arquitectura. O objectivo é descreverem os componentes que "equipam" o sistema de todas as funcionalidades necessárias ao desempenho das tarefas perspectivadas. Fundamentalmente estas abordagens caracterizam a função, estratégia de implementação e as relações entre elas.

Para um tratamento mais detalhado de arquitecturas de controlo em geral ver [Rosenblatt 97], [Parker 00].

5.1.1 A arquitectura de modelo de referência

A arquitectura de modelo de referência, tem em vista sistemas de controlo para Máquinas Inteligentes que incluem sistemas automatizados, sistemas embebidos e robóticos. Máquinas Inteligentes são sistemas em malha fechada no sentido em que utilizam realimentação do ambiente para manifestar "Comportamento Inteligente" em tempo real. Este paradigma, exemplificado na Figura 5.1-1, poderá ser representado por arquitecturas hierárquicas tais como NASREM [Albus 88], [Albus 90], [Albus 91], [Albus 94] que forçam a modularidade e a metodologia de projecto de dos componentes em software. Nesta aproximação, o problema de controlo é particionado em:

geração de comportamentos (decomposição de tarefas);

modelização do mundo;

processamento sensorial;

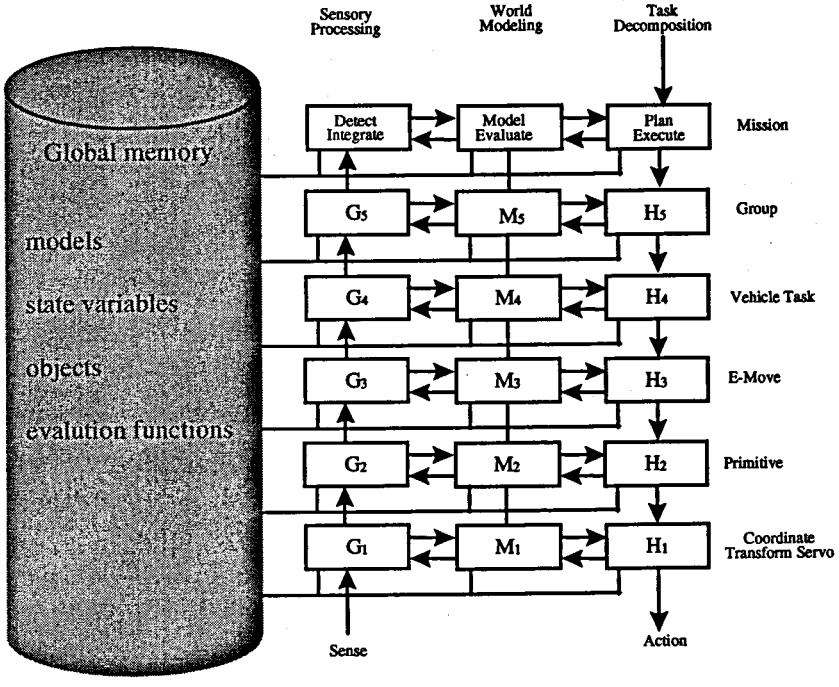
julgamento de valores.

Os vários elementos da estrutura de controlo estão agrupados em nós computacionais dispostos em camadas hierárquicas, sendo cada camada dotada com a funcionalidade de temporização característica. O paradigma de Sistema de Controlo em Tempo – Real (RCS "Real time Control System") começa com a análise de tarefas e objectivos, desenvolve algoritmos e estruturas de dados e, por fim, determina a informação sensorial necessária para efectuar as tarefas requeridas para a obtenção desses mesmos objectivos.

Este paradigma tem sido refinado ao longo do tempo. Começando com o computador de modelo aritmético CEREBELAR nos anos 70, evoluiu para ao RCS-4 nos anos 90, tendo sido aplicadas arquitecturas de complexidade intermediária para Automação de Manufacturas, Transporte de Campo Militar, Múltiplos Veículos Submarinos e um Sistema Telerobótico de Voo.

Figura 5.1-1

Arquitectura hierárquica NIST.



A metodologia associada com a arquitectura RCS endereça explicitamente o projecto dos seguinte componentes:

- modos de execução de software e tarefas de comportamentos em tempo real;
- interfaces e métodos de comunicação entre módulos de software, recursos de hardware e operadores humanos;
- modelos de informação e gestão de bases de dados;
- alocação de recursos; e
- regras para a decomposição espacial e temporal do projecto.

5.1.2 A arquitectura ATLANTIS

A arquitectura ATLANTIS ("Architecture for Navigating Through Intricate Situations" introduzida por [Gat 92] combina planeamento simbólico com módulos reactivos sensoriais e motores. Baseia-se no conceito de

"planeamento-como-comunicação" proposto por [Agre 90] em oposição ao conceito tradicional de planeamento como programa ou procedimento. No conceito utilizado o plano realizado é visto como um recurso que orienta a actuação em vez de a controlar directamente. A ideia é aproximar o planeamento aos objectivos (do mesmo modo utilizado pelo homem) em vez de uma definição exacta e precisa de sequência de tarefas.

Esta arquitectura encontra-se dividida em três camadas: um módulo deliberador em Lisp, um sequenciador e um controlador reactivo.

O deliberador toma decisões assincronamente gerando um plano acessível pelo sequenciador. Este pode determinar o início e fim do planeamento ou o replaneamento se o plano se tornar inadequado (tal como na camada de coordenação proposta na arquitectura de hierárquica de três níveis [Saridis 89]). O sequenciador estabelece o interface com o controlador reactivo determinando o início e o fim dos comportamentos adequados.

O controlador é constituído por um conjunto de comportamentos responsáveis pela leitura dos sensores e pela determinação da actuação reactiva.

Esta arquitectura foi aplicada em diversos robots desenvolvidos pelo JPL ("Jet Propulsion Laboratory") e nomeadamente no Mars Rover. Para melhorar as características de fiabilidade e lidar com problemas de cognição detectados, Gat propôs [Gat 94] a implementação de planos de recuperação de contingência para situações de falha detectadas por experiência prévia.

5.1.3 A arquitectura DAMN

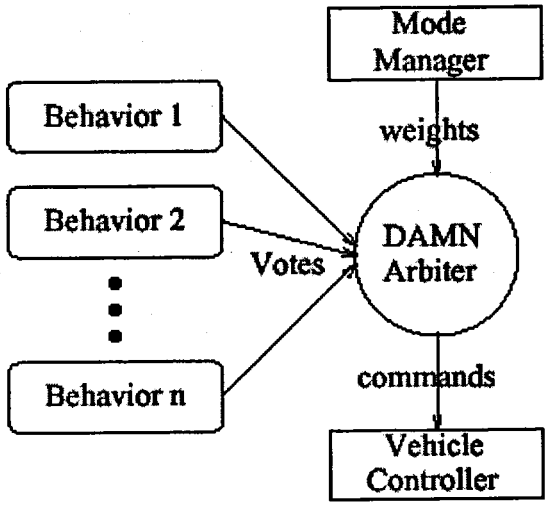
A arquitectura DAMN ("Distributed Architecture for Mobile Navigation") [Rosenblatt 95], [Rosenblatt97], [Payton 90] consiste num módulo árbitro centralizado e num conjunto de módulos distribuídos (comportamentos) que produzem acções.

O árbitro recebe votos a favor ou contra as acções de cada módulo. Cada comportamento vota a sua própria preferência no conjunto de acções possíveis e a fusão comportamental é efectuada pelo árbitro.

Esta fusão é efectuada recorrendo a formalismos de lógica difusa evidenciando algum paralelismo com a arquitectura baseada em comportamentos difusos descrita em [Saffioti 95].

Figura 5.1-2

A arquitectura comportamental DAMN.



Existem componentes de natureza deliberativa que efectuam o seu voto de forma semelhante a cada outro módulo. Desta forma não existe um plano definido como procedimento de acordo com os princípios definidos em [Agre 90] mas, os comportamentos reactivos são influenciados (pela composição na votação) pelos módulos deliberativos permitindo a não existência desta forma uma estrutura hierárquica rígida.

A natureza distribuída da arquitectura fornece-lhe uma boa tolerância a falhas sendo o módulo arbitral central o ponto crítico neste aspecto.

Esta arquitectura foi aplicada com sucesso em vários robots desenvolvidos pela CMU ("Carnegie Mellon University"), com particular destaque no projecto Navlab, no qual um robot "outdoor" viajou de Pittsburg até Los Angeles (EUA) autonomamente em 95% do tempo de percurso.

5.1.4 A arquitectura hierárquica de controlo inteligente.

Saridis e os seus colaboradores têm vindo a desenvolver uma teoria consistente para Máquinas Inteligentes baseada na teoria de controlo hierárquico inteligente [Saridis 84], [Saridis 85], [Saridis 89], formalmente suportada por uma linguagem de redes de Petri. Tal como é descrito na Figura 5.1-3. o quadro proposto envolve três níveis principais: organização; coordenação; e execução. Uma característica distintiva desta metodologia surge da noção de decomposição de tarefas em funções primitivas.

Ao plano da missão, a camada de organização (o organizador de tarefas) gera tarefas de alto nível para o sistema Robótico Móvel Inteligente executar.

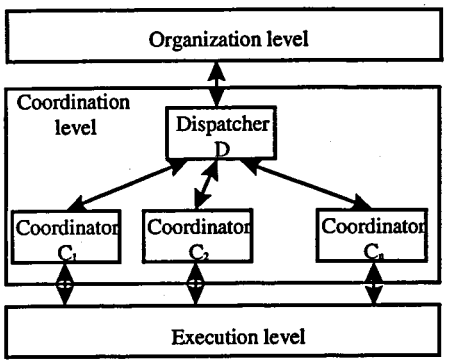


Figura 5.1-3

A estrutura hierárquica de uma Máquina Inteligente.

O nível de coordenação [Wang90], actua como um interface entre a organização e o nível de execução. Este nível consiste num DISPATCHER e vários coordenadores de tarefas. O DISPATCHER decompõe as tarefas do plano da missão geradas pelo organizador em acções de controlo orientadas para coordenação e fornece-as para os coordenadores correspondentes nos tempos apropriados. Os coordenadores traduzem comandos de controlo em instruções de operação que são carregadas para o nível de execução e tornadas disponíveis aos correspondentes mecanismos de execução.

Na estrutura de coordenação proposta, lugares representam processos e transições representam algoritmos. Uma vez que modelos de redes de Petri são incapazes de descrever a tradução da linguagem tal como o requerido para a implementação de esquemas de decisão linguística [Saridis 84], são introduzidas Tradutores de Redes de Petri [Wang 91]. Estes tradutores de linguagem transformam um dado plano de tarefas (de entrada) noutro com um diferente nível de detalhe (de saída).

O nível de execução opera em tempo real. Executa as instruções vindas do nível de coordenação e fornece informação de volta para este.

O modelo analítico subjacente à estrutura de controlo hierárquica baseada no principio de "Increasing Precision with Decreasing Intelligence" tem sido formulada num quadro probabilístico onde incerteza e imprecisão são expressas em termos de entropias. A estratégia óptima para o planeamento de decisões e execução de tarefas é a que minimiza a entropia total do sistema. Desta forma, podem ser formulados problemas de controlo óptimo para endereçar questões de projecto de máquinas inteligentes. Desenvolvimentos recentes deste quadro analítico envolvem a definição de uma metodologia suportada na teoria da complexidade baseada em Informação para medição

conjunta de custo e fiabilidade de algoritmos [Lima 95]. Esta aproximação melhora ainda a base para o projecto analítico de máquinas inteligentes.

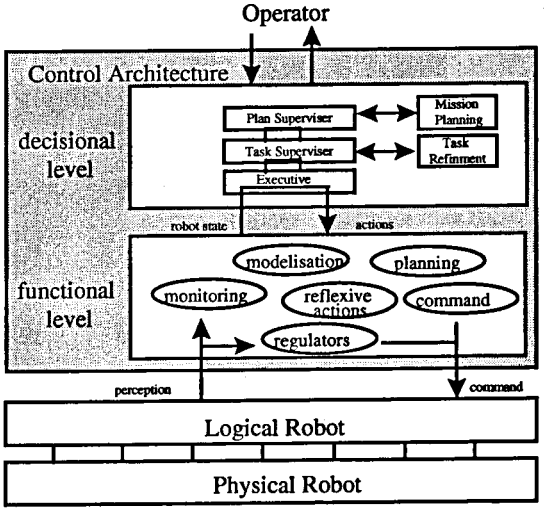
5.1.5 A arquitectura desenvolvida no LAAS

Muito do trabalho tem sido desenvolvido no LAAS na área de projecto de plataformas móveis autónomas [Chatila92], [Chatila95], [Giralt79], [Giralt83], [Noreils95], [Fleury96], [Alami91]. Apesar da construção de vários robots, vamos concentrar a nossa atenção no trabalho conceptual. Critérios gerais para o projecto de uma estrutura de controlo de um protótipo experimental de uma máquina autónoma incluem reactividade, flexibilidade, reconfigurabilidade, autonomia e programabilidade.

A arquitectura proposta é organizada em três níveis correspondendo a dois níveis de decisão acima do nível funcional. Os dois níveis mais altos são construídos de acordo com o paradigma de supervisor/planeador. Enquanto que o nível mais alto usa um planeador temporal, o segundo nível recebe tarefas e transforma-as em procedimentos compostos por acções elementares e supervisiona a execução destes procedimentos num contexto de reacção a acontecimentos assíncronos. Neste nível, o planeamento consiste na refinação da tarefa usando conhecimento específico sobre o domínio.

Figura 5.1-4

A arquitectura hierárquica LAAS.



No que diz respeito ao nível de planeamento de missão, foi desenvolvido um planeador temporal IxTeT para "raciocinar" sobre relações numéricas ou simbólicas temporais. O plano produzido pelo IxTeT consiste num conjunto de tarefas parcialmente ordenado associado com as restrições temporais de forma

que limites mínimos e máximos na duração de execução são definidos e tomados em conta na sincronização com acontecimentos externos.

As tarefas de refinamento e supervisão são integradas através de um Sistema de Raciocínio Procedimental que oferece um contexto bem adaptado ao suporte da interacção entre decisão e reacção.

A natureza do papel do nível de execução é puramente reactiva sem qualquer planeamento. Controla a execução de acções de acordo com esquemas predefinidos seleccionados de acordo com o contexto. Esta implementação é baseada num conjunto de regras que permite a compilação "off-line" do conjunto produzindo um programa para procurar uma arvore de decisão em tempo limitado.

Uma boa descrição do trabalho conceptual desenvolvido sobre a arquitectura (ver Figura 5.1-4) é apresentada em [Noreils 95], [Fleury 96]. Em [Chatila 95] e [Alami 91] é descrita uma arquitectura genérica para robots, com particular ênfase no nível de decisão. Aqui, a supervisão é implementada como um módulo de "raciocínio" procedimental adaptado aos requisitos de tempo real associados com a execução das funcionalidades que são necessárias com qualquer actividade.

5.1.6 A arquitectura de "subsumption"

Como uma alternativa às arquitecturas hierárquicas temos a "subsumption" de Brooks ([Brooks 86a], [Brooks 86b]) que evita o uso de um modelo explícito do mundo e em vez disso implementa comportamentos primitivos e complexos através de um relacionamento mais directo entre percepção e acção. Esta estrutura foi inspirada por estudos no campo da etimologia sobre comportamento animal.

Apesar de menos previsível em tarefas de rotina, esta arquitectura parece ser mais adequada para reagir a acontecimentos inesperados. Brooks é de opinião que "o mundo é o seu melhor modelo" (*"world is its own best model"*) e de que acções complexas e aparentemente úteis podem surgir da competição entre comportamentos ordenados de acordo com um esquema predeterminado de prioridades, sem recurso a um modelo interno do mundo. Neste contexto, não existe planeamento nem acções explícitas de controlo.

Neste tipo de arquitectura, as reacções são organizadas em níveis hierárquicos onde cada nível corresponde a um conjunto de comportamentos

possíveis. Um comportamento particular é requerido sob o efeito de um estímulo interno ou externo. Após ser chamado gera uma comunicação com o nível inferior. Neste nível, surge outro comportamento como resultado da combinação simultânea da comunicação do nível superior e dos outros estímulos. O processo continua até que sejam activados comportamentos terminais. Cada nível implementa um comportamento único que pode inibir entradas e saídas do nível imediatamente inferior. Comportamentos a níveis mais altos de abstracção dizem-se emergentes face aos de níveis inferiores. Brooks argumenta que as interacções entre estes comportamentos permitem atingir comportamentos mais complexos.

5.1.7 A arquitectura Fate e KAMARA

KAMARA é uma arquitectura de controlo distribuída que evoluiu da arquitectura FATE [Laengle 97], [Lueth 94]. A arquitectura FATE consiste num nível de planeamento tipo "blackboard" para gerar operações elementares específicas de manipuladores dependentes do contexto. Estas são executadas por controladores robóticos em tempo real (RT-Cs). O controlador em tempo real é capaz de controlar manipuladores quer independentemente ou integrados numa cadeia cinemática fechada.

A nova arquitectura KAMARA usa um algoritmo recursivo para trocar tarefas a um nível de célula (cooperação entre robots) e ao nível de robot (cooperação entre componentes de um robot) e permite uma coordenação e escalonamento distribuído das actividades. Esta arquitectura pode ser formalizada numa estrutura multi-agente onde cada agente é composto pelos seguintes componentes: Comunicador; Planeador/Seleccionador de Acções e; Executor. É útil referir que a comunicação entre vários agentes pode ocorrer síncrona ou assincronamente. Uma vez que usa vários componentes da arquitectura centralizada FATE, a execução de uma operação elementar em ambos os sistemas é efectuada por sistemas de controlo robótico RT- Cs.

5.1.8 A arquitectura AuRA

A arquitectura AuRA ("Autonomous Robot Architecture") [Arkin 87], [Arkin 97] baseia-se no conceito de esquemas originado em psicologia e neurologia.

De certa forma semelhante com o conceito de comportamentos presente na arquitectura "subsumption" protagonizada por Brooks, o esquema no presente contexto consiste as primitivas necessárias para a realização das actividades de percepção e motoras. O esquema produz um dado comportamento e é simultâneamente um repositório de informação e uma descrição do processo de aplicação da mesma.

A actividade motora é determinada por esquemas motores que definem o comportamento como o robot responde a estímulos externos. A percepção do ambiente é definida por esquemas de percepção que fornecem informação aos esquemas motores.

Adicionalmente são definidos esquemas de sinal, que são responsáveis por monitorizar o estado interno do robot e modificar o seu comportamento de acordo com esse mesmo estado.

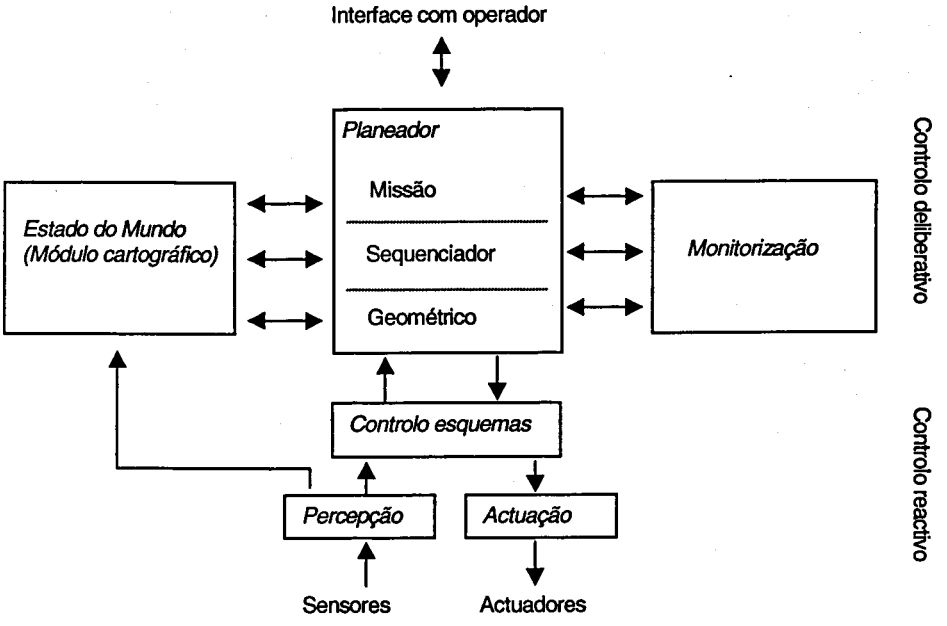


Figura 5.1-5

A arquitectura híbrida AuRA.

Esta arquitectura é constituída por cinco subsistemas principais: Percepção, Subsistema motor ou de actuação, Cartográfico, Planeador e de Monitorização.

O subsistema cartográfico consiste na descrição do conhecimento do mundo, contendo para além dos modelos conhecidos à priori, informação adquirida em tempo real bem como a descrição de incerteza associada. Os subsistemas de Percepção e Motores efectuam o interface com os sensores e

os actuadores do robot, permitindo a aquisição e filtragem de dados sensoriais bem como a acção nos diversos actuadores do sistema.

O Planeador é constituído em três níveis hierárquicos (Missão, Planeamento Geométrico, Sequenciador). A missão é decomposta num conjunto de trajectos espaciais pelo nível geométrico utilizando a informação de navegação global (Subsistema Cartográfico) sendo estes traduzidos num dado número de esquemas motores pelo sequenciador. Estes esquemas motores ao definirem o comportamento motor do veículo constituem também parte do planeamento pelo que esta arquitectura possuí características híbridas quer deliberativas quer reactivas.

Esta arquitectura tem sido utilizada quer em robots móveis terrestres desenvolvidos no Georgia Institute of Technology quer com variantes em veículos submarinos autónomos como o caso do EAVE ou Orca.

5.1.9 Arquitectura por camadas e controlo híbrido

Em [Varaya 97] é apresentada uma arquitectura por camadas para um sistema de auto-estradas automáticas.

Esta arquitectura tem como base do seu desenvolvimento a aplicação no controlo de veículos em auto-estradas inteligentes desenvolvidas no âmbito to programa PATH ("Partners for Advanced Transit and Highways") do estado da Califórnia.

O controlo é distribuído por várias camadas organizadas hierarquicamente com agregação crescente da informação transmitida de cada camada inferior para uma camada superior. Em [Varaya 97] enunciam-se regras necessárias para que se possa analisar, implementar e validar cada camada de forma isolada. Adicionalmente propõem-se princípios que devem orientar a definição de uma arquitectura de controlo. Esta deve permitir o crescimento incremental do sistema, a sua distribuição heterogénea, a robustez e recuperação de falhas. Reconhecendo a dificuldade prática da aplicação do mesmo corpo teórico a todas as camadas, preconiza-se como vector de orientação a maximização da consistência entre as teorias utilizadas em diferentes camadas.

A arquitectura de controlo implementada baseia-se na definição hierárquica de sistemas híbridos de controlo [Lygeros 98], tendo sido utilizada

noutras aplicações em robótica como no controlo de veículos aéreos autónomos (UAV – "Unmanned Aerial Vehicle") [Sastry 99].

Esta implementação possuí 4 níveis hierárquicos: Regulação, Planeamento de Trajectórias, Planeamento Táctico e Planeamento Estratégico. O nível de regulação implementa leis de controlo de baixo nível em tempo contínuo e fornece informação ao nível superior (planeamento de trajectórias) sob a forma de erros de seguimento. Este nível é responsável pela geração da trajectória a seguir pelo veículo recebe do planeamento de táctico os modos de controlo em que o veiculo se encontra. Estes modos ou manobras parametrizam as trajectórias a efectuar e a sua comutação depende da detecção de eventos relevantes por parte de um módulo detector ou observador. A missão global é determinada pelo planeador estratégico, fornecendo os pontos de controlo que definem a execução da mesma.

5.1.10 Arquitecturas baseadas em Redes Neuronais

A capacidade das redes neuronais de lidarem com a incerteza e de resolverem problemas de controlo sem necessitarem de um modelo explícito do ambiente torna-as atractivas como soluções para o controlo de sistemas robóticos e em particular no caso de veículos submarinos autónomos, [Yuh 90] ,[Venugopal 92].

O fundamento neurológico destas metodologias torna também relevante a sua aplicação no desenvolvimento de sistemas robóticos tendo em vista a mímica da vida. Destes, o projecto de um insecto artificial preconizado por Berr [Beer 90], influenciou a aplicação de redes neuronais para o controlo de um robot hexápode [Beer 92].

De igual forma foram utilizadas redes neuronais recorrentes e "feedforward" como componentes básicos da arquitectura utilizada no robot hexápode LAURON na Universidade de Karlsruhe [Berns 95]. Esta encontra-se organizada hierarquicamente em três camadas. Existe um Elemento Reactivo a um nível superior que avalia a informação sensorial e selecciona a melhor coordenação das pernas. A camada de coordenação de perna que gera os padrões de passo ("gait") e os controladores de cada perna que num nível inferior são responsáveis pelo controlo de movimento da mesma.

A utilização de redes neuronais embora apresente boas características de robustez tem como um dos seus principais problemas o tempo necessário para o treino bem como a necessidade de uma elevada dimensão para poder exibir comportamentos complexos de interesse.

5.1.11 Arquitectura hierárquica da NPS

O desenvolvimento de veículos submarinos autónomos na Naval Postgraduate School permitiu o desenvolvimento de uma arquitectura hierárquica em três níveis [Healey 96], [Marco 96]. Esta baseia-se nos conceitos enunciados por Saridis e espelha na sua definição o modo organizacional de operação na instituição militar a que não é de estranhar o facto de a marinha americana ser o principal interessado nos veículos em desenvolvimento.

A arquitectura divide-se nos níveis de Execução, Táctico e Estratégico. O nível inferior actua no domínio numérico e em tempo contínuo. Corresponde ao nível de operação em termos militares e desta forma implementa as leis de controlo que determinam de forma imediata as acções nos actuadores e a recolha da informação proveniente dos sensores. O nível táctico intermédio efectua o interface entre a informação de carácter numérico do nível inferior e a de carácter simbólico do nível superior. Corresponde ao planeamento táctico na organização militar e como tal é responsável por determinar quais as operações ("leis de controlo") adequadas à execução da estratégia superiormente definida.

O planeamento estratégico é efectuado no nível superior e actua nos domínios simbólicos e discretos, determinando a missão a ser executada. A sua implementação utiliza um sistema de regras (em Prolog) permitindo a definição dos objectivos da missão de uma forma declarativa. Este aspecto particular da implementação revelou-se de particular interesse pela facilidade a que permite a utilização e operação dos veículos submarinos autónomos por parte da comunidade militar.

5.2 Contextos de desenvolvimento e linguagens de "tempo-real"

Neste ponto vamos descrever ferramentas para o desenvolvimento de sistemas de "tempo-real", no contexto da robótica móvel. Fundamentalmente, estas abordagens preocupam-se com as regras de "convivência" que todos os subsistemas devem adoptar no sentido do sistema responder aos requisitos. Estas regras passam por definições de sincronização, formas de comunicação e a satisfação de restrições temporais tais como a duração e o tempo de resposta

5.2.1 ControlShell

ControlShell [Schneider 95] é um sistema muito leve e reconfigurável, desenvolvido por forma a permitir um contexto de integração de sistemas em tempo real, e , que considera a validação obtida a partir da experimentação em oposição à verificação formal do comportamento global do sistema. Fornece:

um conjunto de ferramentas para construção de componentes partilháveis e especificar as suas interacções;

uma máquina de estados finita, que implementa um sistema executivo programável para a supervisão da execução das actividades dos vários componentes.

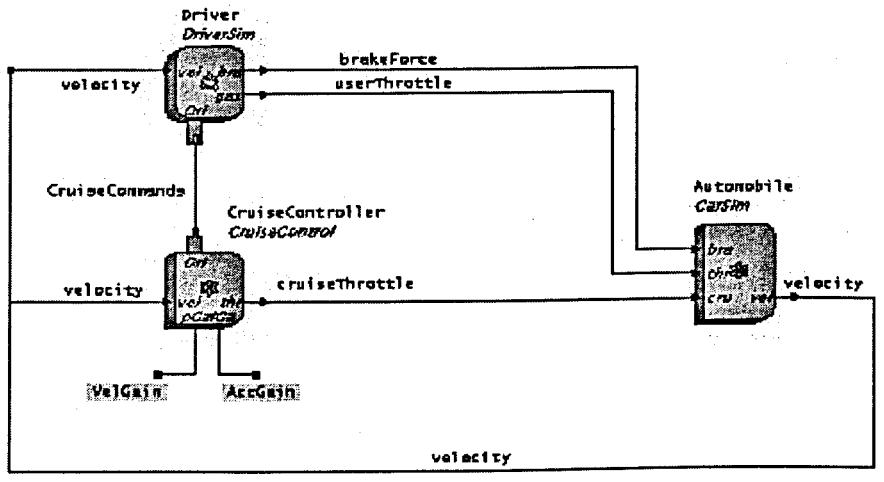


Figura 5.2-1

Componentes em ControlShell para o controlo de velocidade de um veículo.

Os elementos básicos são organizados numa biblioteca de módulos expansíveis e reutilizáveis que podem ser articulados através de um editor

gráfico que permite a especificação do fluxo de dados ao nível do componente. Podem-se programar estratégias para activação/desactivação dos módulos na forma de um gráfico de estados que é implementado por um modelo de transição de estados.

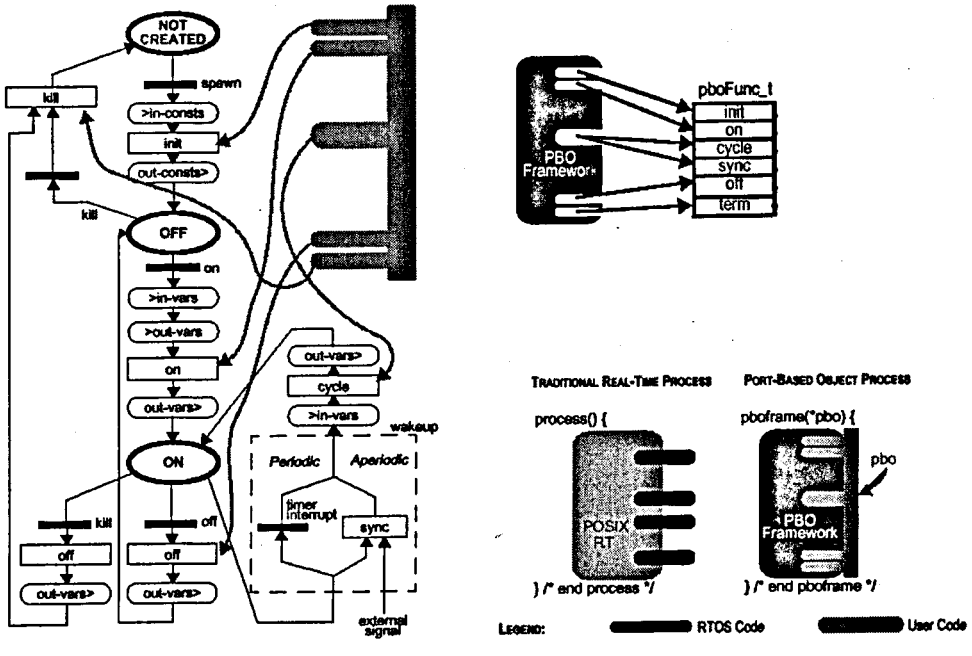
No sentido de integrar todo o processo de desenvolvimento de software, esta ferramenta evoluiu com o acréscimo de novas funcionalidades [RTI 02] como a utilização de diagramas UML ("Uniform Modeling Language") ou o uso do modelo CORBA para comunicação entre componentes.

5.2.2 Chimera

Chimera [Stewart 96], [Stewart 00] é um ambiente de integração modular para sistemas em tempo-real. Este utiliza o modelo PBO ("Port Based Objects") que permite a construção de sistemas dinamicamente reconfiguráveis. Os componentes são definidos como instâncias de um objecto genérico que inclui um estado interno, um interface constituído por portos entrada-saída e um método que encapsula dados e recursos. Os portos permitem a comunicação entre vários objectos.

Figura 5.2-2

Constituição interna do modelo PBO utilizado no Chimera



Cada objecto têm quatro estados: NOT-CREATED , ON, OFF e ERROR. Quando activado, cada objecto origina uma actividade que é executada independentemente das dos outros objectos. A execução de uma actividade gera ou requer dados de e para uma área de memória local que é exportada ou importada de uma memória comum.

5.2.3 Orccad

Vários métodos genéricos para o controlo em tempo-real e para a verificação da consistência têm sido desenvolvidos em algumas instituições (exemplo: ver [Henzinger93], [Henzinger94], [Ravn93], [Samson90], [Alur93], [Espiau95], [Yorger86]). Em particular, no INRIA, algum trabalho tem focado o desenvolvimento de uma infra-estrutura geral para suportar o projecto de arquitecturas dedicadas para sistemas de controlo de veículos móveis autónomos. Foi tomada especial atenção aos problemas decorrentes da utilização dos sistemas em meios subaquáticos onde a adaptabilidade em tempo real como, por exemplo, o auto-replaneamento e o controlo sensorial detêm um papel importante. Parte deste trabalho foi utilizado na implementação de um veículo subaquático (o Vortex), no IFREMER, [Simon93], [Simon95]. Em [Espiau95], são aplicados métodos de verificação formal a vários problemas de robótica em ORCCAD, sendo este um ambiente de desenvolvimento para controladores robóticos desenvolvido no INRIA. Este ambiente fornece um conjunto de ferramentas para verificar antes do início da missão se o veículo irá efectuar correctamente as tarefas previstas. A verificação da consistência lógica e a satisfação das restrições temporais permite concluir se:

- as especificações estão correctas;
- programa corresponde às especificações; e
- restrições de implementação em tempo-real não perturbam o comportamento.

Neste ambiente [Simon93], a tarefa Robot Task, RT, é a entidade principal que está associada com um comportamento local controlado por um conjunto de observadores e modelizado por um autómato de estados finito. É composta por um conjunto de tarefas elementares comunicantes em tempo real, as tarefas dos módulos cuja especificação requer restrições temporais e de sincronização explícitas. A RT é composta por uma tarefa elementar de

servo-controlo e um módulo lógico instanciado num autómato que gere o comportamento global. O módulo, MT, é a entidade básica que implementa cada processo elementar da lei de controlo e desta forma lida com todas as questões temporais. No que diz respeito a ferramentas de verificação, o ambiente ORCCAD inclui um "interface" gráfico pessoa-máquina amigável e software de simulação.

5.2.4 Matlab – Real Time Workshop

O ambiente de calculo matemático Matlab tem uma vasta implantação na comunidade de controlo como ferramenta de análise e teste de soluções. A sua utilização estendeu-se à simulação de sistemas com o pacote Simulink, que permite a simulação de sistemas contínuos ou discretizados de uma forma gráfica recorrendo aos tradicionais diagramas de blocos. Dado o vasto número de utilizadores, bem como a sua facilidade de uso, tornou-se relevante a consideração da possibilidade de geração de código executável. Tal veio a tornar-se realidade com a inclusão da ferramenta "Real-Time Workshop" [Mathworks 97] que permite gerar código a partir de um diagrama Simulink.

Foi também introduzida uma ferramenta adicional: o Stateflow que permite definir a supervisão de sistemas. Esta baseia-se no formalismo de "statecharts" definido por [Harel 87] que constituí a base da maior parte dos formalismos baseados em diagramas de estados incluindo os de UML. O Stateflow permite definir estados discretos que evoluem consoante transições determinadas pelos fluxos de dados existentes nos diagramas Simulink associados. Desta forma, é efectuada a ponte entre o domínio contínuo e quantitativo modelizado no Simulink e o domínio discreto e simbólico modelizado no Stateflow.

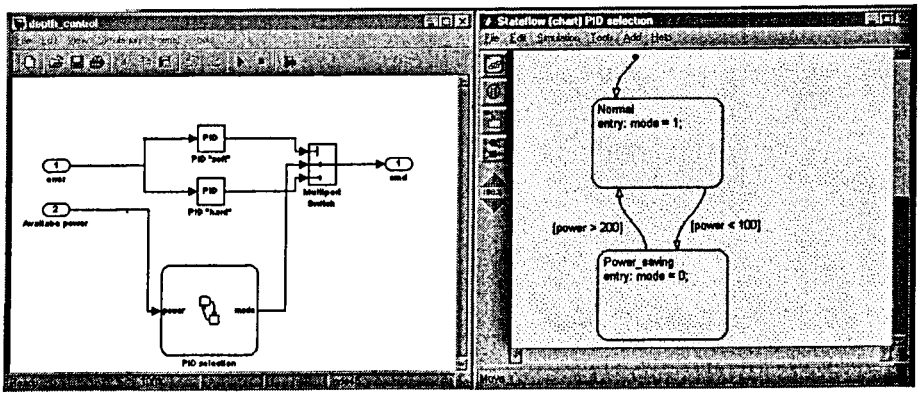


Figura 5.2-3

Exemplo de um modelo em Simulink e Stateflow

A geração automática de código associada a estas duas ferramentas (por via do RTW) encontra-se ainda a alguma distância face à sua implementação imediata uma vez que em muitos casos não tem em consideração aspectos particulares de implementação de software como as questões relacionadas com o sistema operativo, nem a definição de uma metodologia clara de integração com software previamente existente.

5.2.5 TEJA

A ferramenta Teja [Deshpande 97] fornece uma estrutura de software orientada aos objectos para a implementação de sistemas de controlo. Possui como modelo de desenvolvimento o de uma rede dinâmica de autómatos híbridos. Permite a existência de eventos síncronos ou assíncronos bem como a reconfiguração dinâmica da propagação de eventos entre os componentes permitindo desta forma a reconfigurabilidade da rede.

Um componente sendo um autómato híbrido, possuí estados discretos nos quais é necessário definir as equações de fluxo (e logo a evolução do estado contínuo) bem como a definição da recepção, processamento e emissão de dados e de eventos (que são implementados em métodos da classe).

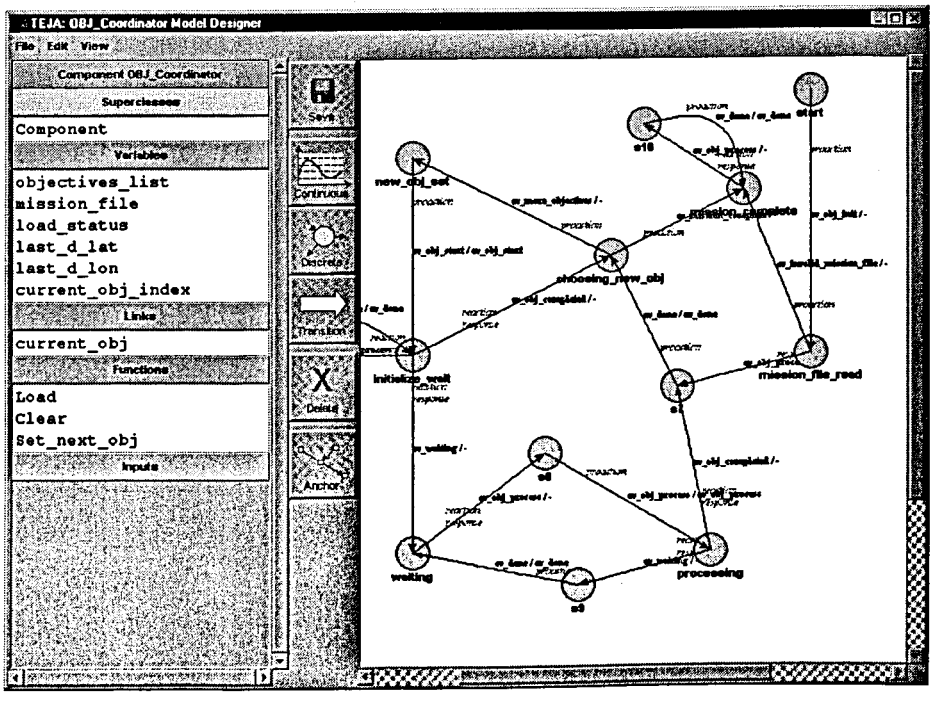


Figura 5.2-4

Interface gráfico da ferramenta TEJA.

A capacidade de aliar as virtudes da programação orientada aos objectos com a caracterização do sistema de controlo em termos de autómatos híbridos constituí uma das suas mais-valias significativas no desenvolvimento da aplicação.

O código gerando é ligado a um conjunto de classes base e um escalonador fornecidos, sendo construída uma aplicação monolítica. Esta aplicação pode correr num dos sistemas operativos definidos (Linux, SunOs e QNX). No entanto ao utilizar o seu próprio escalonador e ao produzir uma única aplicação, o interface do código desenvolvido quer com o sistema operativo base quer com outros módulos previamente desenvolvidos é de dificil implementação. Outro dos problemas evidenciados por esta ferramenta é a fraca qualidade do interface gráfico de desenvolvimento.

Esta página foi intencionalmente deixada em branco.

PARTE III

Arquitectura para um Sistema Autónomo

Esta página foi intencionalmente deixada em branco

6 ABORDAGEM AO PROJECTO DE SISTEMAS AUTÓNOMOS

6.1 Introdução

A definição (ou esboço) de uma arquitectura é a componente principal da resposta à questão, que com grande frequência nos deparamos, e que é, a seguinte: "Como é que se organiza o sistema de controlo de um sistema complexo?" A resposta a esta questão assenta em parte na insuficiência de metodologias formais para o projecto e implementação do sistema de controlo. Caso existisse uma metodologia (fechada) que nos conduzisse desde a especificação (passando pelo projecto) à implementação, a classe de arquitecturas seriam uma consequência da aplicação da metodologia ao problema. Como tal não é o caso, a arquitectura surge como a fonte da orientação para a organização sistémica (apresenta as linhas orientadoras) e não apenas como consequência do perfil organizacional do sistema.

É de salientar que a ideia de "arquitectura do sistema de controlo" aparece comummente associado à organização de funcionalidades e à organização dos componentes de hardware ou de software, sem existir mecanismos para mapear a relação entre estas diferentes realidades. É também comum surgir na mesma estrutura organizacional, referências a diferentes realidades [Fleury 96], [Rosenblatt 97], [Laengle 97], não se diferenciado o que implementa o quê.

Arquitectura que vamos apresentar e que serviu de referência aos projectos endereçados nos capítulos anteriores, foi sendo refinada ao longo do desenvolvimento dos mesmos não é imune às dificuldades inerentes da passagem do plano conceptual à implementação. Para além do refinamento que sofreu devido à análise da instância implementadas e ao processo de projecto, beneficiou da dialéctica que se desenvolve em torno do binómio arquitectura – metodologia. Neste estágio de desenvolvimento, não é claro o que é causa ou consequência, se a arquitectura ou se a metodologia. No entanto, é claro para nós, que esta espiral de desenvolvimento leva a uma "escola" de desenvolvimento de sistemas autónomos bem característica [Sousa 94], [Silva 94], [Sousa 95], [Sousa 96], [Po-Robot 99a], [Po-Robot 99b], [Silva 99], [Martins 99], [Sousa 01], [Martins 01], [Martins 02].

6.2 Sistema autónomo

Façamos aqui algumas definições preliminares que nos ajudarão a manter a consistência das linhas metodológicas por nós propostas neste trabalho.

Sistema autónomo é um sistema em que não existe défice algum entre a capacidade de percepção gerada pelo próprio e aquela que é necessária para a sua actuação. Significa isto que os requisitos comportamentais do sistema controlado através da realimentação apenas no estado e na informação adquirida pelo sistema – ou seja, sem incluir sinais exógenos – é suficiente para garantir que o estado do sistema evolui de acordo com os requisitos especificados na concepção do sistema.

Vejamos agora uma definição mais formal.

Consideremos:

x - estado;

i - informação adquirida pelo sistema;

u - controlo;

e - indicadores externos impostos ao sistema.

Seja a dinâmica do sistema definida como

$x_{t'} = f(x_t, u_{t'}, i_t)$ em que genericamente $u_t = g(x_t, i_t, e_\tau) \forall t \leq t'$

Um sistema autónomo é sistema tal que $\forall t' > t$ preserva as propriedades $P[x_{t'}]$ com $u_t = g(x_t, i_t)$, ou seja u_t não depende de e_τ.

Para enquadrar este conceito de autonomia do ponto de vista de controlo vamos definir a organização e os componentes conceptuais de um sistema autónomo. A definição destes componentes é muito importante pois será ao longo destes que faremos o desenvolvimento do projecto.

Recorrendo a diferentes mecanismos de síntese, o subsistema de decisão gera formas de actuação consistentes em diferentes horizontes temporais. Estas incluem desde a reactividade até ao planeamento de actividades (reactividade a planeamento). Esta capacidade de decisão consistente pode ser considerada em domínios temporais (contínuos e discretos) distintos.

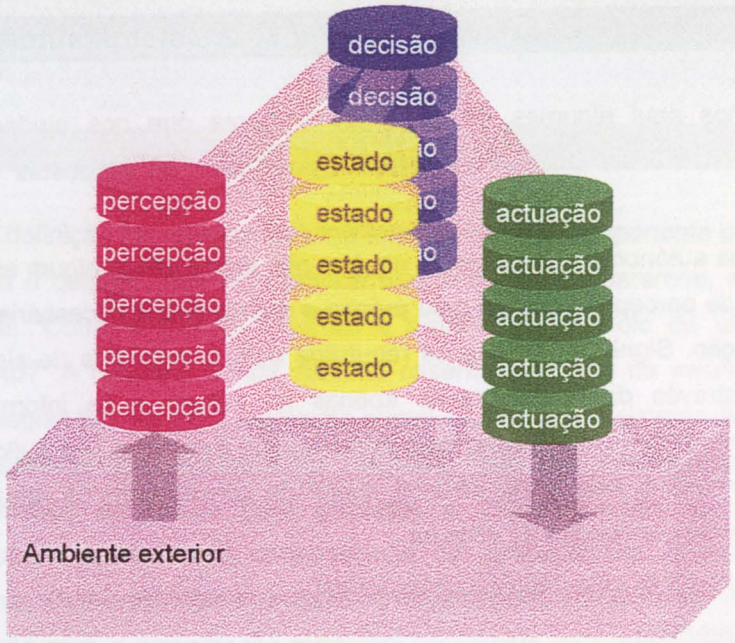


Figura 6.2-1

Componentes básicos de um sistema autónomo:

Decisão;

Percepção;

Estado ou consciência;

Actuação.

O subsistema de percepção permite ao sistema "ler" o mundo exterior e construir "cenários" prontos a ser interpretados pelo sistema de decisão. No nosso conceito a percepção vai desde a capacidade da leitura de um sinal eléctrico até a estimação de um valor utilizando um filtro de Kalman.

O subsistema estado ou consciência permite ao sistema manter a estrutura de informação em vários níveis semânticos, projectar em instantes de tempo futuros, bem como manter activamente observações consistentes com as estimações.

O subsistema de actuação recruta os recursos necessários nos diferentes níveis de abstracção para tornar efectiva a saída do sistema de decisão no que respeita à promoção da alteração das variáveis do mundo exterior que controla através dos actuadores bem como fazer evoluir o estado da "consciência".

A complexidade da aplicação dos conceitos conceptuais gerais apresentados aumenta substancialmente quando, no processo de projecto e especificação, caminhamos na direcção da definição de componentes implementáveis, ou quando pretendemos definir os anéis de controlo do nosso sistema bem como a sua organização hierárquica.

Assim, neste processo, a estrutura organizacional vai se assemelhando a uma teia, ou a uma estrutura cristalografia de hierarquias relativas.

6.3 O projecto de sistemas autónomos

6.3.1 Metodologias de Projecto

No contexto actual ainda não existem desenvolvimentos teóricos e tecnológicos para que, através de uma especificação formal de objectivos e propriedades (físicas, lógicas, operativas, temporais), possamos obter de forma automática o projecto, ou o objecto (sistema autónomo) que é solução da especificação. Contudo, enquanto não se estabelecem essas metodologias, podemos identificar um grande conjunto de "processos rotineiros" que ajudam na realização "à mão" do projecto de sistemas autónomos. A vastidão dos tópicos e o conjunto de problemas ainda em aberto (exemplo: síntese de controladores, verificação, navegação, sensores actuadores, arquitecturas computacionais, RTOS, etc.) associados a este processo construtivo, é ainda um obstáculo para que a vertente "de como fazer", apareça como uma abordagem integrada, surgindo quase sempre limitada ao tópico teórico ou tecnológico objecto de investigação. No entanto, os grupos de investigação têm vindo a focar a questão desenvolvimento do projecto quase através da descrição da organização proposta para as relações dos componentes através do que normalmente se identifica como a arquitectura.

Não se julgue contudo, que o significado de arquitectura está consolidado e representa a mesma coisa para todos. Para uns a arquitectura, é a especificação, para outros é a consequência da actividade de projecto. No entanto, parece ser comum a todos que a arquitectura funciona como uma ferramenta que todos usam para desenvolver o projecto, sendo caracterizada como o quadro unifica os diferentes componentes do projecto e sustenta a evolução do mesmo.

Para quem na prática constrói sistemas autónomos a arquitectura resulta como uma tentativa de "standardizar" o processo de obter uma solução, isolando componentes e funções, classificando-os de várias maneiras com o objectivo de obter padrões que originem processos de rotina no projecto. Normalmente, a arquitectura é caracterizada em documentos com diagramas, caixas que identificam componentes (ou funções) e setas que definem as ligações dos componentes (ou funções) em diferentes aspectos tais como fluxo

de dados, sequenciamento de tarefas (melhor seria dizer composição de tarefas),, definição de mecanismos de sequenciamento de acontecimentos, relações hierárquicas de causalidade ou relações tipo cliente-servidor. Cada um destes "componentes" pode ser ainda objecto de uma análise mais específica surgindo naturalmente as sub-arquitecturas.

Ainda sem entrar numa definição precisa podemos dizer que: arquitectura é um conjunto de instâncias de modelos de componentes heterogéneos, de ligações entre esses componentes e de regras de composição. Quando a complexidade e a dimensão dos sistemas é elevada, tipicamente procede-se a decomposição da arquitectura em sub-arquitecturas.

Constata-se pois que os sistemas autónomos são projectados recorrendo a abordagens "ad hoc" baseadas fundamentalmente na análise da experiência da construção de sistemas similares. Já existem alguns esboços para o desenvolvimento de metodologias formais que orientam a possível aplicação da verificação formal e da síntese automática de controladores do comportamento de sistemas ao processo de projecto até de implementação. Contudo, estes esforços na verificação e na síntese automática têm-se limitado a desenvolvimentos em níveis de grande abstracção e geralmente aplica-se a abordagens conceptuais, não sendo ainda possível permitindo o uso generalizado destas técnicas e das ferramentas em sistemas de elevada complexidade e heterogeneidade.

Com o ritmo crescente de inovação tecnológica tem-se assistido à multiplicação de projectos de desenvolvimento de sistemas autónomos e, consequentemente, à tentativa de automatizar o projecto, ou numa perspectiva intermédia, definir ferramentas que suportam o projecto, e promovam a síntese automática de alguns componentes. Exemplo muito concreto é o esforço levado a cabo pelo projecto "Smart Vehicle" [Smart 02] que tem por objectivo o desenvolvimento de uma metodologia que integra ferramentas que auxiliam o projecto de sistemas ligados à indústria da automação. "Smart Vehicle" integra tecnologia desenvolvida no projecto MoBIES [Mobies 01]. O ambiente que o MoBIES pretende proporcionar é caracterizado por: modelização e simulação de sistemas híbridos; análise e síntese de sistemas de controlo; prototipagem de controladores; geração automática de código, verificação de código via simulação e verificação de componentes electrónicos via "Hardware-in-the-Loop". O grande desafio do projecto "Smart Vehicle" é o de proporcionar às equipas de projecto um conjunto de ferramentas que permita melhorar performances do projecto em parâmetros como a qualidade e o "time-to-

Notar que nem o "Smart Project", nem o MoBIES reclamam ter estabelecido um quadro formal.

market", bem como uma abordagem de "modelo único" nas múltiplas vistas. Deste modo, permite aos engenheiros das diferentes disciplinas trabalhar no seu domínio específico e, simultaneamente, manter a consistência e a eficiência de uma abordagem de modelo único. Outro aspecto fundamental deste projecto consiste na utilização de modelos dos componentes no desenvolvimento de modelos dos sistemas, facilitando desta forma a reutilização de código nas diversas fases de projecto. Se bem que os ambientes de projecto ainda não promovam a síntese na totalidade, obrigam ao desenvolvimento de boas práticas e evitam os erros.

6.3.2 Cenário para o desenvolvimento de uma metodologia

Por forma a estabelecermos ideias quanto ao enquadramento que é necessário efectuar para realmente obtermos uma metodologia formal, vamos tentar clarificar como se esquematizam as diferentes semânticas do projecto Os passos iniciais ocorrem a nível conceptual que, no caso dos sistemas robóticos, se inicia no estudo dos comportamentos admissíveis (para simplificar pensemos no movimento) e no projecto dos controladores.

As especificações de projecto são realizadas para uma situação de execução real e não simulada e a semântica desenvolvida pelo sistema na sua interacção com o ambiente tem uma expressão física. Na realidade, todas as trajectórias do estado do sistema pertencem ao conjunto de trajectórias que respeitam as leis da física que chamaremos conjunto W. A especificação do comportamento desejado resulta num subconjunto destas trajectórias S. As trajectórias dos comportamentos que resultam da implementação I são tais que $I \subseteq S \subset W$. O problema é, como garantir que $I \subseteq S$.

Normalmente (em laboratório) o que se passa é que os projectistas enumeram as situações que acham mais críticas, e após a construção o sistema ou de um subsistema, realizam alguns testes para verificar se o comportamento está dentro do esperado. Esta forma enumerativa de realizar testes de comportamento de forma a serem retiradas algumas conclusões não será muito útil se não existir outra forma de avaliar o projecto.

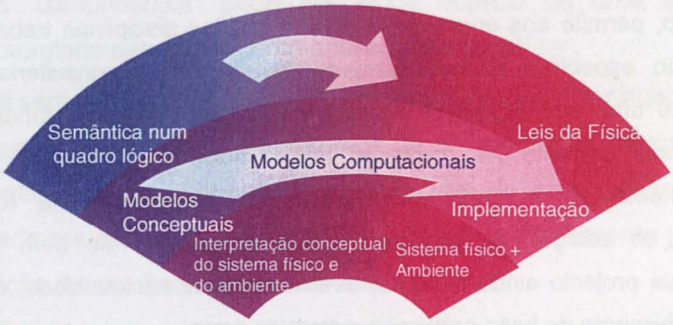


Figura 6.3-1

Mapa semântico da metodologia

Contudo as primeiras fases de projecto ocorrem a um nível mais abstracto, no qual as leis da física são interpretadas num quadro lógico, cuja a semântica deverá ser $W_c \subseteq W$. Consequentemente, a especificação no quadro conceptual será $S_c = S \cap W_c$, onde os comportamentos implementáveis (verificáveis) e sintetizáveis serão $I_c \subseteq S_c$. O processo de síntese tem de garantir que só $(I_c \cap W_c) \subseteq S_c$ são implementados. A qualidade do processo de síntese dependente pois da definição de W_c, da capacidade de garantir o quadro semântico da especificação implementada incluído no da especificação conceptual, por outro lado do processo maximizar I.

6.4 Formalização da metodologia

Se bem que ainda não exista a definição de um quadro formal e consequentemente o "algoritmo procedimental" para desenvolver o projecto de sistemas autónomos, têm sido desenvolvidos vários esforços no sentido de formalizar aspectos parcelares do projecto em áreas afins, como a teoria de controlo ou a teoria das linguagens.

[Edwards 99] propõe uma estrutura para a formalização do projecto de sistemas "embedded" que, em combinação com as abordagens formais nas áreas disciplinares que integram, tornam os sistemas autónomos um bom ponto de partida.

Notar que o quadro que [Edwards 99] reclama como sendo formal, ainda é muito genérico, faltando definir que modelos, ferramentas a utilizar na análise, verificação e síntese de comportamentos, como passar da

especificação conceptual para a implementável. O quadro formal proposto é composto por:

a) uma especificação funcional através da descrição explícita das relações entre as entradas e as saídas bem como informação de estado,

b) um conjunto de propriedades que o objecto do projecto deve satisfazer, em conjunção com a especificação funcional,

c) índices de desempenho que permitam avaliar a qualidade do projecto e o objecto em causa, e

d) restrições aos índices de avaliação (pode ser interessante limitar o custo).

Este quadro é fundamentado da seguinte forma: a especificação caracteriza a operação do sistema; as restrições ao índice desempenho limitam o "custo"; as propriedades não descrevem o comportamento do sistema mas afirmam características deste; as propriedades organizadas em dois tipos – as inerentes ao modelo utilizado na especificação (ou seja o modelo garante de forma construtiva) que podem ser provadas sintacticamente e as que podem ser provadas semanticamente (por simulação exaustiva, ou por transformação simbólica).

6.5 Quadro metodológico proposto

Podemos dizer que como resultado de estágios de projecto deve resultar uma arquitectura de software e uma hardware.

Para quem projecta, uma parte do problema consiste em decidir como organizar o software e hardware, decidir que partes devem ser implementadas em componentes programáveis, em hardware especializado ou implementado num " kernel" tempo-real. A outra parte do problema é encontrar as soluções para o controlo do sistema, ou seja desenhar a arquitectura de controlo e encontrar as melhores estratégias de controlo a implementar.

Notar que a metodologia que está a ser por nós proposta não possui uma estrutura formal, pois não é pelo facto utilizar modelos formais em aspectos parcelares do projecto que asseguramos que uma metodologia garante que a implementação é uma concretização da especificação, nem que as propriedades que eventualmente conseguimos provar em algum dos estágios da especificação, estão presentes na implementação.

As linhas metodológicas por nós propostas para o projecto de sistemas autónomos orientam-se numa primeira etapa para o desenvolvimento conceptual do projecto. Para esta fase do desenvolvimento propomos decompor o sistema segundo uma linha funcional, para o qual utilizamos um modelo da arquitectura conceptual proposto [Sousa 95] e [Pereira 01]. O desenvolvimento da arquitectura conceptual é o primeiro passo, onde são esboçados e identificados os manobras (tarefas e actividades) elementares e as leis de controlo contínuo. Nesta fase são implementados testados os primeiros esboços das leis de controlo.

Neste quadro de relações funcionais, utiliza-se um ou mais modelos formais para descrever o comportamento do sistema, acolhendo aí a síntese funcional dos controladores. Normalmente utilizam-se ferramentas de simulação ou verificação para ajudar no processo de síntese e modelização. Nos projectos que estamos envolvidos, a titulo exemplificativo podemos dizer que normalmente recorre-se à ferramenta Matlab para ajudar na síntese de estratégias de controlo [Almeida 97] ou a Shift [Fidalgo 99]. Na ajuda da síntese e modelização dos operadores de composição de tarefas utilizamos a ferramenta Kronos. É evidente que enquanto não existir de forma integrada uma toolbox que permita manter de forma consistente os diferentes componentes do projecto, vai ser necessário manter de forma heterogénea as diferentes vistas do projecto, para simular o comportamento do sistema e sintetizar os controladores, por forma a solucionar as diferentes componentes, quer do controlo quer da navegação.

Numa segunda fase decompor (especificando os componentes, ou seja sintetizar os componentes - aqui o uso de modelos formais pode ficar mais complexo-) em componentes de software e hardware as soluções para o implementação do sistemas e respectivos controladores, aqui utilizamos a arquitectura originalmente apresentada por [Silva 99] e que será aqui neste texto refinada. Este processo resulta no mapeamento da arquitectura conceptual numa arquitectura de software/hardware, e passa inicialmente pela selecção dos modelos computacionais que permitem implementar os desenvolvidos na arquitectura conceptual sem perda das respectivas propriedades). A introdução de modelos computacionais no projecto dos sistemas de controlo em combinação com a consideração do modelo físico da instalação a controlar vai permitir a verificação e a análise do sistema implementado. Este movimento resulta em somar aos modelos funcionais, os aspectos estruturais da implementação, aproximando o modelo conceptual da solução implementada.

Após os primeiros passos na identificação e decomposição das funcionalidades, especificam-se os componentes básicos e os mecanismos de comunicação, permitindo analisar o impacto destes no desempenho da arquitectura de controlo.

Como estão muitas variáveis em jogo, "à mão" não nos é possível realizar o projecto em dois passos, é pois necessário proceder de forma cíclica, de modo a reflectir nos modelos conceptuais os aspectos que afectam o controlo resultantes da implementação (os componentes irão ser baseados em

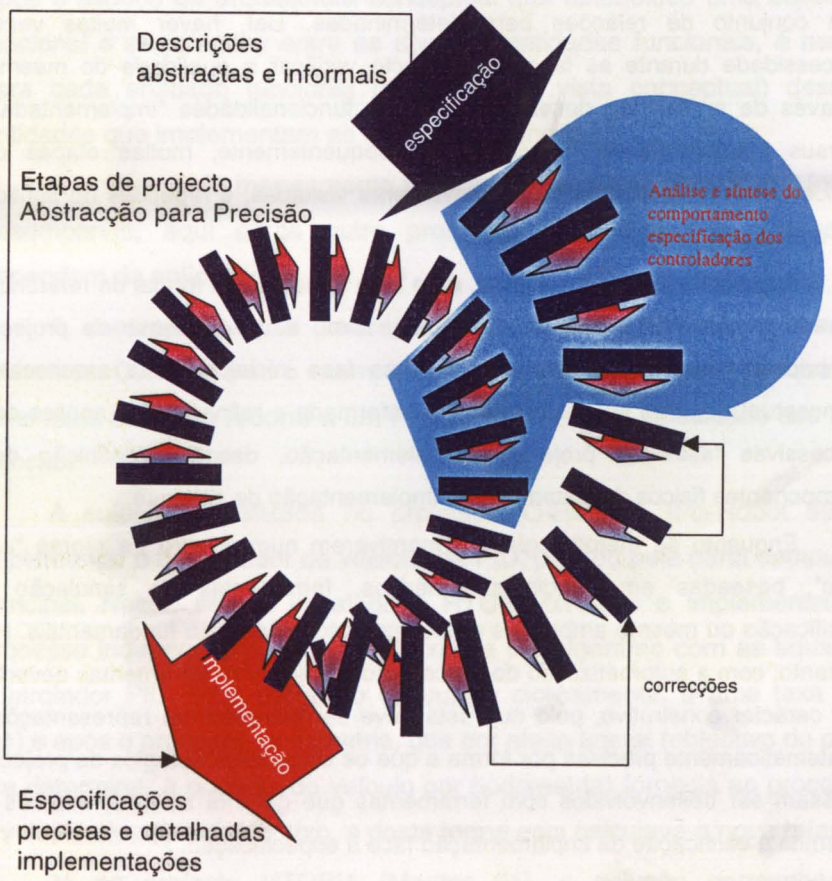
sistemas digitais, que partilham tarefas), e consequentemente reflectir na implementação (modelos que permitem a especificação dos componentes implementáveis) as alterações sugeridas nos controladores. Deverá ser objectivo da metodologia caminhar para um quadro em que as sucessivas etapas no processo de síntese (da abstracção para a implementação) sejam realizadas de uma forma automática (logo formal) garantindo que os componentes implementados resultam correctos.

Figura 6.5-1

processo metodológico



Desenvolvimento incremental e cíclico da arquitectura, partindo da estrutura inicial (que resulta do mapeamento) e através do sucessivo refinamento dos componentes da arquitectura, implementar soluções com desempenho crescente e verificação gradualmente mais exigente.



A metodologia por nós seguida é vista de uma forma genérica como uma sequência de etapas que transforma um especificação descrita de uma forma informal numa especificação precisa e detalhada que permita a implementação. Os estágios intermediários do processo são caracterizados por transformar descrições abstractas em descrições mais concretas. Para cada uma das fases do processo do projecto a descrição do "problema" é a especificação e a descrição do resultado do processo é a implementação. Um dos grandes problemas que se coloca é que a descrição dos problemas nas diferentes etapas do projecto é informal, e não é logicamente ligada através de um conjunto de relações bem determinadas. Daí, haver muitas vezes necessidade durante as fases de projecto verificar a qualidade do mesmo, através da análise do desempenho e das funcionalidades "implementadas" versus funcionalidades "descritas". Consequentemente, muitas etapas do processo de projecto têm de ser novamente visitadas, e repetidas no sentido de corrigir.

A validação do projecto, ou das sucessivas etapas de projecto podem ocorrer por:
Construção, em que as propriedades são consequentemente garantidas através da síntese formal.
Verificação, em que as propriedades são demonstradas sintáctica e semanticamente.
Simulação, em que, através de um modelo equivalente ao implementado, se verifica o comportamento do modelo em função das entradas.
Intuição, a propriedade é tautológica

A ausência de quadro formal, e de uma metodologia formal de referência para o projecto de sistemas autónomos, levou a um processo de projecto "imbricado" com o desenvolvimento numa fase inicial de uma arquitectura conceptual, que foi sucessivamente transformada e refinada pela análise das sucessivas fases do projecto e implementação, desde a definição dos componentes físicos do sistema até à implementação de software.

A semântica associada ao comportamento de um sistema autónomo desenvolve-se restringida pelas leis da física.

Enquanto as metodologias se mantiverem num quadro de regras "ad-hoc", baseadas em princípios genéricos, ferramentas de simulação e verificação ou mesmo ambientes de código automático, são fundamentais. No entanto, com a automatização do processo de projecto as ferramentas deverão ter carácter construtivo, pelo que este deve ser baseado em representações matematicamente precisas por forma a que os sucessivos estágios de projecto possam ser desenvolvidos com ferramentas que garanta as performances e permita a verificação da implementação face à especificação.

6.6 Princípios de Projecto de Sistemas

Este quadro metodológico cria a necessidade do desenvolvimento de duas vistas sobre o objecto de estudo. Uma que definimos como arquitectura conceptual, onde se processa a análise do sistema e se promove o estudo da síntese do comportamento. Outra que promove o desenvolvimento do projecto

até se obter uma especificação implementável. Desta forma, criamos dois quadros com vistas distintas que conformam o projecto "limitando" a acção do projectista.

Na primeira vista, os detalhes de implementação são transparentes, e o grande objectivo é definir as estratégias de controlo a utilizar, determinando as referências de trajectória para os estados dos subsistemas e definindo os requisitos para os anéis de realimentação nos vários níveis de abstracção.

Na segunda, os detalhes de implementação são importantes pois afectam directamente as estratégias de controlo desenvolvidas.

O percurso do desenvolvimento metodológico segue o seguinte rumo: Após o esboço da arquitectura conceptual que estabelece uma especificação funcional e as relações entre as diversas entidades funcionais, é necessário para cada entidade funcional (do ponto de vista conceptual) desenvolver entidades que implementam as respectivas funções.

Exemplo, o "custo" associado à dimensão física de um componente num veículo autónomo submarino ou aéreo é muito maior de que num terrestre...

O processo de mapeamento funcional devia ser orientado por índices de desempenho, aqui surge outro problema a escolha dos índices, estes dependem da aplicação.

A mesma solução conceptual pode ter soluções implementáveis diferentes. Considere-se o exemplo do controlo de movimento em que, no seu nível mais baixo, se recorre a um PID para controlar a velocidade dos eixos de tracção.

A solução sintetizada no projecto PO-ROBOT [Po-Robot 99b] para implementar o controlador de velocidade PID, passou pela carta computacional principal. Nesta, estava instalado o RTOS VxWorks e implementou-se um processo independente em que executava um algoritmo com as equações do controlador PID. Este processo executava ciclicamente, a uma taxa fixa (20 ms) e após o processo hodometria, que por efeito lateral (objectivo do processo era determinar a posição do veículo por hodometria) fornecia ao processo PID a velocidade corrente do eixo, e desta forma este calculava a nova referência.

A plataforma utopia é a base dos jogadores de futebol robótico da equipa IsePorto [Iseport 01].

Já no projecto UTOPIA [Martins 01], a solução correspondeu, ao desenvolvimento de uma carta dedicada ao controlo dos eixos do veículo, que executa de forma independente da carta principal e comunica com esta pelo barramento.

As duas soluções são eficazes, e corresponderam de forma apropriada aos objectivos dos projectos. No entanto as soluções de arquitectura

Hardware/Software são diferentes e transportam consigo também problemas diferentes.

Na primeira situação, a concorrência do processo PID é lógica, sendo a sequência de acontecimentos do sistema marcada pelo "time-slot" da carta do CPU e pela política de escalonamento associada aos processos da carta principal. Como estes dois processos são executados dentro do mesmo "time-slot" o máximo atraso que se introduz entre a leitura do sensor e actuação do controlador PID é de 20 ms. Contudo como os processos que implementam as estratégias de controlo de movimento executam com taxas inferiores ao PID quando existe nova referência para o processo (PID), o atraso entre o requisito da nova referência pelo processo cliente e a sua efectiva atribuição atinge valores bem maiores e que no caso do PO-ROBOT podia chegar aos 50 ms.

No segundo, existe paralelismo real dos processos, e executam a taxas de relógios independentes, o "time-slot" da carta dedicada é variável e sempre inferior a 10ms. Como a carta é dedicada ao controlo de vários eixos, podemos optar pela leitura de sensor no instante imediatamente anterior à actuação do PID, desta forma o atraso máximo introduzido é o tempo que é gasto na execução do código de leitura do sensor e na execução do código que implementa o controlador PID e que é inferior a 1 ms. Na aplicação em causa, este atraso é desprezável e o erro cometido está dentro dos erros introduzidos pela modelização sistémica.

Serviu o exemplo para mostrar que um componente que resulta de uma arquitectura conceptual, quando é mapeado numa solução implementável pode evoluir de diferentes maneiras. Caracterizemos:

> o facto de existir uma das soluções com peças de hardware com relógios independentes, leva a especificações (modelização) de soluções diferentes em termos de lógica de acontecimentos. O "interleaving" que existe no caso do "PO-ROBOT" afecta a ordem, pois o relógio entre os dois processos é lógico.
>
> como na primeira situação o atraso podia ser significativo, houve uma alteração funcional ao componente para ter em conta o atraso, que foi consequência da solução implementável.

Na área dos do desenvolvimento de arquitecturas de software já existe trabalho considerável no desenvolvimento de metodologias de síntese, só que neste caso o Hardware é uma restrição e não uma variável a ter em causa. O nosso problema é querermos sintetizar o algoritmo e o computador ao mesmo tempo [edwards].

Decompor é um dos processos mais complexos na síntese destes sistemas, mais ainda quando a tarefa de controlo é complexa e a heterogeneidade do hardware/software é considerável. O quadro metodológico por nós proposto ainda não dá resposta a este problema.

Uma instância da arquitectura conceptual é o resultado do primeiro passo do projecto de sistemas autónomos, como veremos esta deverá respeitar a granularidade associada às entidades do caso em estudo, pois só assim a semântica dos modelos é equivalente à semântica que resulta da execução, pelo que aqui surge a primeira decomposição.

Quando a síntese passa para a fase da implementação, neste domínio é mais difícil determinar as soluções de decomposição, mas estas vão depender de uma forma mais concreta da arquitectura. Mecanismos de memória, partilha de informação e barramentos.

6.7 Arquitectura conceptual de controlo

Como já referimos a arquitectura de controlo conceptual é o primeiro passo no desenvolvimento de um sistema autónomo. O seu papel metodológico é servir de base para a análise, servir de guia para a simulação, e permitir a síntese dos controladores quer esta ocorra em "off-line" ou em "on-line". Historicamente, as arquitecturas conceptuais têm tido um papel mais abrangente, ultrapassando a mera função conceptual, mas servindo também de estrutura que especifica a implementação. Os esforços em formalizar o processo de projecto e o processo de síntese e de verificação do comportamento dos controladores têm originado uma clarificação no papel das arquitecturas. No entanto, como a síntese (ainda não formal, e muito menos automática) do sistema não é um mero processo combinatório, estas ainda incorporam mecanismos que permitem a introdução de detalhes que correspondem a implementação. A arquitectura conceptual adoptada baseia-se na apresentada inicialmente por [Sousa 95], e que [Pereira 2001] apresenta de um forma mais refinada e já associada a um conjunto de conceitos e modelos com vista a formalização e síntese de estratégias de movimento.

Esta arquitectura está fundamentalmente vocacionada para o desenvolvimento de estratégias de movimento

A arquitectura conceptual têm um carácter predominantemente funcional e passa pela organização das funcionalidades de forma a ser possível a obter a síntese de comportamento. Esta vai evoluir ao longo de três sub-arquitecturas hierárquicas: Organização; Coordenação; e Regulação.

Notar que o estágio de síntese denominado Regulação, era inicialmente denominado por estágio Funcional, ver publicações [Sousa 95], [Silva 99] ,no entanto regulação é uma denominação mais apropriada quando pois funcional possui um carácter muito mais abrangente

Organização

O estágio da Organização é o nível conceptual onde se processa a transformação dos requisitos colocados pelo utilizador (pode ser um operador ou um "programa computacional") numa referência para o controlo.

Neste nível definimos as entidades (fisicamente ou logicamente distintas) e os conceitos que as relacionam que permitem ao utilizador especificar a missão. A função deste estágio é a definição das actividades que irão ser desenvolvidas nos recursos do sistema e o seu respectivo escalonamento. Esta referência é obtida através de funcionalidades de planeamento, e de transformação para referência do nível de coordenação através da síntese de referências.

O resultado deste nível tem em conta os subsistemas que podem prestar serviços, e que actividades podem ser executadas em cada um dos subsistemas, bem como a incompatibilidade temporal e espacial destas actividades.

A missão é baseada na composição de tarefas (chamadas de manobras no caso do movimento) cuja a estrutura já foi sintetizada em "off-line", e que os respectivos controladores parametrizam em "on–line".

Fundamentalmente na análise dos requisitos da aplicação já definimos o conceito informal de manobra, e essa é a entidade como qual o planeamento trabalha

Coordenação

O grande papel deste estágio é gerir o desenvolvimento de uma dada missão, que se expressa num conjunto de subsistemas.

É preciso ficar claro a diferença entre distribuído e descentralizado.

Do ponto de vista conceptual é fundamental gerir de uma forma unificada (não é o mesmo que centralizada, poderá ser descentralizada) as diferentes actividades que é necessário fazer evoluir, definir a sequência, o paralelismo e a concorrência, a sobreposição temporal.

É nossa concepção que se a granularidade da coordenação necessita de ser mais fina, então esta deve ocorrer do ponto de vista de uma manobra e definição de um novo recurso que englobe os subsistemas em causa.

Então o nosso conceito de coordenação assume duas vertentes:

Nesta função a coordenação assume um carácter distribuído

Uma estrutural, que implica a definição do que é um subsistema, que tarefas (manobras) que estão sintetizadas e que pode o subsistema executar. Ou seja, a definição de uma estrutura dinâmica sobre a qual podem ser projectadas tarefas. Tarefa assume como uma sequência de leis de realimentação contínua cuja a sequência emerge "on-line".

Nesta função a coordenação assume um carácter descentralizador

Outra de supervisor da execução das manobras, em que a coordenação acompanha execução das mesmas e sincroniza as actividades nos diferentes subsistemas, de acordo com a missão corrente.

Por exemplo o controlo de posição (pan-tilt) de uma câmara pode ser realizado de forma relativa à posição do veículo ou de forma absoluta relativamente ao mundo. Na primeira, o controlo da câmara é realizado de forma independente, pelo que a dinâmica do "pan-tilt" não depende endogenamente do veículo. Já no segundo caso, o controlo da câmara e do veículo têm de ocorrer de forma "unificada". Várias soluções podem resultar, e uma lei de controlo possível de implementar é aquela em que a coordenação dos dois sistemas ocorre continuamente, ou seja, a câmara e o veículo são trabalhados como se fossem um único sistema dinâmico.

Regulação

No nível de regulação são implementados os mecanismos de controlo que lidam com os sinais contínuos e que actuam directamente com os sistemas físicos. Esta relação e estes componentes são modelados recorrendo a equações diferenciais ordinárias ou a modelos não lineares. A leis de controlo implementadas por reguladores, seguimento de trajectória, estabilização.

Um sistema de equações neste nível define um subsistema, se por qualquer motivo as entidades físicas associadas a esses subsistemas interactuam ou têm objectivos de controlo que se interceptam então esta composição define um novo subsistema.

Esta reconfiguração sistémica a nível da regulação resulta das leis de controlo activas e das consequentes mapas de realimentação que se estabelecem.

6.7.1 Modelização

Face ao quadro dos domínios semânticos que definimos a utilização de modelos híbridos é fundamental.

Os técnicos que desde sempre estiveram associadas a projectar sistemas de controlo "reais" (com sentido físico), sempre lidaram com fenómenos discretos na dinâmica dos sistemas, e já desde essa altura o "switching mode control" foi abordado e implementado, mesmo num quadro sem formalização Matemática. É frequente em sistemas de controlo não lineares adoptar esta abordagem, através da aproximação do sistema por zonas de linearidade (em número finito) em que é considerado para cada uma das zonas um regulador linear. A selecção do regulador activo é feita por um controlador discreto que observa o estado continuamente.

A grande diferença entre esta abordagem e a abordagem sugerida pela teoria dos sistemas híbridos, é que nesta ultima a selecção do regulador não depende somente do estado contínuo mas também do estado do controlador discreto.

Contudo a nossa motivação para a introdução de modelos híbridos, não é consequência da introdução de computadores na malha de realimentação de um sistema, nem com o objectivo de encontrar a solução para o "switch mode control", mas resulta fundamentalmente da inadequação dos modelos matemáticos baseados em sinais contínuos, para a descrição da maioria dos sistemas autónomos (sistemas complexos). A teoria de controlo para os sinais contínuos apresenta soluções bem consolidadas para a síntese dos controladores.

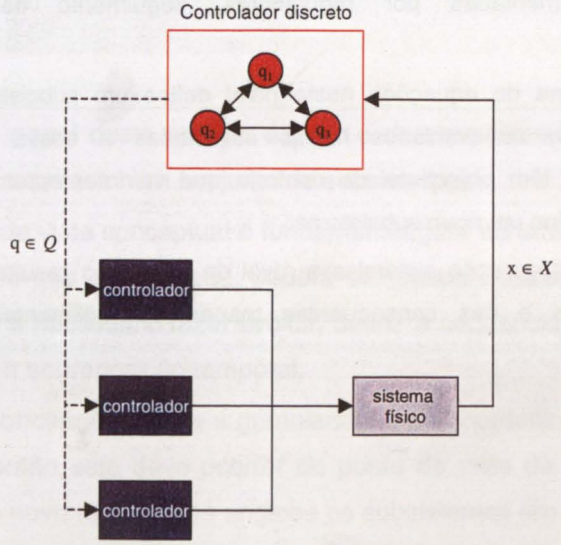


Figura 6.7-1

Esquema para o "switch mode control"

Os modelos de operação de múltiplos sistemas dinâmicos devem ter em conta os seguintes problemas:

a comutação entre os diversos modos de operação,

as interacções dinâmicas entre os diversos subsistemas.

Não existe (ainda) um modelo único para o desenvolvimento da arquitectura conceptual, de um sistema autónomo. Consequentemente é necessário definir modelos apropriados aos diferentes níveis da arquitectura.

No estágio de organização, o ambiente e a relação deste com o veículo é modelizada de uma forma topológica e operativa, através de um grafo cíclico.

Os nós do grafo representam configurações topológicas do domínio (domínio produto do estado do ambiente e do sistema autónomo) e os arcos representam, por um lado, as restrições topológicas associadas à transição entre as duas configurações, e por outro representam o "procedimento" que permite ao sistema transitar entre estas duas mesmas configurações.

Reparar que neste nível hierárquico o sistema é observado como uma unidade. Ainda não é aqui que aparecem as noções de paralelismo e concorrência. A este nível, é preciso clarificar o que é concorrência e paralelismo.

O significado do "procedimento", no nível da Organização assume um carácter bastante abstracto, e depende do nível de complexidade (talvez mais de abstracção) que as tarefas assumem no estágio de coordenação. Reparar que o resultado semântico a este nível se concretiza na concatenação de procedimentos, ou sequências de configurações topológicas que o sistema necessita de assumir, em que o estado evolui o estado restringido nessas transições. Ainda não existe a noção de subsistema, e problemas como paralelismo e concorrência ainda não apareceram.

As configurações topológicas que são associadas aos nós, permitem estabelecer o primeiro nível de fidelidade na granularidade, pois identifica os pontos do espaço de estado (qual estado? do topológico? claro) que são relevantes para o "utilizador". Por outro lado especifica claramente qual o comportamento esperado do sistema entre essas duas configurações quer através do "procedimento" que "une" as duas configurações, quer através da topologia que é vizinha dessas duas.

Para o nível de coordenação vamos adoptar dois modelos, um modelo de acontecimentos discretos, modular e hierarquicamente supervisionado para modelar a missão do sistema (com carácter hierárquico e descentraliza dor na relação com o sistema que é protagonista da missão, mas distribuído do ponto de vista da especificação do comportamento), e um modelo baseado em autómatos híbridos para modelar manobras e tarefas (neste nível a

coordenação assume um carácter claramente distribuído). O produto dos autómatos híbridos representa o sistema para o nível de coordenação.

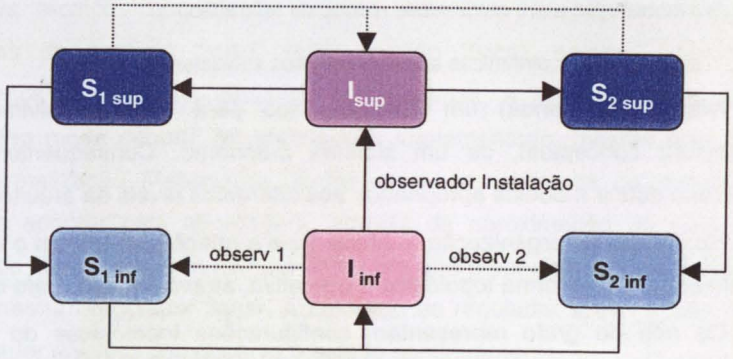


Figura 6.7-2

Modelo para a coordenação, Supervisão hierárquica e modular de modelos híbridos

Na figura anterior podemos observar os dois supervisores hierárquicos, que representam a especificação, a *Instalação superior* representa abstractamente o sistema, a *Instalação inferior* é o produto dos autómatos híbridos (parte discreta) que representam as diversas actividades desenvolvidas por subsistemas independentes. Os canais identificados como observadores podem não permitir observações totais mas sim parciais (este problema surge quando evoluímos no projecto para fases mais próximas da implementação). Deste facto resulta problemas na ordem da ocorrência dos acontecimentos.

Os modelos híbridos seleccionados partilham parte do modelo com o nível de regulação.

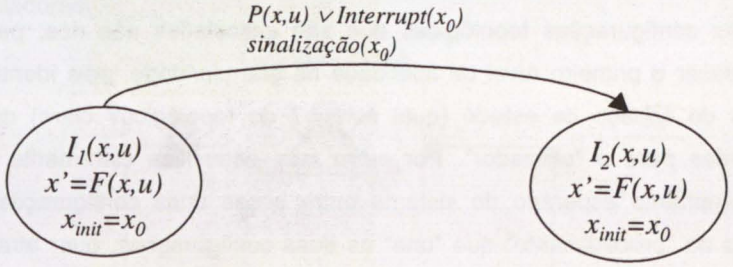


Figura 6.7.3

A modelização das interacções entre os diversos subsistemas pode assumir, bastantes particularidades e mecanismos, e que pode afectar o modelo conceptual usado, entre os diversos subsistemas

Um sistema híbrido é um sistema dinâmico de tempo contínuo, caracterizado por estados discretos e transições instantâneas. Os estados discretos são caracterizados por restrições nas trajectórias contínuas identificados na figura como *I (x,y)*, estas restrições designadas normalmente por invariantes, definem o espaço de estado válido para o estado corrente. do sistema. As transições podem ocorrer por dois mecanismos, um externo simbolizado pela etiqueta *Interrupt* que permite a sincronização deste autómato com outro causal da transição, outro mecanismo de carácter aleatório ocorre

quando o domínio da guarda e da invariante se interceptam. Quando o conjunto da intercepção é composto só por um ponto esta transição chama-se emergente. Quanto ocorre uma transição pode-se "emitir" um sinal por forma, a que outros autómatos possam sincronizar e efectuar o "reset" de uma ou mais variáveis. O mecanismo de sincronização que pode ocorrer é de ***1:N***.

O nível de regulação é modelado pelas equações dinâmicas nos estados dos autómatos híbridos.

Nesta fase do processo metodológico a arquitectura desenvolvida permite-nos, desenvolver as estratégias de controlo através da definição das funcionalidades que é necessário implementar e da sua síntese. Algumas da propriedades podem ser conseguidas construtivamente (a propriedade de "non-blocking" - normalmente associada a modelos computacionais - pode ser conseguida na modelização das manobras, bastando para isso que a reunião dos domínios da guardas das transições que ocorrem num estado seja igual ao domínio da invariante) sem ainda recorrer a processos de síntese "automática", outras num processo interactivo avaliadas por simulação ou recorrendo a ferramentas de software apropriadas para verificação formal. O ideal é que toda a síntese dos controladores fosse já nesta fase realizada de um forma automática.

6.8 Algumas questões sobre arquitecturas de controlo

Nesta secção iremos apresentar algumas reflexões e as consequentes interrogações apresentadas por [Varaiya 98] sobre sistemas hierárquicos, que sustentam um novo caminho para a formalização de sistemas complexos. Enquanto historicamente todos os desenvolvimentos que procuram estabelecer uma abordagem formal, fazem-no na busca de uma uniformização de estrutura semântica, no trabalho referido é sustentado que, quando implementamos sistemas complexos, o fazemos num quadro de múltiplas semânticas e que, consequentemente, será com esta multiplicidade que devemos procurar o estabelecimento do quadro semântico.

[Varaiya 98] sustenta, que não restam dúvidas quanto a característica fundamental de uma arquitectura de controlo, que esta deverá ser distribuída hierarquicamente, pois desse facto resulta (ou é consequência de - notar a bidireccionalidade da causa/consequência):

- redução da complexidade dos requisitos de comunicação e computação;
- um processo mais fácil de obter um conhecimento profundo induzido pela estrutura hierárquica;
- a modularidade que permite uma adaptação facilitada à mudança;
- maior robustez;
- permite lidar mais facilmente com a escalabilidade;

esta questão só se coloca quando se passa ao processo de mapeamento num arquitectura implementável

e sugere que devemos centrar a discussão em torno de questões tais como, "Que arquitecturas permitem avaliar a performance do sistema?", "Que ferramentas ajudam no desenvolvimento e no projecto de boas arquitecturas de controlo?".

Os métodos formais são uma forma de definir um sistema matematicamente correcto, permitindo provar que um dado sistema satisfaz uma dada propriedade.

Pnueli [Pnueli 95] identificou os principais elementos que devem constar de um quadro formal de análise, e que são:

- Um modelo semântico para capturar o comportamento do sistema.
- Uma linguagem de especificação para descrever o sistema.
- Uma linguagem de especificação de propriedades, para descrever as propriedades que o sistema deve satisfazer.
- Técnicas de análise das propriedades dos sistemas.

Notar que, o quadro formal proposto por [Edwards] possuía um carácter metodológico. O objectivo deste quadro é garantir que as propriedades se mantêm da especificação até à implementação. Ou seja como passar estágios de projecto da abstracção até à concretização. Já Pnueli apresenta um quadro de análise do sistema sem a preocupação de extrair formalmente a solução de projecto.

A formalização de uma arquitectura conceptual (organização do sistema de controlo), e o estabelecimento de ferramentas matemáticas que permitam a síntese e análise do controlo de sistemas complexos numa perspectiva formal, ainda esta longe de atingir um patamar, onde existam um conjunto de soluções que permitam dar respostas à questões colocadas em [Varaya 99].

os recentes desenvolvimentos em sistemas híbridos e discretos, são bons pontos de referência para o estabelecimento de uma arquitectura, mas ainda não permitem o desenvolvimento de uma metodologia de projecto completa

Esta discussão sobre a formalização de sistemas complexos, explicitando as relações entre os elementos caracterizadores do quadro formal, criando uma maior aderência das definições formais aos objectos reais e sugerindo um nova perspectiva na definição do quadro formal, no que concerne especialmente à semântica.

Em [Varaya 99] é apresentado três estruturas para a definição de um quadro teórico para o estudo de sistemas complexos,:

Sintaxe – regras para descrever e especificar o sistema

Semântica – modelo matemático para interpretar as expressões sintácticas.

Ambiente – real ou simulado, ou imaginado para implementação das descrições.

e estabelece as seguintes relações entre os elementos caracterizadores:

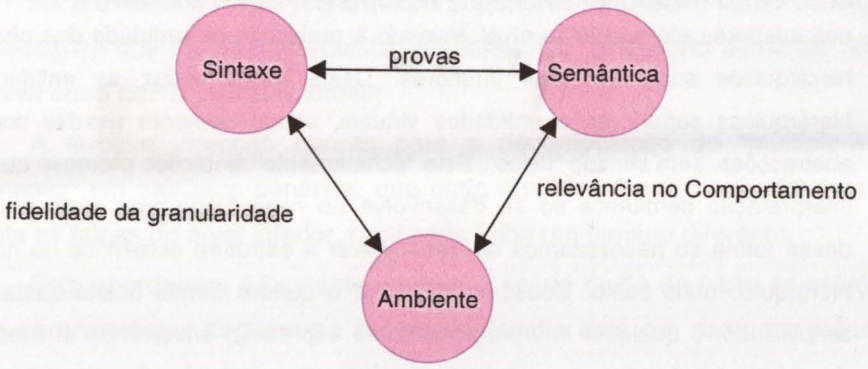
relações de prova – factos relatados por expressões sintácticas são demonstradas no domínio semântico

Comportamentos relevantes – a relevância do comportamento expressável pela semântica é verificado no ambiente.

Fidelidade da granularidade – elementos relevantes do ambiente correspondem a entidades na sintaxe.

Figura 6.8-1

Estrutura do quadro formal proposto em [Varaya99]



Notar que nos primeiros três capítulos da dissertação realizamos uma exaustiva análise do problema, permitindo por um lado avaliar os domínios semânticos usados no projecto dos controladores e por outro definir claramente a decomposição e a identificação das entidades em jogo.

O desenvolvimento de uma arquitectura de controlo deve ser desenvolvida ao longo destas três estruturas, mantendo a harmonia (coerência) dos desenvolvimentos através das relações que entre elas se estabelece. No entanto alerta para as dificuldades que se encontram no desenvolvimento do modelo semântico.

A característica fundamental para a estrutura sintáctica, é que esta tem de permitir captar os elementos relevantes da hierarquia distribuída. Por um lado têm de permitir identificar as diversas entidades quer pela sua identidade quer pela sua funcionalidade, por outro tem de permitir estabelecer as relações entre as diferentes entidades.

A definição das relações de hierarquia que se pretende estabelecer, assume um carácter mais complexo, que vai desde a necessidade da definição de hierarquia de objectivos, tarefas, conceitos, classificação, composição de sistemas, etc., assumindo um entendimento que vai desde da noção "ranking" até à noção de abstracção. O núcleo central da estrutura sintáctica deve permitir o estabelecimento de relações de ordem parciais entre as diversas entidades de forma a estabelecer as diversas e diferentes relações de hierarquia.

No que concerne à estrutura semântica, esta é normalmente caracterizada pelo o que é referido como, "one-world semântica". Esta abordagem permite uma única interpretação para a descrição e avaliação do sistema, e permite um bom desenvolvimento do projecto, garantindo uma clara transição entre os requisitos do sistema, a especificação, o projecto e a implementação. Esta vantagem contrapõe-se à rigidez da estrutura que inviabiliza a sua utilização para a maioria dos sistemas complexos e heterogéneos.

Uma estrutura semântica caracterizada pelo o "one-world-semantic" promove a tradução das entidades das hierarquias superiores sucessivamente nas inferiores até ao último nível, levando à projecção da entidade dos níveis hierárquicos superiores nas inferiores. Desta forma reduz as entidades hierárquicas superiores a entidades virtuais, sendo somente usadas como abstracções, sem sentido "físico". Este "achatamento" sintáctico promove que a interpretação semântica só se desenvolva no nível hierárquico mais baixo, dessa forma só necessitamos de desenvolver a estrutura semântica no nível hierárquico mais baixo. Consequentemente o quadro formal ficaria bastante simplificado, e qualquer interpretação duma expressão envolvendo entidades dos níveis hierárquicos superiores, era realizado em dois passos, primeiro a tradução e depois a interpretação.

Esta opção metodológica parece ser vantajosa, mas vejamos a implicação desta opção nas relações entre as três estruturas propostas. Como a metodologia se baseia no princípio da tradução, e do respectivo achatamento semântico, qualquer prova de uma propriedade do nível hierárquico superior (relação de prova sintaxe-semântica) realiza-se no nível inferior da hierarquia. Ora, desta forma o resultado desta relação de prova está fortemente dependente dos pormenores dos níveis hierárquicos inferiores, e que caso o nível superior não tenha em conta corre-se o risco de se obter as respostas erradas. Este problema no desenvolvimento dos projectos de sistemas complexos não existe porque nós na prática não o fazemos assim. Na prática nós usamos uma estrutura semântica caracterizada por "multi-world semantics".

A proposta para a estrutura semântica do quadro formal, vai no sentido de que as provas se realizam em cada nível hierárquico associado à sua semântica. Esta organização corresponde às práticas de projecto comummente utilizadas. No entanto esta "separação" semântica sacrifica alguma capacidade de decisão. A título exemplificativo, falhas no nível hierárquico inferior não são

tido em conta nos níveis superiores. Todas as provas em cada nível supõem que os níveis inferiores têm um comportamento ideal. Ora há decisões ou "planeamentos" que necessitam ter em conta cenários que não os ideais (o "one-world" tinha em parte esta vantagem). A nossa intuição leva a optar por múltiplas semânticas desconexas, mas que ainda não sabemos como as relacionar.

Para solucionar esta questão em [Varaiya 98] é apontado três direcções de análise que poderão conduzir a uma resposta.

Alternativamente a tradução sintáctica usada no "one-world semantic" seguida de interpretação no nível hierárquico inferior, compilar para um situação idealizada no nível hierárquico inferior e então interpretar;

As provas dos níveis hierárquicos superiores funcionam como condições necessárias dos níveis hierárquicos inferiores. No "one-world semantic" estas provas eram condições suficientes;

A terceira direcção aponta para a decomposição do "multiple-world semantic" em múltiplos cenários, que cada nível hierárquico superior tem em conta as falhas do nível inferior, para cada falha um cenário diferente.

Esta abordagem é bastante intuitiva e segue muito de perto as soluções que os projectistas seguem na prática, o desafio está em dar uma estrutura formal a esta abordagem.

6.9 Contribuição para estabelecimento de um quadro semântico

6.9.1 Problema

Varaiya aponta como um dos problemas, para a formalização de um quadro de "multi-world-semantic", o facto de não sabermos como relacionar as "desconexas" (aparentes) semânticas dos diferentes níveis hierárquicos, e aponta como vimos anteriormente três direcções para possíveis desenvolvimentos e que poderão resultar na resolução deste problema. As questões que Varaiya coloca apontam quase sempre para a inconsistência semântica, ilustrando com um exemplo paradigmático, que falhas do nível

hierárquico inferior não são tido em conta nos níveis hierárquicos superiores. Quem implementa sistemas autónomos é muito sensível a esta questão, pois a síntese de uma estratégia de controlo pode ter sido muito bem conseguida, mas quando implementada num ambiente real e não ideal pode não corresponder as expectativas e falha. No mosaico da hierarquia nem sempre um objecto (e aqui objecto deve ser entendido de forma muito liberal) possui representação nos diferentes níveis hierárquicos. A título de exemplo podemos ver que a posição do veículo no espaço possui representação nos diferentes níveis hierárquicos, no entanto os componentes que implementam os diferentes processamentos, só tem representação a partir do nível hierárquico que implementam.

6.9.2 Como compatibilizar estruturas infinitas com estruturas finitas?

Como é que utilizamos as ferramentas (por exemplo teoria dos autómatos) que dispomos para lidar com processos computacionais para lidarem com a continuidade. Arbib indica uma direcção para a resolução deste problema, introduzindo o conceito da relação de tolerância e a definição de espaços tolerantes [Kalman 69], originalmente referidos por [Zeeman 62]. A definição de tolerância é muitas vezes ilustrada com a noção da indistinguibilidade, ou utilizando um exemplo muito concreto com a noção da acuidade visual, em que dois pontos contíguos não são distinguíveis (por exemplo $a=b, b=c, a \leq c$ caracteriza os espaços tolerantes). Claramente se observa que a definição da relação de tolerância desenvolvida por [Zeeman 62] (ver definição de Arbib em [Kalman 69), não possui a propriedade da transitoriedade, pelo que não podemos trata-la como uma relação de equivalência (ver anexo A.4). No entanto analisemos a seguinte questão: Suponhamos que para controlar uma dada variável contínua, utilizamos um multímetro analógico que faz deslocar um ponteiro sobre uma escala. O deslocamento do ponteiro é um mapa de um para um, relativamente à grandeza medida. O domínio semântico da grandeza medida é contínuo, ou seja precisão infinita, o domínio semântico do deslocamento do ponteiro é contínuo, precisão infinita (estamos a admitir que o mecanismo físico é infinitamente preciso). Se duas medidas consecutivas forem diferentes conseguimos estabelecer uma relação de ordem entre elas, no entanto se duas medidas consecutivas não alterarem "significativamente" a posição do ponteiro,

Arbib coloca a questão "How can we put a relevant topology on a discrete set wich is not a discrete topology?" [ref Arbib].

somo levados a dizer que são iguais. Desta forma o domínio semântico dos valores efectivamente medidos é um espaço tolerante. No entanto, se por traz do ponteiro colocarmos uma escala podemos realizar medidas num domínio finito através da discretização da escala. Ou seja, há medidas com valores diferentes no domínio da grandeza, que não vão ser distinguíveis (reparar que é diferente da situação sem escala). No entanto a caracterização dos valores por classes, pode permitir definir uma relação de equivalência.

Notar que o problema da discretização do domínio da variável de controlo, não está na mesma dimensão do problema da utilização de modelos de tempo contínuo ou discreto.

Levantamos esta questão não só pela tentativa de lidar de forma coerente com as diferentes topologias (contínuas, discreta), mas também com a questão que se coloca no problema da "medida". A interacção de um sistema autónomo com a realidade física estabelece-se através dos processos sensoriais e de actuação. A medida da grandeza quer "sentida" quer "actuada", possui um limite de precisão. Desta forma o domínio semântico de uma variável do ponto de vista conceptual, pode não corresponder ao mesmo domínio quando esta variável é mapeada noutra que recebe a medida de uma dada realidade física. Este facto alerta-nos para a necessidade de analisar que implicações esta "perturbação" tem na demonstração de propriedades dos sistemas no plano conceptual e na sua aderência à realidade. Outra questão que se levanta é se os modelos utilizados nos níveis conceptuais devem capturar esta "realidade".

Outro factor que afecta o mapeamento das sucessivas arquitecturas dos componentes conceptuais em componentes implementáveis (aqueles que tem representação implementável, poderá haver componentes só de natureza conceptual) é o facto de muitas vezes usarmos um único processador para implementar vários módulos da arquitectura. Aceitamos que existe implementada uma certa política de sequenciamento, em que a única regra para a partilha do CPU pelos diferentes componentes foi uma "justa" distribuição de tempo face as necessidades de cada componente. Todos sabemos que a precisão do relógio do CPU é finita, e a resolução temporal que se consegue é utilizada para a implementação da politica de sequenciamento dos processos. Por este motivo, os acontecimentos que ocorrem (na realidade) em instantes temporalmente distintos e ordenados num intervalo de dois "ticks" consecutivos do relógio, não se vão distinguir. Resulta pois, que independente da utilização nos níveis conceptuais de modelos de tempo denso ou discreto, a ordenação temporal dos acontecimentos no mínimo deixa de ser estritamente crescente. Se entre os instantes t e $t+1$ ocorrerem os acontecimentos

$\vartheta_1(t+\tau_1)$ e $\vartheta_2(t+\tau_2)$ com $t<t+\tau_1<t+\tau_2<t+1$, e como para os componentes do sistema não há capacidade de distinguir $t+\tau_1$ e $t+\tau_2$ consequentemente a ordem desaparece. Os modelos normalmente utilizados para modelar sistemas dinâmicos consideram o conjunto T como um subconjunto ordenado de R (ou de N) mas, como vimos, este mapeamento da arquitectura conceptual numa implementada estabelece artificialmente sobre o domínio da variável tempo um espaço tolerante. Este facto perturba todas a as noções de causalidade (ou seja, nos modelos utilizados, o futuro não afecta o passado, tal como nos componentes que estes modelam).

A situação ilustrada anteriormente não assume um problema importante quando estamos a analisar a implementação de um dado sistema, pois esta incorrecta atribuição e implementação dos componentes são facilmente verificadas através de testes ou simulações. Geralmente, são introduzidos no sistema mecanismos que impõe que o conjunto T "apareça outra vez" ordenado. No entanto, num quadro de síntese automática, temos que introduzir na metodologia mecanismos que não permitam esta "alteração" nas propriedades do domínio semântico do T. Caso não o façamos corremos o risco das propriedades demonstradas num dado perfil de arquitectura (por exemplo conceptual) não terem correspondência noutro nível.

6.9.3 Sistemas dinâmicos congruentes

Vamos apresentar alguns conceitos que apontam um caminho para o desenvolvimento de uma base metodológica no projecto de sistemas autónomos, e que sugere o estabelecimento de propriedades entre as funções dos diversos níveis hierárquicos, e os respectivos domínios semânticos. O objectivo é desenvolver um quadro que promova a "invariância" semântica (relativos a uma propriedade) entre os diversos níveis hierárquicos e a congruência entre os diferentes representações dinâmicas (nos diferentes níveis hierárquicos). Os desenvolvimentos apresentados neste secção são baseados em desenvolvimentos bastante standards, e que podem ser encontrados em [Kalman 69] [Wonham 01], no qual em anexo apresentamos um pequeno resumo para referência rápida e suporte ao que seguidamente se desenvolve.

Vamos considerar o conjunto de estados sempre como um semigrupo, em que esta definida uma operação binária, e o elemento identidade. Na literatura muitas vezes referenciado com um monoide. A operação binária definida é da concatenação de trajectórias justapostas.

Consideremos a representação do sistema (ou antes de uma parte) num nível hierárquico inferior como o sistema dinâmico num conjunto X (domínio semântico) em que $\alpha: X \to X$ representa a função de transição de estado (uma qualquer) com seguinte interpretação. Os elementos $x \in X$ são os "estados" do sistema. Escolhido um estado inicial $x_0 \in X$, deixando o sistema evoluir sucessivamente pelos estados $x_1 = \alpha(x_0)$, $x_2 = \alpha(x_1)$, ... A sequência $\{\alpha^k(x_0) \mid k \in N\} \in X^N$ é o caminho de (X, α) com estado inicial x_0.

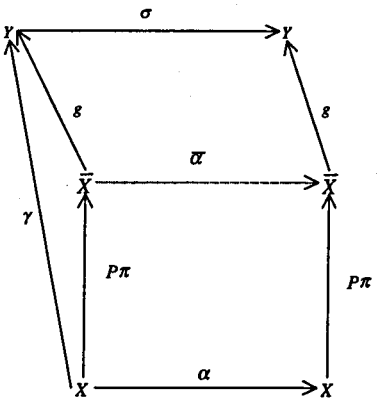
Consideremos Y o domínio semântico de um nível hierárquico superior. Seja $\gamma: X \to Y$, (definida tal como no anexo C.5) a função que define a hierarquia, e seja a $\sigma: Y \to Y$ a função transição de estado do nível hierárquico superior.

Consideremos agora $\Omega(X)$ como um lattice de relações de equivalência em X. Seja $\pi \in \Omega(X)$ uma relação de equivalência e denotaremos por $\overline{X}$ (alternativamente por X/π) o conjuntos das células distintas de π (ou partição de $\overline{X}$). Complementarmente denotaremos por P_π a função sobrejectiva que mapeia qualquer x no respectivo ***coset***:

$$P_\pi : X \to X/\pi : \; x \to [x].$$

Figura 6.9-1

Diagrama de interacções entre dois níveis hierárquicos.



Se a partição induzida pelo núcleo de equivalência de $\gamma: X \to Y$ (ver no anexo a definição de núcleo de equivalência) for superior à partição π, ou seja $\ker \gamma \geq \pi$, então existe um mapa único $g: X/\pi \to Y$ tal que $\gamma = g o P\pi$.

Seja π uma partição congruente (ver definição de congruente no anexo) para $\alpha: X \to X$, pelo que existe um mapa $\overline{\alpha}: \overline{X} \to \overline{X}$ tal que $\overline{\alpha} \circ P_\pi = P_\pi \circ \alpha$, consequentemente o sistema dinâmico $(\overline{X}, \overline{\alpha})$ pode ser interpretado como um agregação consistente do modelo de (X, α).

Se da mesma forma o $\ker\gamma \in \Omega(X)$ for uma partição congruente então existe um mapa $\sigma: Y \to Y$ tal que $\sigma \circ P_{\ker\gamma} = P_{\ker\gamma} \circ \alpha$. Pela relação que estabelecemos entre π e o $\ker\gamma$, intuitivamente podemos dizer como π preserva informação suficiente para calcular γ (via g), então $\sigma \circ g \circ P_{\pi} = g \circ P\pi \circ \alpha$.

Consideremos agora A o conjunto das funções de transição $\alpha: X \to X$. Seja $C(X) \subseteq \Omega(X)$ o conjunto de todas a relações de equivalência congruentes com todo elemento $\alpha \in A$, então as relações de equivalência $C(X)$ também são congruentes para toda a composição $\alpha_k \circ \alpha_{k-1} \circ \alpha_{k-2} \ldots \circ \alpha_1$.

Ver demonstração [referência]. Facilmente se compreende que assim se estabelece uma boa base semântica para a definição de autómatos híbridos

A observação do desenvolvimento dos projectos constata-se que, na prática, a definição de Y e de X e das respectivas funções de transição dos sistemas dinâmicos associados ocorre de forma "independente", e é sucessivamente refinada em função da análise dos resultados do desempenho desses sistemas dinâmicos.

Definamos a seguinte relação de equivalência:

$$\omega_0 := \sup\{\omega \in \Omega(X) \mid \omega \leq (\ker\gamma) \wedge (\omega\alpha_k) \wedge (\omega\alpha_{k-1}) \wedge \ldots \wedge (\omega\alpha_1)\},$$

como a partição mais fina do que o $\ker\gamma$ (mas a mais "grosseira" das mais "finas") congruente com todas as funções de transição. Esta partição deverá ser entendida como o limite de "precisão" com que o nível hierárquico superior poderá "observar" o nível hierárquico inferior.

Definir a função e o seu ponto fixo, implica solução algorítmica.

Consequentemente seria do interesse do projectista definir a função γ tal que $\ker\gamma = \omega_0$.

Não perdíamos generalidade se definíssemos a função γ como um *spray* que relaciona X para Y, e "construir" X como a reunião das células do $\mathrm{cov}(\gamma)$, ou seja:

Ver definições de *spray* e *cover* no anexo C.

$$X = \bigcup\{\gamma_y \mid y \in Y\}, \text{ tal que } \gamma_y := \{x \in X \mid (x, y) \in \gamma\} \text{ e}$$

Da mesma forma se poderia definir composições de relações e definir sucessivas factorizações das funções hierárquicas, desde que os "covers" induzidos em X formassem um lattice.

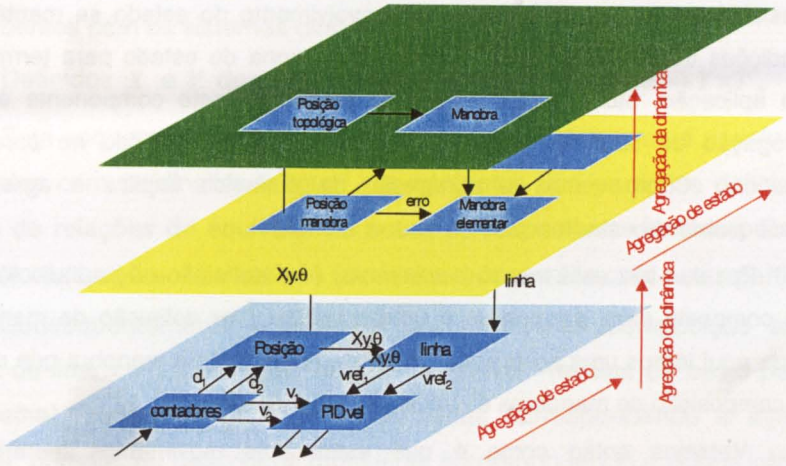
6.9.4 Agregação Hierárquica

Os desenvolvimentos descritos anteriormente, garantem a existência de funções e quais as características das propriedades a preservar nas relações hierárquicas de sistemas complexos para garantirmos a coerência dinâmica entre os diversos subsistemas.

Uma reflexão mais pormenorizada sobre a análise que descrevemos nos capítulos iniciais sobre a implementação de sistemas robóticos, ou sobre as implementações propostas por [Silva99], verificamos que a agregação hierárquica ocorre em duas direcções. Foquemos a nossa atenção no seguinte exemplo.

Figura 6.9-2

A organização hierárquica de um sistema autónomo faz-se em duas dimensões. Uma em torno da agregação das dinâmicas, outra em torno da agregação do estado.



Para não perdermos o contexto do que pretendemos ilustrar, não nos preocupemos com as acções de controlo que são exercidas, principalmente as acções de controlo hierárquico. Seja o caso de um sistema robótico que possui solidário com sistema mecânico de propulsão um sistema de contagem do deslocamento angular percorrido por cada roda do sistema de tracção. Associado a esses contadores existe um processo computacional que utiliza directamente esses resultados no controlo de velocidade dos actuadores (assim, há uma malha de controlo que se fecha). Essas medições são utilizadas por um outro processo que faz o cálculo da atitude do veículo, estes valores são utilizados no controlo do movimento elementar do mesmo, que por sua vez exerce uma acção de controlo sobre o processo que calcula as velocidades a desenvolver por cada roda propulsora (outro anel que de controlo que se fecha). Na primeira situação o domínio semântico a considerar era o das velocidades das rodas propulsoras. No segundo caso, o domínio semântico para a acção de controlo podia ser considerado o erro entre a

posição do veículo relativamente a uma recta fornecida por referência. Facilmente se compreende que o segundo anel de realimentação possui a mesma dinâmica do primeiro, contudo o mesmo não se passa para o estado em que existe uma agregação do estado. Basta pensar que se o veículo inicia de uma posição em que o vector de estado de erro é zero, todas os estados das trajectórias das velocidades das rodas que mantêm o vector de erro nulo são mapeados neste ponto do estado de erro igual a zero. Resumindo as semânticas são diferentes, mas a dinâmica é a mesma.

Continuemos a analisar a situação. A lei de controlo "linha" é implementada através do "encapsulamento" desta lei de controlo através de um componente que para além de implementar esta estratégia, avalia da possibilidade de a aplicar, se no desenvolvimento do estado se mantêm as condições da sua aplicação, se estamos na zona do estado para terminar a sua aplicação, etc.... Como vemos, a dinâmica deste componente é uma agregação da dinâmica "imposta" sobre o domínio referido no paragrafo anterior. Como vemos a agregação da dinâmica implica a agregação consequente do estado.

Por sua vez esta manobra elementar (ver definição no capítulo 4) pode ser composta para dar origem a uma manobra (ver definição de manobra), ainda aqui temos uma agregação dinâmica. No entanto a manobra que resulta da composição de manobras é uma agregação de estado.

Vejamos então como é que estes dois movimentos hierárquicos mapeiam e estabelecem as relações entre os domínios semânticos dos diferentes níveis de arquitectura. Este quadro de relações foi originalmente definido por [Wonham 01].

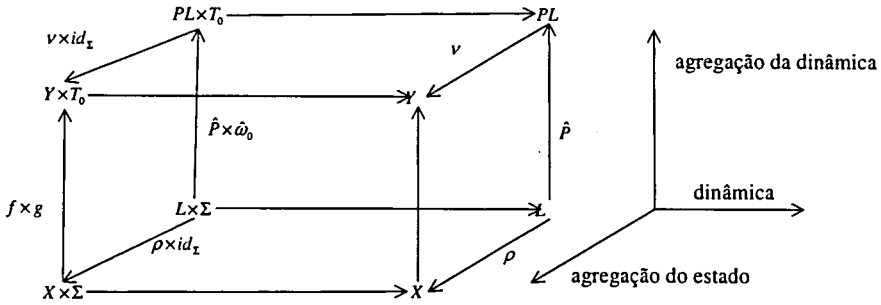


Figura 6.9-3

A linhas ortogonais de hierarquia.

Sem perda de generalidade da aplicação do quadro de relações, não vamos atender às topologias (continua ou discreta) dos domínios associados na estrutura, consideremos então $L \in Pwr(\Sigma^*)$ as trajectórias admissíveis do estado representativos da dinâmica do sistema (nível hierárquico mais baixo),

e Σ o domínio dos acontecimentos do sistema. Seja L um comportamento fechado.

Defina-se um segundo sistema com domínio de acontecimentos T, e seja $\hat{P}$ a função que realiza a tradução de trajectórias no domínio do primeiro sistema (hierarquia mais baixa) nas trajectórias do domínio do segundo, de forma muito própria W chama repórter a P o autómato que permite a implementação de $\hat{P}$.

A dinâmica do segundo sistema (hierarquia mais alta) pode ser definida $PL \in Pwr(T^*)$, ou seja depende da dinâmica definida por L, e pela função de tradução por $\hat{P}$.

W define $\rho : L \rightarrow X$, e $\nu : PL \rightarrow Y$ como a projecção natural dos sistemas dinâmicos, ou seja os estados de X e Y, são núcleos de equivalência congruentes com os sistemas dinâmicos.

Definidos X e Y desta forma então existe $f \times g : X \times \Sigma \rightarrow Y \times T$.

Este esquema de relações de domínios estabelece que é possível definir conjuntos com cardinalidade igual aos conjuntos das partições definidos pelo lattice de relações de equivalência sobre os respectivos domínios. Existe um conjunto de funções de transição (dinâmicas) coerentes com estas partições.

Consequentemente é possível encontrar uma composição sistémica capaz de lidar com dinâmicas "rápidas" e "lentas" (evoluir de baixo para cima na grelha), com descrições pormenorizadas correspondendo a informação "quase" perfeita, ou descrições grosseiras correspondendo a pouca informação (evoluir de traz para a frente na grelha).

6.9.5 Como utilizar este quadro semântico

Uma das dificuldades inerentes ao projecto de sistemas que lidam com "realidades físicas" resulta na leitura que os próprios sistemas fazem da sua interacção com o ambiente. Conceptualmente esta interacção é conformada pelas leis da física em uso, resultando na maioria das situações na caracterização contínua da topologia dos domínios das variáveis alvo da nossa atenção. Infelizmente, as trajectórias do estado destas variáveis que representam a dinâmica da interacção, não são facilmente observadas (como parece nos modelos conceptuais). Normalmente, esta observação é realizada noutros domínios parciais com os quais existe uma relação bem conhecida, e

seguidamente inferimos a dinâmica das variáveis que pretendemos "controlar", Este bloco da solução designa-se de fusão sensorial. Estabeleçamos então o quadro das relações das semânticas dos diferentes componentes em jogo. Consideremos X o espaço de estados das variáveis que pretendemos "controlar", e α a dinâmica (uma função de transição de estado que nós não conhecemos com precisão) da trajectória do estado.

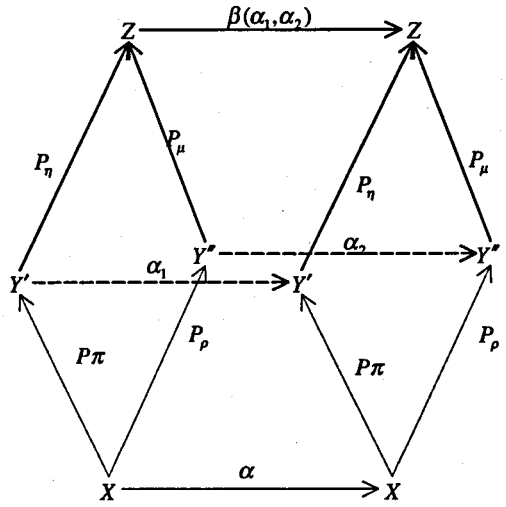


Figura 6.9-4

Quadro de relações semânticas.

Consideremos Y e Y' os estados e α_1 e α_2 as dinâmicas observadas. Admitamos que existem P_π e P_ν projecções canónicas de X para Y e Y' respectivamente, cujas partições induzidas sejam congruentes com a dinâmica α do estado X. Então devemos encontrar o domínio Z que é simultaneamente contra domínio das projecções canónicas P_η e P_ρ. Esta por sua vez devem induzir partições congruentes para α_1 e α_2 respectivamente. Como α_1 e α_2 também não são totalmente conhecidas, consideremos A_1 e A_2 os conjuntos das dinâmicas nas "vizinhanças" de α_1 e α_2. Pelo que P_η e P_ρ deverão ser congruentes com todos os elementos dos conjuntos A_1 e A_2, e consequentemente com qualquer composição das funções de transição pertencentes a estes conjuntos. Construtivamente $\beta(\alpha_1,\alpha_2)$ depende α_1 e α_2 (motor da causalidade) e representa a dinâmica observada. Claro está que Z é uma partição de X congruente com a dinâmica α.

(Se houver tempo ver como a baixas velocidades os contadores de impulsos podem não induzir uma partição congruente, das distância percorridas por cada roda).

Verificar que esta mesma análise semântica pode ser desenvolvida para um quadro de refinamento de sucessivas arquitecturas.

Esta página foi intencionalmente deixada em branco

7 ARQUITECTURA

7.1 Arquitectura implementável de controlo

A característica principal que diferencia uma arquitectura conceptual de uma arquitectura implementável, é que na primeira desenvolve-se o mapa do controlo do sistema, e na segunda desenvolvem-se os componentes para a execução do controlo.

O quadro conceptual definido anteriormente adequa-se facilmente ao mapeamento da arquitectura conceptual numa implementável. Entende-se como arquitectura implementável a modelização da organização e respectivos componentes que irão implementar as funcionalidades especificadas na arquitectura conceptual. Uma funcionalidade pode ser implementada em mais que um componente, e um componente pode incluir mais que uma funcionalidade.

Os modelos utilizados na abordagem conceptual, migram facilmente para uma abordagem computacional, e já possuem mecanismos que permitem modelar parcialmente aspectos que surgem durante a implementação.

A nossa proposta de arquitectura é um refinamento da arquitectura de [Silva 99], que é caracterizada fundamentalmente por uma modelização dos seus componentes e das relações que se podem estabelecer entre eles. A arquitectura originalmente apresentada por [Silva 99], se bem que apresente características que lhe permitem estabelecer procedimentos de verificação e síntese automática (ainda não o faz, mas o TTM em que esta é baseada permite desenvolvimento de alguns mecanismos), está fortemente orientada para os procedimentos de projecto e implementação.

Enquanto nas arquitecturas conceptuais a verificação ocorre para a especificação, ou seja o modelo verifica uma dada propriedade no modelo de lato nível, nas arquitecturas implementáveis também é necessário verificar se o modelo de baixo nível implementa ou satisfaz uma propriedade dependente da implementação. Tradicionalmente as propriedades são as de segurança e as de "*liveness*"

A arquitectura implementável é composta por três grandes sub-arquitecturas hierarquicamente organizada: Supervisão; Coordenação e Funcional, e em cada um destes níveis é possível definir hierarquias parciais entre os diferentes componentes.

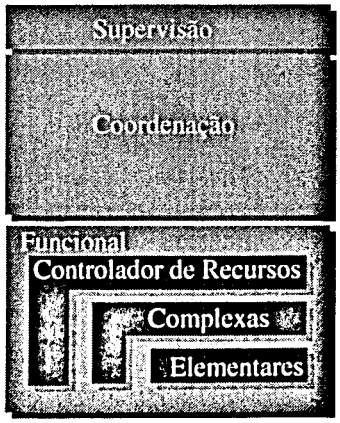


Figura 7.1-1

Modelo da arquitectura de um sistema autónomo

Fundamentalmente, o modelo de Arquitectura expressa-se por uma rede dinâmica de autómatos [Silva 99], em que a competência e a função do componente são definidos no contexto do nível hierárquico a que pertence.

A camada ***funcional*** consiste num conjunto organizado de entidades que suporta a execução das actividades que materializam a saída do sistema de decisão para o mundo físico ou para a representação que o sistema tem dele. A maioria destes componentes contribui para a implementação de um conjunto de leis de controlo parametrizadas pela coordenação que, baseadas em informação sensorial (sistema físico), produzem referências para os actuadores (sistema físico).

O nível de ***coordenação*** é composto fundamentalmente por um conjunto de autómatos "supervisores" e observadores discretos. Os observadores discretos tem por objectivo a acompanhar a trajectória da camada funcional e definir a trajectória do estado da coordenação. Por sua vez os supervisores "calculam" as referências (comandos) para a camada funcional. Estes "supervisores" são parametrizados pela camada ***supervisão*** da arquitectura.

A natureza claramente discreta da coordenação não se adapta facilmente à natureza predominantemente contínua do nível ***funcional***. Dado o nível relativamente mais abstracto da ***coordenação*** e a grande variedade de leis de controlo que é possível implementar, é necessária a introdução de entidades no nível funcional que garantam a interface entre a ***coordenação*** e o nível ***funcional***. É necessário garantir que:

(i) a coordenação seja independente dos tipos de leis de controlo ou controladores que se pretendam instalar.

(ii) seja ultrapassada a dificuldade associada ao facto dos controladores "contínuos" terem de interpretar directamente os comandos enviados pela coordenação

(iii) seja ultrapassada a incapacidade da coordenação interpretar directamente o estado da camada funcional.

Deste modo a interface ***coordenação*** camada ***funcional*** vai materializar-se através de uma entidade a que chamaremos controlador de recursos. Este assume um carácter estruturante, uma vez que uniformiza controladores e observadores.

Na arquitectura implementável surge um (novo) estágio hierárquico (superior), que se designa por supervisão, e sua principal missão é estabelecer a relação do sistema com o "utilizador". Fundamentalmente, vai permitir ao utilizador eliminar ou acrescentar missões, suspender tarefas e colaborar na resolução de alguns problemas que surjam na execução das missões.

Esta arquitectura (ou metodologia que dela resulta) apresenta uma grande variedade de modelos permitindo a escolha apropriada para as diferentes partes do sistema [Silva 99]. Os modelos dos componentes são caracterizados pela sua interface, que define o fluxo informativo e a natureza do modelo computacional.

Por exemplo a implementação de um PID (no nível funcional), pode ser realizada recorrendo a dois componentes (ver capitulo 5 e 7 de [Silva 99]), um ***filtro elementar*** (processamento sensorial), e um ***regulador***. Estes dois componentes implementam actividades periódicas e estruturalmente invariantes (na literatura, referenciados como "sample-data-systems"). Outra questão relevante é que os modelos propostas por [Silva 99], têm em conta o facto do controlo ser implementado num sistema digital e consequentemente o modelo adoptado para os componentes "embeberem" os modelos contínuos associados aos dispositivos físicos.

O facto da arquitectura apresentar vários modelos de componentes, permite facilmente o projecto e a modelização de sistemas a que a comunidade científica chama normalmente de sistemas híbridos.

O mapeamento entre as duas arquitecturas não é "linear". Na arquitectura conceptual, a hierarquia temporal e espacial confundem-se nos mesmos estágios hierárquicos e baseiam-se em objectivos e tarefas (manobras). Na implementável emerge uma "nova" organização, por um lado baseada na organização sistémica, na ordem de autoridade dos componentes e das propriedades, e a transformação da hierarquia temporal numa sequenciação de acontecimentos. Surge paralelamente uma nova função para a coordenação, que é a gestão da "causalidade" do comportamento do sistema. Enquanto na conceptual só focamos "no" comportamento que o sistema desenvolve (e "como" têm de o desenvolver), na execução focamos também nos "meios" e "como" e "quando" se mobiliza os meios para gerar o "como se têm do o desenvolver".

E surgem novas hierarquias, referentes aos componentes que implementam as funções definidas conceptualmente, a noção de "off-line" e "on-line" ou surge pela primeira vez ou os contornos temporais ficam mais claros.

A coordenação do ponto de vista conceptual vai mapear-se em dois estágios. A função de supervisão de manobras e a correspondente coordenação da missão, através da composição das tarefas (manobras) mapeia no nível de coordenação da arquitectura implementável. A componente estrutural vai ser mapeada no nível funcional da arquitectura, nas entidades definidas como controladores de recursos (ver definição nos pontos seguintes).

As funções do nível de organização da arquitectura conceptual são mapeadas no nível funcional da arquitectura implementável

O mapeamento da arquitectura conceptual (essencialmente orientada para o projecto de manobras de movimento) numa implementável orienta-se

com o objectivo que esta estrutura vai ser a responsável pela execução do controlo.

7.2 Algumas questões sobre a implementação de sistemas de controlo

Ser um sistema autónomo implica que é um "embedded system", mas ser um "embedded system", não implica ser um sistema autónomo.

Um sistema autónomo do ponto de vista da estrutura e da implementação dos seus componentes pode ser considerado um "embedded system". Mas neste caso existe um conjunto complexo de propriedades que necessitam de estar presentes no desempenho tais como segurança, "deadlock", "liveness", "fairness", fiabilidade, confiabilidade.

A imaturidade da área do projecto de sistemas "embedded" (autónomos), não facilita a escolha dos modelos computacionais a utilizar. Pode-se facilmente verificar que muitos dos exemplos de arquitectura de sistemas autónomos, por um lado não possuem uma referência explícita aos modelos utilizados, e por outro os que os referem, não adoptam um modelo de referência, mas recorrem a diversos modelos de computação. O processo "construtivo" destes sistemas ainda possui contornos pouco definidos, no entanto independentemente dos modelos adoptados existem algumas ideias que são comuns à maioria das "implementações",

Tipicamente os sistemas autónomos são caracterizados por um conjunto de processos concorrentes e coordenados. Alguns destes interagem com o ambiente, evoluem de uma forma independente e comunicam entre si. O fenómeno da comunicação entre estes processos pode ocorrer de uma forma implícita quando partilham parte do estado do sistema, ou explicitamente quando um dos processos informa os outros sobre alguns aspectos parciais do seu próprio estado.

A adopção de um sistema operativo tempo-real, possui mecanismos que facilitam lidar como o "interleaving"

Os sistemas autónomos são tempo-real, e consequentemente é tão importante o instante (o tempo) em que uma dada informação é produzida, como o conteúdo dessa informação. Quando surge a implementação dos processos a concorrência pode ocorrer pela combinação das seguintes situações: paralelismo físico através do uso de diferentes recursos computacionais, ou "interleaving" quando partilham o mesmo recurso computacional.

Os subsistemas que estão implementados em diferentes e paralelos recursos computacionais usufruem da mesma noção de "tempo", e os instantes

a que os acontecimentos ocorrem num subsistema possuem uma ordem natural relativamente aos instantes a que outros acontecimentos ocorrem noutro subsistema. Mas quando falamos de subsistema lógicos que partilham o mesmo recurso computacional, o mecanismo de "interleaving" cria uma outra ordem que depende da política de sequenciamento dos processos e da carga computacional. Sem formalizar esta questão o projectista resolve substituindo a ordenação total dos acontecimentos por um ordenação parcial, recorrendo a mecanismos de sincronismo/comunicação (uso de semáforos "messages queues" entre processos no Po-Robot) ordenando os acontecimentos que são importantes para um dado processo.

Existem algumas estratégias para gerir a ordem dos acontecimentos, mas lembremos aqui as mais comuns, e que, caso os Sistemas Operativos Tempo Real não possuam já implementadas, possuem mecanismos suficientes para a sua fácil implementação:

- o "lock" do ciclo de "read-modify-write" de estruturas de dados partilhadas (utilizado no Po-Robot, através do mecanismo de "LOCK" do VxWorks).
- o "umbounded" FIFO que permite a ordenação total dos acontecimentos no processo que consome a informação. Quando a política de sequenciamento fornece o CPU a este processo, este lê o buffer totalmente e só depois produz a saída. (utilizado no Po-Robot, através dos mecanismos do VxWorks).
- o "bounded" FIFO, nesta situação fica limitado o número de acontecimentos que um sistema possa consumir sem produzir uma saída.
- o Rendezvous pode ter várias versões, mas recorrendo a um mecanismo tipo monitores ou semáforos é possível que as leituras e escritas de dois ou mais processos ocorram na ordem natural relativamente a acontecimentos em vários processos em simultâneo (utilizado no Po-Robot, através dos mecanismos de semáforos do VxWorks).

7.3 Introdução ao Nível Funcional

O nível funcional constitui o grupo de funcionalidades que realizam o "interface" entre o sistema de decisão e o mundo físico. Do ponto de vista dos níveis superiores da arquitectura, estas funcionalidades são associadas em grupos. Estes grupos funcionais (navegação, controlo de movimento etc.) estão naturalmente associados aos recursos sensoriais, actuadores ou de cálculo. Existindo relações de interdependência entre estes, torna-se necessário modelar as relações que se estabelecem entre os diversos grupos.

A materialização da decisão e sua consequência no mundo físico expressa-se através de uma actividade (sequências de alterações das variáveis de estado observáveis do sistema ou do mundo). O nível funcional é a estrutura operacional responsável pelo desenvolvimento da actividade sendo a sua parametrização da responsabilidade da coordenação. A capacidade reactiva é garantida pelo nível funcional, sendo a reactividade do sistema reconfigurável e cada configuração dependente dos anéis de realimentação estabelecidos entre os diversos componentes do nível funcional.

Antes de apresentarmos o modelo da camada funcional vamos revisitar alguns requisitos gerais de controlo:

possibilidade de malhas de controlo alternativas,
possibilidade de algoritmos de controlo orientados para estratégias sensoriais.

A implementação destes dois requisitos implica que o sistema seja capaz de configurar em "tempo-real" as ligações funcionais.

O tratamento da diversidade dos grupos funcionais leva à necessidade de uniformizar os aspectos estruturais e comportamentais das funcionalidades. Neste sentido, apresenta-se um modelo em que estas são definidas abstractamente de forma hierárquica. Este modelo descreve a evolução de actividades espacial e temporalmente disjuntas, garantido o princípio da composibilidade. A abstracção de funcionalidades em sucessivos níveis, conjugada com a possibilidade (necessidade) de activação e configuração das mesmas, permite dinamicamente alterar a camada funcional.

A camada Funcional é constituída por uma rede dinâmica de autómatos organizados em três categorias [Silva 99]:

(i) Controladores de Recursos.
(ii) Funcionalidades Complexas.
(iii) Funcionalidades Elementares.

As ***Funcionalidades Elementares*** processam informação directamente sobre o nível de abstracção.

As ***Funcionalidades Complexas*** conjugam, agregam ou produzem informação, de e para componentes de natureza distintas. Podem ainda activar ou desactivar outras funcionalidades (de qualquer categoria) de forma paralela e/ou sequencial.

Os ***Controladores dos Recursos*** surgem geralmente no topo da hierarquia das funcionalidades. No entanto, é possível encontrar estas entidades hierarquicamente abaixo das funcionalidades complexas. Quando no

Recurso é uma estrutura hierárquica de processos dinamicamente reconfigurável por forma que este execute a actividade requerida por uma dada tarefa.

topo da hierarquia, os controladores de recursos realizam o interface com o sistema de coordenação. Também compete aos controladores de recursos em primeira-mão" a gestão da evolução da camada funcional.

7.3.1 Componente da camada funcional

Um componente do estágio funcional é globalmente definido por:

(i) Interface com um ou mais componentes de um nível hierárquico superior (coordenação ou do próprio nível funcional mas em que existe uma agregação de dinâmica);

(ii) Interface com um ou mais componentes contextuais (com componentes em que existe uma agregação de estado);

(iii) Interface com um ou mais componentes do nível hierárquico inferior (decomposição dinâmica);

(iv) um modelo de transições temporizadas que descreve actividade da funcionalidade implementada.

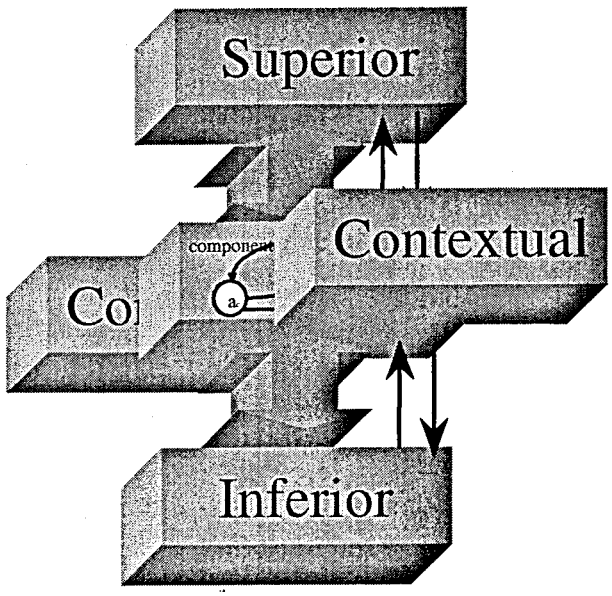


Figura 7.3-1

Modelo genérico de um componente do nível funcional.

.partilha de variáveis

sincronização de acontecimentos

O interface para os componentes da rede de autómatos é independente do seu posicionamento hierárquico e constituído por conjuntos de:

- variáveis que permitem a partilha de informação entre os componentes;
- mensagens que indicam quais os acontecimentos observáveis e controláveis que irão sincronizar com outros componentes.

Nem todos os componentes vão possuir ligações aos diversos níveis hierárquicos, dependendo estas do tipo de componente que se vai desenvolver.

Tabela 7.3-1

Mapa das possíveis ligações de um componente do nível funcional.

	agregação dinâmica entrada	agregação dinâmica saída
variáveis	√	√
acontecimentos	√	√

	decomposição dinâmica entrada	decomposição dinâmica saída
variáveis	√	√
acontecimentos	√	√

	agregação/decomposição de estado entrada	agregação/decomposição de estado saída
variáveis	√	√
acontecimentos	√	√

O interface parece ser redundante face à definição do autómato (é na vertente conceptual), no entanto do ponto de vista da metodologia do projecto, permite verificar o resultado da função, bem como as ligações entre componentes.

A definição do interface indica ao implementador quais a variáveis que deve partilhar para os outros processos, quais as variáveis que estão disponíveis pelos outros componentes, bem como os acontecimentos que é necessário partilhar com terceiros. Metodologicamente as definições devem ocorrer dos níveis inferiores para os superiores (ao contrario os requisitos devem ocorrer dos superiores para os inferiores). As ligações entre autómatos podem ser entre vários autómatos, e as sincronizações podem ser ocorrer em múltiplos autómatos. O interface ajuda a "unificar" o modelo conceptual da actividade e a implementação computacional.

A semântica dos acontecimentos que assinalam as transições na actividade dos processos é local (e deve ser assim modelado). Contudo quando os acontecimentos que assinalam as transições da actividade são partilhados e sincronizam com terceiros processos, a semântica resultante da composição adquire significados mais ricos.

7.3.2 Funcionalidades Elementares

Os componentes elementares realizam na maioria dos casos o interface com o estágio abstracto e disponibilizam o estado da plataforma para a arquitectura de controlo. Implementam actividades periódicas e estruturalmente invariantes, ou seja, a execução destas não depende de uma condição. Como. consequência o tempo de execução de cada uma das actividades deste tipo de componente é constante. Tipicamente, estes componentes não possuem autonomia para finalizarem a actividade, significando que o acontecimento que termina a actividade é gerado externamente. Estes componentes executam sincronamente dependentes da transição de um relógio (frequência fixa) ou dependentes de um acontecimento externo (frequência variável).

Definimos três tipos de componentes elementares, os filtros elementares, os reguladores e as memórias.

Filtros elementares

Os filtros elementares são fundamentalmente integradores de informação de vários formatos para um formato mais abstracto (como exemplo ver o processo hodometria). Como é notório na figura seguinte promovem o fluxo de informação dos níveis hierarquicamente inferiores para os níveis hierarquicamente superiores.

Exemplo de filtros, os processos (ver capítulo 7 em [Silva 99]):
pára-choques; hodometria; acelerómetros; bússola; sensor acústico; detector do tapete; botão de pressão; sonar rotativos.

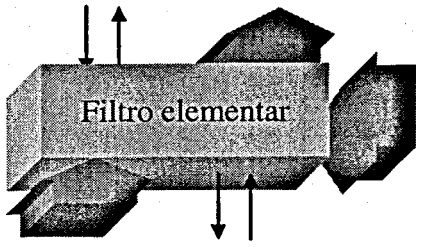


Figura 7.3-2

Filtro elementar.

Vista do exterior do componente.

Estes componentes são a base do processamento sensorial. A actividade de cada um destes é dependente da aplicação sendo no entanto modelada segundo a definição exposta em 4.2.3. O interface destes componentes é globalmente caracterizado pelo mapa esboçado na tabela seguinte.

	Agregação Dinâmica entrada	Saída
variáveis	**N.A.**	√
acontecimentos	√	√

	agregação/decomposição de estado	
	entrada	saída
variáveis	√	√
acontecimentos	**N.A**	**N.A**

	decomposição dinâmica	
	entrada	Saída
variáveis	√	**N.A**
acontecimentos	√	√

Tabela 7.3-2

Mapa da interface dos filtros elementares com os outros processos.

Sem prejuízo de outras mensagens inerentes ao processo que o filtro elementar implementa, estes são caracterizados por mensagens do tipo:

Entrada	Saída
Inicio do processo que executa a actividade do filtro	Nova informação
Transição do relógio	
Mensagem de sincronização de processos	
Fim do processo que executa a actividade do filtro	

Reguladores

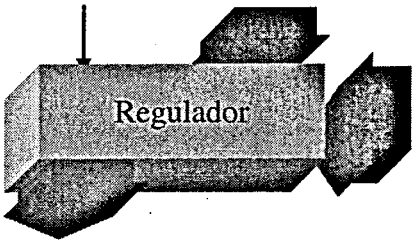
Os reguladores como o próprio nome indica desempenham a função de controlador de um recurso físico bem específico (como exemplo ver o processo PID). Contrariamente aos filtros elementares promovem o fluxo de informação dos níveis hierarquicamente superiores para os níveis hierarquicamente inferiores

Exemplo de reguladores (ver capítulo 7), os processos:
PID; travão; Motor de transferência de carga.

Figura 7.3-3

Regulador.

Vista do exterior do componente.



A conjugação dos reguladores com os filtros elementares, e as memórias vai permitir estabelecer os primeiros anéis de realimentação. Como exemplo tem-se o processo PID (ver capitulo 7.3.1.8) e o processo hodometria (ver capitulo 7.3.1.2). Através do processo hodometria calculamos (entre outras coisas) a velocidade instantânea de cada motor, que vai ser utilizado como sinal de "feedback" para o controlo de velocidade dos motores. A taxa de actualização dos valores de velocidade vai ser utilizada como taxa de amostragem para o controlo dos motores permitindo controlar a velocidade dos motores de tracção. A interface dos reguladores é globalmente caracterizado pelo mapa esboçado na tabela seguinte.

Tabela 7.3-3

Mapa da interface dos filtros elementares com os outros processos.

	agregação dinâmica	
	entrada	saída
variáveis	√	**N.A.**
acontecimentos	√	√

	decomposição dinâmica	
	entrada	saída
variáveis	**N.A**	√
acontecimentos	**N.A**	**N.A**

	agregação/decomposição de estado	
	entrada	saída
variáveis	√	√
acontecimentos	√	**N.A**

Sem prejuízo de outras mensagens inerentes ao processo que o regulador implementa, estes são caracterizados por mensagens do tipo:

Entrada	Saída
Inicio do processo que executa a actividade do regulador	Novo "set point"
Transição do relógio	
Mensagem de sincronização de processos	
Fim do processo que executa a actividade do regulador	
Nova referência	

Memórias

Exemplos de memórias: Memória posição geométrica.

As memórias são componentes que guardam e organizam informação na arquitectura de controlo para que esta relacione o sistema com o mundo exterior. Parte desta informação pode ser obtida pelo estado do sistema, outra há que necessita de ser expressa explicitamente. Normalmente memórias são vistas como componentes estáticos onde se "armazena" dados para serem utilizados posteriormente. Na arquitectura apresentada, memória assume um papel mais dinâmico pois para além da definição dos atributos e natureza da

informação que se pretende guardar, define-se a forma como a confiança nessa informação evolui. Este factor é muito importante nos sistema autónomos pois o ambiente evolui não só pela a acção do sistema mas também pela interacção de todos os sistemas exteriores ao nosso. A forma como a confiança evolui vai ser modelada por um TTM. A interface das memórias é globalmente caracterizado pelo mapa esboçado na Tabela 7.3-4

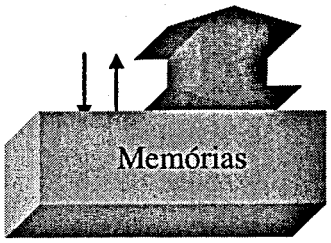


Figura 7.3-4

Memória, módulo elementar. Vista do exterior do componente.

	agregação dinâmica entrada	agregação dinâmica saída
variáveis	√	√.
acontecimentos	√	√

	decomposição dinâmica entrada	decomposição dinâmica saída
variáveis	N.A	N.A
acontecimentos	N.A	N.A

	agregação/decomposição de estado entrada	agregação/decomposição de estado saída
variáveis	N.A	N.A
acontecimentos	N.A	N.A

Tabela 7.3-4

Mapa da interface dos filtros elementares com os outros processos

7.3.3 Funcionalidades Complexas.

Estes componentes são os responsáveis pela capacidade de percepção e pela causalidade do sistema. A capacidade de percepção significa que o sistema, através dos estímulos (externos e/ou internos) processados nos estágios inferiores, é capaz de distinguir. Estes componentes implementam actividades fundamentalmente assíncronas e variantes, ou seja, o tempo de execução das actividades deste tipo de componentes é variável dependendo do estado do sistema.

Os componentes complexos são fundamentalmente compostos por ***sensores virtuais*** e ***servidores***, e podem ser associados em sucessivas camadas hierárquicas. Estes componentes implementam os comportamentos complexos que são obtidos por combinação de actividades geradas por componentes do tipo servidores. A complexidade pode aumentar através da consideração sucessivas camadas de abstracção

Sensores Virtuais

Os sensores virtuais são processos que observam o estado lógico da plataforma. Chamam-se virtuais porque não lhes corresponde directamente nenhum dispositivo físico. São organizados em três classes:

- filtros discretos que podem ser vistos como classificadores do estado;
- filtros contínuos que validam informação de variáveis continuas;
- alarmes que determinam a ocorrência de um dado acontecimento,

Filtros Discretos

Exemplo filtro discreto: processo de Localização topológica.

Os filtros discretos desempenham funções de classificadores, em que com base na observação parcial do estado do sistema o classificam, atribuindo essa classificação a uma variável (discreta) destinada para o efeito. A actividade tal como nos casos anteriores é modelada por um TTM dependente da aplicação coerente com o interface que é globalmente caracterizado pelo mapa da figura seguinte.

Figura 7.3-5

Filtro Discreto

Vista exterior do componente.

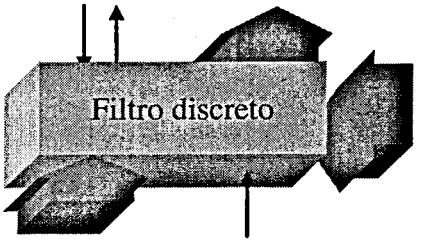


Tabela 7.3-5

Mapa da interface dos filtros elementares com os outros processos.

	agregação dinâmica entrada	agregação dinâmica saída
variáveis	N.A.	√
acontecimentos	√	√

	agregação/decomposição de estado entrada	agregação/decomposição de estado saída
variáveis	√	√
acontecimentos	N.A	N.A

	decomposição dinâmica entrada	decomposição dinâmica saída
variáveis	√	N.A
acontecimentos	√	√

Todos os filtros discretos são caracterizados por mensagens do tipo:

Entrada	Saída
Inicio do processo que executa a actividade do filtro	Nova informação
Nova informação outros componentes	
Mensagem de sincronização de processos	
Fim do processo que executa a actividade do filtro	

Filtros Contínuos

Exemplo filtro contínuo: localização geométrica, baseada em filtro de Kalman, ou um processo de fusão sensorial.

Os filtros contínuos são componentes idênticos aos já definidos nas actividades elementares, com a particularidade da actividade de processamento não ser determinística, e como tal o tempo de processamento é variável. Em termos de modelização vão ser iguais aos filtros discretos diferindo na natureza das variáveis que exportam para os níveis superiores,

que neste caso vão ser contínuas. O interface destes processo é caracterizado pelo mapa esboçado nas figuras seguintes, compatível com o TTM que modela a actividade.

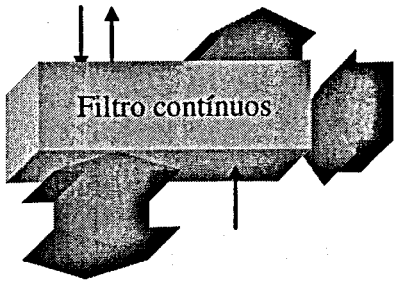


Figura 7.3-6

Filtro Contínuo.

Vista exterior do componente.

	agregação dinâmica entrada	saída
variáveis	N.A.	√
acontecimentos	√	√

	decomposição dinâmica entrada	saída
variáveis	√	√
acontecimentos	√	√

	agregação/decomposição de estado entrada	saída
variáveis	√	√
acontecimentos	N.A	N.A

Tabela 7.3-6

Mapa da interface dos filtros contínuos com os outros processos.

Alarmes

Os alarmes são processos que determinam se a trajectória do estado da plataforma atingiu um subconjunto particular do espaço dos estados. Quando ocorre a transição que conduz o sistema a esse subconjunto particular, então este processo assinala essa transição, permitindo que outros processos sincronizem com esse acontecimento. Os alarmes são globalmente caracterizados pela figura seguinte.

Exemplo de alarmes: Sensor virtual assinalando botão carregado

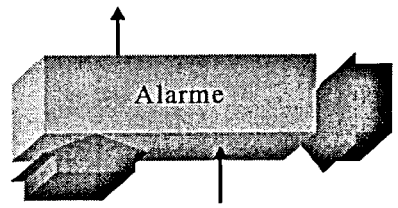


Figura 7.3-7

Alarme, vista exterior do componente.

	agregação dinâmica entrada	saída
variáveis	N.A.	N.A.
acontecimentos	N.A.	√

	decomposição dinâmica entrada	saída
variáveis	√	N.A.
acontecimentos	√	N.A.

	agregação/decomposição de estado entrada	saída
variáveis	√	N.A.
acontecimentos	N.A	N.A

Tabela 7.3-7

Mapa da interface dos filtros contínuos com os outros processos.

Todos os alarmes são caracterizados por mensagens do tipo:

Entrada	Saída
Inicio do processo que executa a actividade do alarme	Alarme
Mensagem de sincronização de processos	
Fim do processo que executa a actividade do alarme	

Servidores

Os servidores são funcionalidades assíncronas que controlam a actividade global do robot. Do desempenho destas funcionalidades resulta, ou uma intervenção física do sistema no mundo ou uma lista de futuras intervenções (planos de actividades). As actividades implementadas por estes processos são os elementos básicos para o sistema de decisão.

Os servidores são organizados em duas classes:

- estratégias de controlo,
- calculadores.

Estratégias de controlo

Exemplo:
Estratégia de controlo movimento tipo linha e curva,

As estratégias de controlo são processos que, através da observação das actividades dos outros, e de um objectivo que lhe é requisitado (um dos objectivos que o sistema é suposto atingir) definem a actuação por forma a manter a trajectória do estado do sistema num determinado conjunto. O interface destes processos é caracterizado pelo mapa expresso nas figuras seguintes.

Figura 7.3-8

Estratégias de controlo, Vista exterior do componente

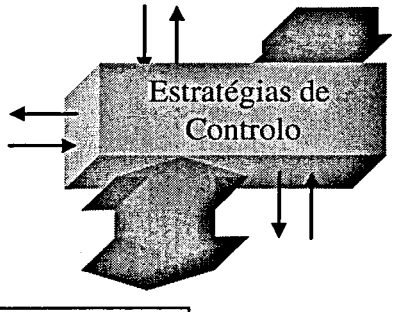


Tabela 7.3-8

Mapa da interface dos processos estratégias de controlo com os outros processos.

	agregação dinâmica entrada	agregação dinâmica saída
variáveis	√	N.A.
acontecimentos	√	√

	agregação/decomposição de estado entrada	agregação/decomposição de estado saída
variáveis	N.A.	N.A.
acontecimentos	√	√

	decomposição dinâmica entrada	decomposição dinâmica saída
variáveis	√	√
acontecimentos	√	√

Calculadores

Exemplo:
processo de planeamento de movimento.

Do resultado da acção dos calculadores não resulta alterações na relação que o sistema possui com o mundo exterior. Da actividade destes

resulta, ou valores para a parametrização dos outros componentes, ou listas de objectivos que o sistema deve atingir, ou planos de actividade para o sistema. Se bem que no esquema que ilustra a vista externa do componente seja indicado a partilha de variáveis com os níveis inferiores, esta só ocorre com componentes do tipo ***memória***.

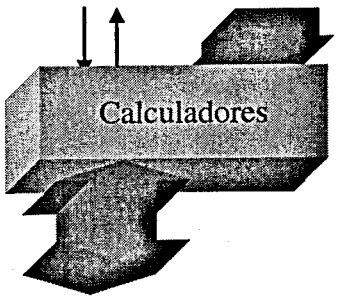


Figura 7.3-9

Calculador.

	agregação dinâmica entrada	saída
variáveis	√	N.A.
acontecimentos	√	√

	decomposição dinâmica entrada	saída
variáveis	√	√
acontecimentos	N.A.	N.A.

	agregação/decomposição de estado entrada	saída
variáveis	N.A.	N.A.
acontecimentos	N.A.	N.A.

Tabela 7.3-9

Mapa da interface dos processos calculadores com os outros processos.

7.3.4 Controladores de recursos

Recurso é uma estrutura hierárquica de processos, dinamicamente reconfigurável, que executa um conjunto de acções de forma a levar a cabo uma actividade elementar. Os componentes (processos) que integram a estrutura pertencem às classes referidas anteriormente. O recurso é externamente caracterizado pelas actividades elementares que lhe estão associadas.

Controlador de recursos é o processo que garante que só as actividades elementares que estão definidas segundo os princípios da arquitectura, é que são executadas pelo recurso. Quando, no topo da hierarquia do nível funcional realiza o interface com a coordenação,

relembrar que à actividade está associado um segmento de trajectória do estado ao qual corresponde um sequenciamento de acontecimentos bem definidos.

Até este ponto definimos as entidades da arquitectura de uma forma muita abstracta, consequência das funcionalidades até agora endereçadas serem totalmente dependentes da aplicação e da instância da arquitectura. No entanto com a introdução da noção recurso e o respectivo controlador, fica claro que as actividades que cada recurso desenvolve estão dependentes da arquitectura pelo efeito do controlador. Desta forma há mecanismos de decisão

que surgem embebidos no nível funcional pela estrutura que os controladores de recursos impõem.

o projectista fica limitado por um lado pelos controladores de recursos, por outro pela aplicação.

Uma vez que os controladores de recursos dependem da arquitectura, e por consequência já estão desenhados no início do projecto do sistema autónomo, estes vão servir de orientação e metodologia para o desenvolvimento do nível funcional,

Os recursos e controladores de recursos de um sistema autónomo podem ser vistos em quatro grandes classes:

- recursos energéticos;
- recursos computacionais,
- recursos mecânicos ou sensoriais e;
- recursos lógicos

A definição dos recursos lógicos, foram originalmente desenvolvidos em [Silva 99], no entanto vamos apresentar de forma mais refinada aqui a definição do recurso mecânico necessário à compreensão do exemplo do capítulo 8.

Recursos Mecânicos ou Sensoriais

O objectivo principal de tipificação dos controladores deste tipo de recursos é a uniformização do interface com a camada de coordenação. Genericamente, um recurso mecânico é um dispositivo composto por componentes físicos móveis que, associados a componentes electromecânicos, electrónicos e computacionais, executam movimentos no domínio de actuação.

relembrar definição de tarefas elementares, capitulo 4.2.5.

Os recursos mecânicos e sensoriais podem ser organizados em duas classes. Uma classe denotada ***1C1SnA*** (1 cliente, 1 servidor, n actividades), que significa que este recurso esta habilitado a executar uma actividade de cada vez para uma única tarefa, de um conjunto de actividades (tarefas elementares) possíveis. O controlador deste recurso vai ser projectado para permitir a utilização de várias estratégias de controlo ou sensoriais (exemplo são as estratégias de movimentação de dispositivos mecânicos, ou esquemas de fusão sensorial). Uma segunda classe denotada ***nC1S1A*** (n clientes, 1 servidor, 1 actividade), que significa que este recurso está habilitado a executar uma única actividade que no entanto pode ser parametrizada para servir vários clientes (exemplo são o caso do gestor dos sensores de ultra-sons).

Classe: 1C1SnA.

Exemplos:
Gestor de recursos movimento, ver capitulo 7.3.3.3.
Gestor do recursos de transferência de carga, ver capitulo 7.3.3.4
Gestor de recurso navegação Global, ver capitulo 7.3.3.5.

Este tipo de recursos podem desenvolver várias actividades, embora em cada momento, uma só tarefa possa ser cliente do recurso. O projecto do controlador deste tipo de recursos tem por objectivo a implementação de múltiplas actividades sem alterar o controlador básico. A génese deste controlador nasceu da necessidade de responder ao sucessivo aumento da complexidade da movimentação da plataforma [Almeida 97], [Po-Robot 97] sendo posteriormente os conceitos estendidos para outro tipo de recursos. Embora a nomenclatura usada esteja fortemente ligada à movimentação, a concepção deste controlador é bem mais abrangente.

O exemplo mais comum do nosso dia a dia para esta noção é o do estacionamento do automóvel.

Os movimentos (ou as estratégias sensoriais) podem ser organizadas em conjuntos de manobras [Almeida 96] que representam movimentações simples, ou complexas.

A noção de manobra da qual fazemos apelo é a mesma do senso comum, ou seja, uma composição de movimentos elementares por forma a atingir um dado objectivo.

No caso de dispositivos sensoriais, de apoio à navegação, podemos analisar as questões na seguinte perspectiva. Os sensores utilizados têm limitações quer a nível espectral quer a nível de amplitude dos sinais usados. Deste modo, uma única lei de processamento sensorial é inviável ou pelo manos não é suficientemente robusta para que em todos os instantes esta respondesse ao objectivo pretendido. No entanto já um conjunto de leis de processamento que, evocadas no momento preciso, poderiam no seu conjunto corresponder de forma satisfatória.

Uma vez que a manobra pode compreender mais que um movimento (processamento sensorial), poderemos pensar que manobra define uma estratégia híbrida.

O controlador associado ao recurso 1C1SnA é caracterizado por aceitar como referência uma manobra ou conjunto de manobras. De uma forma mais precisa podemos dizer que a "Manobra" poderá ser representada por um conjunto de estados discretos, estando associada a cada um deles as actividades necessárias à implementação de leis de controlo ou processamento sensorial, e um conjunto de transições.

Os estados discretos são caracterizados por realizarem actividades tais como a inicialização de uma manobra, o "fecho" de uma manobra quando esta termina com sucesso, o cálculo dos parâmetros iniciais da manobra, o calculo

da referência e testar as condições de transição e activar a referência no actuador apropriado.

O "interface" para a coordenação é constituído por um conjunto de mensagens cuja a semântica deverá ser equivalente às listadas seguidamente

Tabela 7.3-10

Interface com o nível de coordenação.

Entrada	Saída
Pronto	
	inicia_manobra
Suspender	
Abortar	
Continuar	
STOP	
Novas_manobras	
	falhou_manobra
	fim_de_manobra/nova_manobra

Exemplos:
processo estratégias de controlo de movimento, recta – curva, ver 7.3.2.6 estratégia de controlo);
processo estratégias de navegação, ver 7.3.2.7 filtro).

O interface para os componentes de níveis hierárquicos inferiores é constituído por mensagens definidas nos componentes objecto da interacção, e que são ou do tipo ***Estratégias de Controlo*** ou do tipo ***Filtros.***

Modelo de transições temporizadas

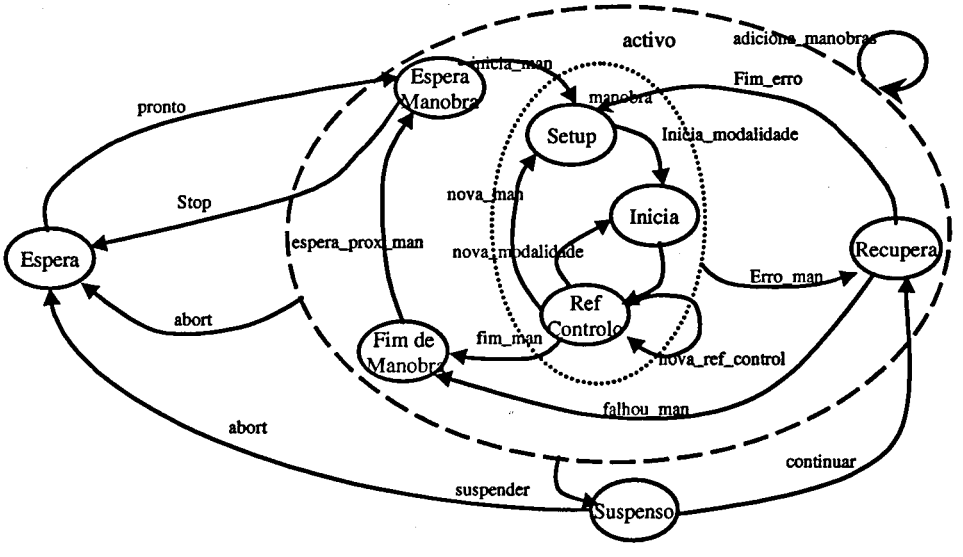
Variáveis

$act_1CnA1S \in \{$ nao_activa, espera, espera_manobra, fim_de_manobra, setup, init, Ref_controlo, suspenso, recupera $\}$	variável que guarda o estado do sistema quanto à actividade de gestão do recurso computacional
manobras	estrutura que armazena as manobras activas

Transições

Figura 7.3-10

Diagrama de transições do controlador de recurso da classe 1C1SnA.



	Condição de activação	Transformação	Inferior	Superior
inicia	(*act_1C1SnA* = nao activa)	[*act_1CnA1S* := *espera*; *inicializa(Manobras)*;]	0	∞
pronto	(*act_1C1SnA* = espera)	[*act_1CnA1S* := *espera_manobra*;]	0	∞
inicia_man	(*act_1C1SnA* = **espera_manobra**) ∧ (**Manobras!** = **vazio**)	[*act_1CnA1S* := *setup*;]	0	0
inicia_modal	(*act_1C1SnA* = Setup)	[*act_1CnA1S* := *inicia*;]	0	tms
nova_modal	(*act_1C1SnA* = **Ref_controlo**) ∧ (**Manobras!** = **vazio**)	[*act_1CnA1S* := *inicia*;]	0	tmexlc
ref_control	(*act_1C1SnA* = inicia)	[*act_1CnA1S* := **Ref_control**;]	0	tinicia
nova_ref_cont	(*act_1C1SnA* = Ref_controlo)	[*act_1CnA1S* := **Ref_control**;]	0	tmexlc
erro_man	(*act_1C1SnA* ∈ {Setup, inicia, Ref_controlo})	[*act_1CnA1S* := *Recupera*;]	0	tmexlc
nova_man	(*act_1C1SnA* = **Ref_controlo**) ∧ (**Manobras!** = **vazio**)	[*act_1CnA1S* := *setup*;]	0	tmexlc
fim_erro	(*act_1C1SnA* = Recupera)	[*act_1CnA1S* := *setup*;]	0	temprec
continuar	(*act_1C1SnA* = Repouso)	[*act_1CnA1S* := *Recupera*;]	0	∞
suspender	(*act_1C1SnA* ∈ {**Setup, inicia, Ref_controlo, espera_manobra, fim_manobra, Recurpera**})	[*act_1CnA1S* := *Suspenso*;]	0	∞
Falhou_manobra	(*act_1C1SnA* = Recupera)	[*act_1CnA1S* := *fim_manobra*;]	0	temprec
abort	(*act_1C1SnA* ∈ {**Setup, inicia, Ref_controlo, espera_manobra, fim_manobra, Recurpera**})	[*act_1CnA1S* := *espera*, *Manobras* := 0]	0	###
esp_prox_man	(*act_1C1SnA* = **fim_manobra**)	[*act_1CnA1S* := *espera_manobra*;]	0	###
STOP	(*act_1C1SnA* = **espera_manobra**) ∧ (**Manobras** = **vazio**)	[*act_1CnA1S* := *espera*]	0	###
fim_manobra	(*act_1C1SnA* = **Ref_controlo**) ∧ (**Manobras** = **vazio**)	[*act_1CnA1S* := *espera_manobras*;]	0	tmexlc
adiciona_man	(*act_1C1SnA* ∉ {**não_activa, suspensa**})	[*actualiza(Manobras)*;]	0	###

Tabela 7.3-11

Diagrama de transições do controlador de recurso da classe 1C1SnA.

As actividades deste recurso caracterizam-se por vários estados alguns deles equivalentes às actividades inerentes à definição de manobra.

não_activo – este estado caracteriza a não actividade total do recurso do sistema.

espera – o controlador aguarda ordem da coordenação para passar ao estado de pronto. Durante este estado é possível ao recurso receber manobras. No entanto, nada ocorrerá enquanto o sistema permanecer neste estado.

espera_manobra – este estado indica que o sistema está pronto a iniciar a tarefa que for requisitada. Após a inserção de uma (ou um pacote de) manobra no buffer o controlador de recursos deverá disparar o acontecimento *iniciar_manobra*.

setup – esta actividade consiste em configurar os dispositivos necessários por forma a viabilizar a execução da manobra. Esta actividade tem um limite

tms: tempo máximo de setup
tmexlc: tempo máximo esperado para a execução da referência activa
temprec: tempo máximo esperado para a recuperação

temporal, permitindo assim determinar falhas nas atribuições aos dispositivos. Quando, para além de atrasos no processo de **setup** este sofre um erro, transita para o estado de **recuperação**

init – quando se activa uma lei de controlo (processamento sensorial), é necessário efectuar cálculos iniciais tais como constantes de matrizes, que vão ser comuns a todos os ciclos de controlo.

ref_control – a actividade deste este estado em conjunto com o estado **init** vai ter tantas instâncias quantas as leis de controlo (ou processamento sensorial), que compõe a manobra e vai ser caracterizada pelas actividades intrínsecas à manobra já descrita.

fim_de_manobra – este estado corresponde à actividade de finalizar uma manobra, já descrito na definição de manobra.

recuperação – sempre que uma manobra detecta um falha na sua execução, esta entidade executa uma actividade que tenta recuperar o estado para que a lei de controlo ou processamento seja retomado.

suspenso – neste estado o recurso encontra-se com actividade suspensa e o controlador espera uma ordem da coordenação para este, ou retomar a tarefa ou abortar a execução.

Os acontecimentos observáveis do recurso são:

inicia – este acontecimento é consequência da activação do sistema realizada pelo operador, transferindo o recurso do sistema para o estado espera.

pronto – este acontecimento é gerado na coordenação e informa o controlador da disponibilidade de execução de manobras.

inicia_man – a partir do momento em que o recurso esteja no estado de pronto, quando o buffer de manobras não estiver vazio, então o controlador de recursos deve disparar este acontecimento e o sistema dará inicio à execução de uma manobra.

inicia_modal – após a actividade de setup da manobra verificar ou configurar o recurso, dará início à execução da mesma originando o acontecimento que transita a manobra para a primeira lei de controlo activa, para a actividade de inicia.

nova_modal – quando a aplicabilidade de uma lei de controlo não é mais possível, então a manobra ou terminou (transita para Fim da manobra) ou transitou para uma nova lei de controlo caso não haja erros durante a execução.

ref_control – quando o cálculos das constantes de uma lei de controlo se encontram efectuados, então o sistema está prestes a transitar para a actividade que calcula e impõe a referência aos processos actuadores, originando este acontecimento.

nova_ref_cont – durante a execução da lei de controlo, esta verificou que devia alterar a referência, então os novos ciclos de controlo devem ser executados com novos valores.

erro_man- este acontecimento ocorre sempre que uma manobra detecta situações que não permitem a continuação da actividade da manobra, sendo necessário corrigir o estado por forma a que este permita a continuação da actividade, nesta situação dá-se a transição para o estado de Recupera.

nova_man – quando uma manobra terminou mas o buffer de manobras não está vazio e existe uma manobra aplicável então o sistema vai iniciar a execução de uma nova manobra, este acontecimento transita o sistema para o estado do setup.

fim_erro – a actividade de recuperação local ao recurso foi sucedida, então o sistema vai voltar à execução da manobra.

continua – este acontecimento é gerado na coordenação, e indica ao controlador de recursos que deve continuar com execução da manobra, após esta ter sido suspensa.

suspender – este acontecimento é gerado na coordenação para suspender temporariamente a actividade.

recupera_fim – este acontecimento é originado na actividade recupera, e indica que esta não conseguiu superar a dificuldade encontrada, e sistema para a execução da manobra e fica pronto para a execução de novas manobras.

abort – este acontecimento é gerado pela coordenação e o controlador do recurso deve promover o sincronismo com todos os módulos do recurso.

esp_prox_man– quando uma manobra terminou mas o buffer de manobras está vazio ou não existe uma manobra aplicável então o sistema vai esperar por novo conjunto de manobras.

STOP – este acontecimento é gerado pela coordenação e provoca a transição do controlador de recursos para o estado de repouso.

fim_manobra – quando uma manobra terminou a sua actividade e esta atingir o objectivo que estava destinado, então deve ser iniciada a actividade fim de manobra.

adiciona_mano – este acontecimento é gerado pela coordenação e indica que ao buffer de manobras foi adicionada uma ou um conjunto de manobras.

Classe: nC1S1A.

Este tipo de recursos só realiza uma actividade continuamente. No entanto, pode ser parametrizada para que possa atender vários clientes. Embora tendo que satisfazer requisitos mais sofisticados, este tipo de recursos evita a duplicação de meios (existem situações em que a duplicação não é possível, caso de sistemas sensoriais activos.) ou perda de desempenho. A

parametrização pode ser possível ou não dependendo dos requisitos dos clientes que estão a requerer o recurso.

Exemplo: este tipo de recursos são os gestores de sonares ou infravermelhos.

O interface para os componentes hierarquicamente superiores (pode não ser a coordenação) constituído por um conjunto de mensagens definidas no controlador do recurso, cuja a semântica deverá ser equivalente às seguidamente listadas.

Tabela 7.3-12

Interface com o componente hierarquicamente superior.

Entrada	Saída
inicia	
novo requisito	
abort	
	requisito aceite
	requisito negado
	cliente deixou de ser valido

Exemplos: processo controladores de sonares, ver 7.3.2.1 [Silva 99]

O interface para os componentes de níveis hierárquicos inferiores é constituído por mensagens definidas nos componentes objecto da interacção, e que são do tipo ***Estratégias de Controlo***

Tabela 7.3-13

Interface com o componente hierarquicamente inferior.

Entrada	Saída
	nova configuração

Modelo de transições temporizadas

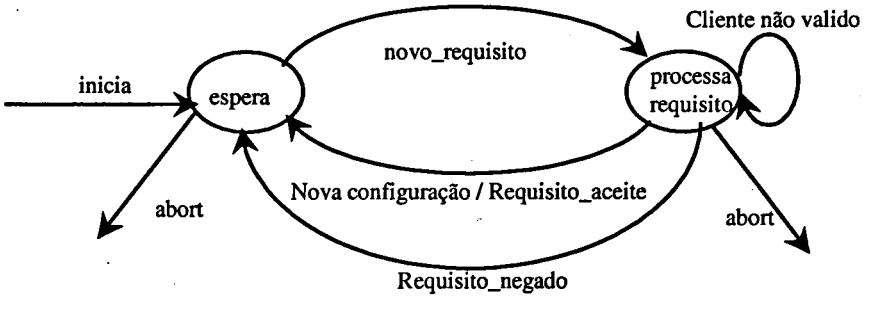
Variáveis

$act_nC1S1A \in \{\text{não_activa}, \text{espera}, \text{processa}\}$	variável que guarda o estado do sistema quanto à actividade do controlador do recurso nC1S1A
$act_r_nC1S1A \in \{\text{não_activa}, \text{activo}, \text{repouso}\}$	variável que guarda o estado do sistema quanto à actividade de recurso nC1S1A
parametrização	estrutura que armazena a parametrização
lista de clientes	estrutura que guarda a lista de clientes e respectivas prioridades

Transições

Figura 7.3-11

Diagrama de transições do controlador de recurso da classe nC1S1A.



Nome	Condição de activação	Transformação	Inferior	Superior
inicia	(*act_nC1S1A* = **não_activa**) ∧ (*act_r_nC1S1A* = **não_activa**)	[*act_nC1S1A* := *espera*; *act_r_nC1S1A* = *activo* *parametrização* = 0 *lista_clientes* = 0]	0	∞
novo_requisito.	(*act_nC1S1A* = **espera**) ∧ (*act_r_nC1S1A* = **activo**)	[*act_nC1S1A* := *requisito*; *act_r_nC1S1A* = *activo*]	0t	∞
req_aceite/ nov_confg	(*act_nC1S1A* = **requisito**) ∧ (*act_r_nC1S1A* = **activo**)	[*act_nC1S1A* := *espera*; *paraetrização* = *Nova*(*parametrização*) *lista_declientes_actializada*	0	tl
req_negado)	(*act_nC1S1A* = **requisito**) ∧ (*act_r_nC1S1A* = **ractivo**)	[*act_nC1S1A* := *espera*;]	tl	∞
cliente_n_vali	(*act_nC1S1A* = **requisito**) ∧ (*act_r_nC1S1A* = **activo**)	[*act_nC1S1A* := *requisito*; *extraicliente*(*lista_de_clientes*);]	0	∞
abortar	(*act_nC1S1A* = **espera**) ∧ (*act_r_nC1S1A* = **activo**)	[*act_nC1S1A* := *nao_activa*; *act_r_nC1S1A* = *repousa*]	0	∞

Tabela 7.3-14

Transições do controlador de recurso da classe nC1S1A.

tl– limite temporal para aceitar o novo requisito.

As actividades deste recurso e respectivo controlador caracterizam-se pela composição dos estados de dois autómatos

não_activo – este estado caracteriza a não actividade do recurso.

espera – o controlador aguarda um pedido de um componente hierarquicamente superior.

processa requisito – este estado indica que o controlador está a calcular a nova parametrização da actividade deste recurso. Caso a nova parametrização seja incompatível com um cliente já servido, se este tiver prioridade superior ao último cliente a nova parametrização é negada, caso seja o contrario, então o cliente com menos prioridade é notificado, que a parametrização não vai ser satisfeita.

Os acontecimentos observáveis do recurso são:

inicia – este acontecimento é consequência da activação do sistema realizada pelo utilizador, transferindo o controlador do recurso do sistema para o estado espera.

novo_req – este acontecimento é gerado por componentes hierarquicamente superiores. Significa que o "cliente" em questão necessita que o recurso adopte um comportamento compatível com as suas necessidades.

requesito_aceite – este acontecimento é gerado pelo controlador do recurso e, após a actividade que processa o novo pedido verificar que este é viável,

ocorre simultaneamente com a alteração da parametrização da actividade do recurso.

requesito_negado – este acontecimento é gerado pelo controlador do recurso, e após a actividade que processa o novo pedido verificar que este não é viável pelo facto do recurso estar a atender um cliente com prioridade superior, e com um requisito incompatível.

cliente_nao_valido– este acontecimento é gerado no controlador do recurso e, após a actividade que processa o novo pedido verificar que não pode continuar a atender um dado cliente.

abort – este acontecimento é gerado pela coordenação e o controlador do recurso deve promover o sincronismo com todos os módulos do recurso.

7.4 Estágio de coordenação

Seguidamente apresentamos de forma resumida e refinada o mecanismo para a coordenação de tarefas de um sistema autónomo apresentado originalmente por [Silva 99].

7.4.1 Modelização de Tarefas

Os controladores de recursos abstraem o nível funcional para o nível de coordenação. Pela especificidade de cada um, estes apresentam uma estrutura particular e, para a modularização da especificação do comportamento, é fundamental que os comportamentos admissíveis do sistema tenham uma estrutura comum. Neste sentido, vamos proceder a uma abstracção na hierarquia das actividades para que qualquer tarefa elementar executada no domínio de qualquer controlador seja vista na coordenação com a mesma estrutura. Neste processo, define-se também uma estrutura geral que uniformiza a representação de tarefa permitindo mais facilmente que as actividades possam evoluir de uma forma composta. Esta abstracção resulta no que chamamos tarefa elementar e é a mais pequena unidade "operativa" representativa de uma tarefa.

Como desejamos que as tarefas tenham uma composição hierárquica, a sua representação devia ser igual independentemente do posicionamento da hierarquia de tarefas. Neste contexto, o modelo apresentado para a definição

de uma tarefa elementar deverá se estendido para a definição de uma tarefa composta.

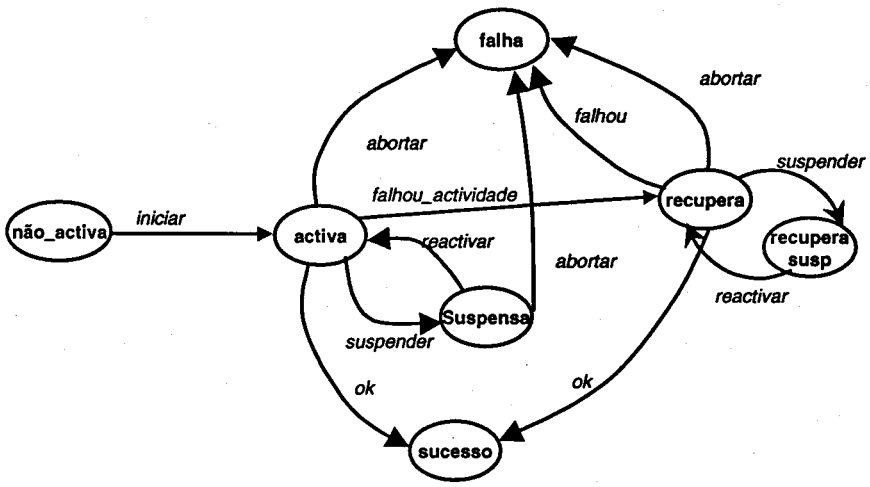


Figura 7.4-1

Diagrama de transições de uma tarefa.

Para se poder formular o problema básico do teoria de controlo supervisionado, precisamos de especificar o comportamento desejado como uma linguagem regular. Assim, uma tarefa é definida como gerador $G = (Q, \Sigma, \delta, q_0, F)$ em que:

Q é o conjunto de estados q e $Q = \left\{ \begin{matrix} \text{nao_activa, activa, suspensa, recupera,} \\ \text{recuperação_suspensa, falha, sucesso} \end{matrix} \right\}$

$F \subset Q$ é o subconjunto de estados finais, ou estados "marcados", tais que $F = Q_s \cup Q_n$ e $Q_s \cap Q_n = \varnothing$, em que Q_s é o conjunto de estados finais bem sucedidos ($Q_s = \{\text{sucesso}\}$), e Q_n é o conjunto de estados finais não sucedidos ($Q_n = \{\text{falha}\}$). Então $F = \{\text{falha, sucesso}\}$.

Σ é o conjunto de acontecimentos, tal que $\Sigma = \Sigma_c \cup \Sigma_{nc}$, e

$\Sigma_c = \{iniciar, abortar, suspender, reacti\,\text{var}\}$, $\Sigma_{nc} = \{falhou, falhou_actividade, ok\}$

$q_0 \in Q$ é o estado inicial, e $q_0 = não_activa$

δ é a função transição de estado, $\delta : \Sigma \times Q \rightarrow Q$, tal que:

$\delta(\text{nao_activa}, iniciar) = \text{activa}$,
$\delta(\text{activa}, ok) = \text{sucesso}$,
$\delta(\text{activa}, suspender) = \text{suspensa}$,
$\delta(\text{suspensa}, abortar) = \text{falha}$,
$\delta(\text{recupera}, ok) = \text{sucesso}$,
$\delta(\text{recupera_susp}, reacti\,\text{var}) = \text{recupera}$,
$\delta(\text{activa}, abortar) = \text{falha}$,
$\delta(\text{activa}, erro_actividade) = \text{recupera}$,
$\delta(\text{suspensa}, reacti\,\text{var}) = \text{activa}$,
$\delta(\text{recupera}, falhou) = \text{falha}$,
$\delta(\text{recupera}, suspender) = \text{recupera_susp}$
$\delta(\text{recupera}, abortar) = \text{falha}$

Os estados do autómato identificam as principais actividades de uma tarefa. O significado dos estados e dos acontecimentos de uma tarefa elementar são seguidamente descritos e facilmente se infere o significado no

caso das tarefas compostas através da observação da semântica dos operadores (serão introduzidos posteriormente no texto).

O estado:

não activa – o significado ocorre directamente da denotação

activa – fase em que o controlador do recurso desenvolve a actividade que implementa a tarefa.

suspenso – situação em que o controlador de recurso suspendeu a actividade por "ordem" da coordenação.

recupera – actividade da tarefa que ocorre após a falha da sua execução no respectivo controlador, e que tenta determinar e executar uma nova sequência de tarefas (com base no objectivo proposto e o estado corrente).

recupera_susp – situação em que estando a tarefa em recuperação o controlador de recurso suspendeu a actividade por "ordem" da coordenação.

sucesso – estado em que a tarefa já terminou e com sucesso, objectivo atingido.

falha – estado em que a tarefa já terminou e sem sucesso, objectivo não atingido.

Como se constatou, os acontecimentos de uma tarefa podem ser organizados em dois grupos, os controláveis e os não controláveis. A semântica dos controláveis, ocorre directamente da notação.

A semântica dos estados não controláveis, é:

erro_actividade – é síncrono com os acontecimentos que nos controladores de recurso assinalam a ocorrência de falha durante a execução

ok – é síncrono com os acontecimentos gerados pelo controladores de recursos que assinalam o sucesso da tarefa, caso o estado corrente seja o **recupera,** este acontecimento ocorre síncrono com o que assinala o sucesso gerado pela estrutura que "nasce" na actividade **recupera.**

falhou – é gerado pela actividade recupera após constatar a impossibilidade de recuperação da tarefa proposta.

7.4.2 Outros aspectos caracterizadores

A tarefa elementar é implementada e executada por um recurso.
A tarefa composição é implementada e executada na coordenação.

Além dos aspectos da execução, a caracterização de uma tarefa deve ter em conta também aspectos da síntese (informação para) e especificação, pelo que uma tarefa é globalmente caracterizada por:

(i) **Nome** da tarefa ou comando pela qual é conhecida, ou pode ser evocada.

(ii) **Parâmetros** e informação necessária à execução da tarefa.

(iii) **Autómato:** No caso de uma tarefa composta um autómato que descreve a composição das tarefas (descrição da tarefa composição).

(iv) **Objectivo:** é descrito por um estado ou família de estados que devem ser atingidos no fim da execução da tarefa. No caso de uma tarefa elementar o objectivo está definido no próprio recurso e é consequência da parametrização da tarefa. No caso de uma tarefa composição, o objectivo é dependente da composição dos objectivos das tarefas. Esta caracterização é importante para o planeamento de tarefas.

(v) **Pré-condições:** são requisitos ou famílias de estados admissíveis para se poder iniciar a execução de uma tarefa. No caso duma tarefa elementar as pré-condições já estão definidas no próprio recurso, sendo consequência da parametrização da tarefa. Nas tarefas composição as pré-condições, são dependentes das pré condições das tarefas hierarquicamente inferiores. Esta caracterização também é importante para o planeamento de tarefas.

(vi) **Condições de comportamento:** Definem estados que não devem ser observados (ou devem sempre). Tal como nas situações anteriores no caso das tarefas composição estes requisitos são a composição das requisitos das tarefas compostas.

Nas tarefas compostas poderá ser necessário apresentar, objectivos, pré-condições e condições de comportamento adicionais Assume particular interesse para o caso de desejarmos introduzir condições temporais ou energéticas extra. Contudo este facto não invalida a utilização de doutrinas de forma a obter o resultado esperado.

7.4.3 Modelização dos Operadores Composição de Tarefas

A escolha dos operadores apresentados é motivada pelo tipo de tarefas elementares já desenvolvidas ou a desenvolver. A descrição da tarefa como composição de estratégias elementares surge de forma natural como a especificação do comportamento desejado.

As tarefas argumentos da composição são descritas por dois autómatos de estados finitos $Tarefa_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ e $Tarefa_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$, segundo a definição do ponto 6.3. Para cada composição, modela-se uma instalação $I = (Q, \Sigma, \delta, q, F)$, que vai ser objecto de acções de controlo. O conjunto Γ dos controlos admissíveis para a instalação I é definido de acordo com Golaszewski [Golaszewski 91] (definição 3.1), como $\Gamma \subseteq \left\{\gamma : \gamma \in 2^{\Sigma}\right\}$.

Segundo o mesmo autor, para $\delta : \Sigma \times Q \to Q$, a função de transição de I, define-se uma nova função de transição $\delta_c : \Gamma \times \Sigma \times Q \to Q$ da seguinte forma:

$$\delta_c(\gamma, \sigma, \mathbf{q}) = \begin{cases} \delta(\sigma, \mathbf{q}), & se \sigma \in \gamma \\ indefinida & se \sigma \notin \gamma \end{cases}.$$

A esta corresponde uma nova instalação $I_c = (Q, \Gamma \times \Sigma, \delta_c, q, F)$.

Os mesmos autores [Golaszewski91] demonstraram a existência de um supervisor determinístico que permite manter a trajectória de I_c no conjunto das trajectórias desejadas. O supervisor para I_c é definido pelo mapa $f: L(I_c) \rightarrow \Gamma$, que para cada sequência de acontecimentos específica a acção de controlo resultante. Este mapa é então realizado por $\mathbf{S} := (S, \phi)$, em que $S = (X, \Sigma, \xi, x_0)$ é o autómato que funciona como observador da instalação e, simultaneamente, funciona como memória do supervisor. Para cada sequência de acontecimentos (ω) geradas por I_c a função de controlo $\phi: X \rightarrow 2^{\Sigma}$ selecciona os controlos admissíveis para cada estado $f(\omega) = \phi(\xi(\omega, x_0))$.

Estas entidades levam então à definição de um sistema em malha fechada $\mathbf{S}/I_c$, denominado processo de acontecimentos discreto supervisionado em que o comportamento do processo I_c é agora restringido pelo estado de S.

Ou seja,

$S/I_c := (X \times Q, \Sigma, \xi \times \delta_c, (x_0, q_0), F)$

sendo $\xi \times \delta_c : \Sigma \times X \times Q \rightarrow X \times Q$, definido por $(\sigma, \mathbf{x}, \mathbf{q}) \rightarrow (\xi(\sigma, x), \delta_c(\phi(x), \sigma, q))$.

O supervisor é consequência da semântica definida para o operador e pelo mecanismo de sincronização de acontecimentos.

Sucesso = Conjunção dos Sucessos

7.4.4 Paralelismo Forte

As transições entre os estados dos autómatos são assinalados por acontecimentos, quando esses acontecimentos são partilhados, estabelece-se uma ligação síncrona entre os dois autómatos exteriores. O nosso modelo de comunicação não é "broadcast". No entanto mais do que dois processos podem partilhar o mesmo acontecimento, ou seja, a sincronização pode ser realizada por dois ou mais autómatos.

Este operador denotado pelo símbolo "&&", tal como o nome indica deve ser utilizado para especificar que duas tarefas devem ser executadas simultaneamente e o sucesso do paralelo só ocorre se ambas forem bem sucedidas. A falha de uma delas implica abortar simultaneamente a outra tarefa. A resultante da operação "paralelismo forte" baseia-se no produto síncrono dos dois autómatos que a compõe. No entanto, o comportamento desejado para esta composição expressa uma linguagem mais restrita de que aquela que pode ser expressa pelo produto síncrono.

composição síncrona de duas tarefas em paralelo

Considera-se a instalação $I = (Q, \Sigma, \delta, \lfloor q_{1_0}, q_{2_0} \rfloor F_1 x F_2)$ controlável e definida da seguinte maneira:

Q é o conjunto de estados q, em que $Q \subseteq \{Q_1 x Q_2\}$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

Vamos acrescentar mais acontecimentos que expressam as acções da coordenação ou das tarefas hierarquicamente superiores sobre a tarefa composição.

δ é a função transição de estado, $\delta : \Sigma \times Q \to Q$, tal que:

$$\delta([q_1,q_2],\sigma) = [\delta_1(q_1,\sigma),q_2] \; para\,todo\,\sigma \in \Sigma_1,\; q_1 \in Q_1\, e\, q_2 \in Q_2$$

$$\delta([q_1,q_2],\sigma) = [q_1,\delta_2(q_2,\sigma)] \; para\,todo\,\sigma \in \Sigma_2,\; q_1 \in Q_1\, e\, q_2 \in Q_2$$

$$e\; q_1 \in \left\{ \begin{matrix} \mathbf{activa_1}, \mathbf{suspensa_1}, \\ \mathbf{recupera_1}, \mathbf{recupera_susp_1} \end{matrix} \right\} \subset Q_1,$$

$$\delta([q_1,q_2],abortar) = [\mathbf{falha_1},\mathbf{falha_2}] \quad e\; q_2 \in \left\{ \begin{matrix} \mathbf{activa_2}, \mathbf{suspensa_2}, \\ \mathbf{recupera_2}, \mathbf{recupera_susp_2} \end{matrix} \right\} \subset Q_2$$

$$\delta([\mathbf{nao_activa_1},\mathbf{nao_activa_2}],iniciar) = [\mathbf{activa_1},\mathbf{activa_2}]$$
$$\delta([\mathbf{activa_1},\mathbf{activa_2}],suspender) = [\mathbf{suspensa_2},\mathbf{suspensa_2}]$$
$$\delta([\mathbf{activa_1},\mathbf{recupera_2}],suspender) = [\mathbf{suspensa_1},\mathbf{recupera_susp_2}]$$
$$\delta([\mathbf{recupera_1},\mathbf{activa_2}],suspender) = [\mathbf{recupera_susp_1},\mathbf{suspensa_2}]$$
$$\delta([\mathbf{recupera_1},\mathbf{recupera_2}],suspender) = [\mathbf{recupera_susp_1},\mathbf{recupera_susp_2}]$$
$$\delta([\mathbf{suspensa_1},\mathbf{suspensa_2}],reacti\,\text{var}) = [\mathbf{activa_1},\mathbf{activa_2}]$$
$$\delta([\mathbf{suspensa_1},\mathbf{recupera_susp_2}],reacti\,\text{var}) = [\mathbf{activa_1},\mathbf{recupera_2}]$$
$$\delta([\mathbf{recupera_susp_1},\mathbf{suspensa_2}],reacti\,\text{var}) = [\mathbf{recupera_1},\mathbf{activa_2}]$$
$$\delta([\mathbf{recupera_susp_1},q_2],reacti\,\text{var}) = [\mathbf{recupera_1},q_2]\; \forall q_2 \in \{\mathbf{sucesso2},\mathbf{falha2}\}$$
$$\delta([\mathbf{recupera_1},q_2],suspender) = [\mathbf{recupera_susp_1},q_2]\; \forall q_2 \in \{\mathbf{sucesso2},\mathbf{falha2}\}$$
$$\delta([\mathbf{activa_1},q_2],\text{sup}ender) = [\mathbf{suspensa_1},q_2]\; \forall q_2 \in \{\mathbf{sucesso2},\mathbf{falha2}\}$$
$$\delta([\mathbf{suspensa_1},q_2],reacti\,\text{var}) = [\mathbf{activa_1},q_2]\; \forall q_2 \in \{\mathbf{sucesso2},\mathbf{falha2}\}$$
$$\delta([q_1,\mathbf{recupera_susp_2}],reacti\,\text{var}) = [q_1,\mathbf{recupera_2}]\; \forall q_1 \in \{\mathbf{sucesso1},\mathbf{falha1}\}$$
$$\delta([q_1,\mathbf{recupera_2}],suspender) = [q_1,\mathbf{recupera_susp_2}]\; \forall q_1 \in \{\mathbf{sucesso1},\mathbf{falha1}\}$$

supervisor paralelismo forte

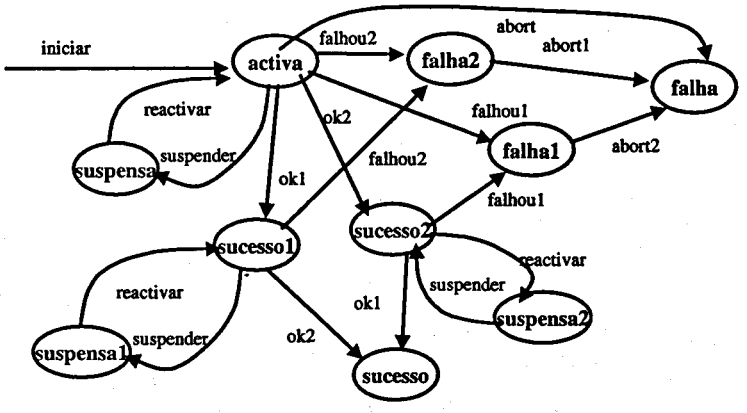
Esta instalação (composição paralela das duas tarefas) vai ser objecto de acções de controlo $\Gamma \subseteq \{\gamma : \gamma \in 2^{\Sigma}\}$, pelo que a instalação controlada é então $I_c = (Q, \Gamma \times \Sigma, \delta_c, \lfloor q_{1_0}, q_{2_0} \rfloor, F_1 x F_2)$

O supervisor para I_c será então definido como o mapa $f : L(I_c) \to \Gamma$. Sendo este realizado por $\mathbf{S} := (S,\phi)$, em que $S = (X,\Sigma,\xi,x_0)$ é o observador da instalação paralela, e $\phi : X \to 2^{\Sigma}$ a função de controlo, para cada sequência de acontecimentos ω geradas por I_c,

Figura 7.4-2

Diagrama do supervisor do operador "&&".

Notar que a transição abort *não esta representada nos estados* ***sucesso1*** *e* ***sucesso2*** *simplesmente para facilidade de leitura do diagrama.*



O observador $S=(X,\Sigma,\xi,x_0)$, para o operador "&&", é definido como se segue.

X é o conjunto de estados x, em que

$$X=\left\{\begin{matrix} nao_activo, activa, sucesso1, sucesso2, falhou1, falhou2, \\ suspensaA, suspensaS1, suspensaS2, sucesso, falha \end{matrix}\right\}$$

$\Sigma=\Sigma_1\cup\Sigma_2\cup\{iniciar, abortar, suspender, reacti\,\text{var}\}\cup\{\varepsilon\}$, notar que $\Sigma_1\cap\Sigma_2=\varnothing$

sendo que $\Sigma_{nc}=\Sigma_{1_{nc}}\cup\Sigma_{2_{nc}}\cup\{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c=\Sigma_{1_c}\cup\Sigma_{2_c}\cup\{iniciar\}$

ξ é a função transição de estado, $\xi:\Sigma\times X\rightarrow X$, tal que:

ξ(**nao_activa**,*inicio*) = **activa**	ξ(**activa**,*ok*1) = **sucesso1**
ξ(**activa**,*ok*2) = **sucesso2**	ξ(**activa**, *falhou*1) = **falha1**
ξ(**activa**, *falhou*2) = **falha2**	ξ(**activa**,*suspender*) = **suspensaA**
ξ(**suspensoA**, *reacti* var) = **activa**	ξ(**sucesso1**, *suspender*) = **suspensaS1**
ξ(**sucesso1**,*ok*2) = **sucesso**	ξ(**sucesso1**, *falha*2) = **falha**
ξ(**sucesso2**,*suspender*) = **suspensaS2**	ξ(**sucesso2**,*ok*1) = **sucesso**
ξ(**sucesso2**, *falha*1) = **falha**	ξ(**falha1**,*abortou*2) = **falha**
ξ(**falha2**,*abortou*1) = **falha**	ξ(**suspensoS1**, *reacti* var) = **sucesso1**
ξ(**suspensoS2**, *reacti* var) = **sucesso2**	ξ(**x**,*abortou*) = **falha** $\forall$**x** $\in$ {**activa,suce1,suce2**}

A função $\phi:X\rightarrow\Gamma$ (realimentação) do nosso supervisor que define o operador "&&" será então:

ϕ(**nao_activa**) = $\{inicia\}$	ϕ(**activa**) = Σ_{nc}
ϕ(**sucesso1** = Σ_{nc}	ϕ(**sucesso2**) = Σ_{nc}
ϕ(**falha1**) = $\{abort2\}$	ϕ(**falha2**) = $\{abort1\}$
ϕ(**falha**) = $\{\varepsilon\}$	ϕ(**sucesso**) = $\{\varepsilon\}$
ϕ(**suspensoA**) = Σ_{nc}	ϕ(**suspensoS1**) = Σ_{nc}
ϕ(**suspensoS2**) = Σ_{nc}	

Estas entidades levam então à definição de um sistema em malha fechada S/I_c (definido como em 6.5).

7.4.5 Paralelismo Fraco

Este operador, denotado pelo símbolo "||", especifica a composição de duas tarefas evoluir paralelamente, a que corresponde o sucesso se pelo menos uma delas suceder. Só a falha das duas tarefas pode determinar o insucesso da tarefa composição. A tarefa "paralelismo fraco" tal como a tarefa "paralelismo forte", assenta no produto síncrono de dois autómatos.

Insucesso = Conjunção dos Insucessos.

composição síncrona de duas tarefas em paralelo

Ver definição descrita no operador "paralelismo forte".

supervisor paralelismo fraco

Tal como no caso anterior esta instalação (composição paralela das duas tarefas) vai ser objecto de acções de controlo $\Gamma \subseteq \{\gamma : \gamma \in 2^{\Sigma}\}$, pelo que a instalação controlada é então $I_c = (Q, \Gamma \times \Sigma, \delta_c, \lfloor q_{1_0}, q_{2_0} \rfloor, F_1 x F_2)$

O supervisor para I_c será então definido como o mapa $f : L(I_c) \to \Gamma$. Sendo este realizado por $S := (S, \phi)$, em que $S = (X, \Sigma, \xi, x_0)$ é o observador da instalação paralela, e $\phi : X \to 2^{\Sigma}$ a função de controlo, para cada sequência de acontecimentos ω geradas por I_c,

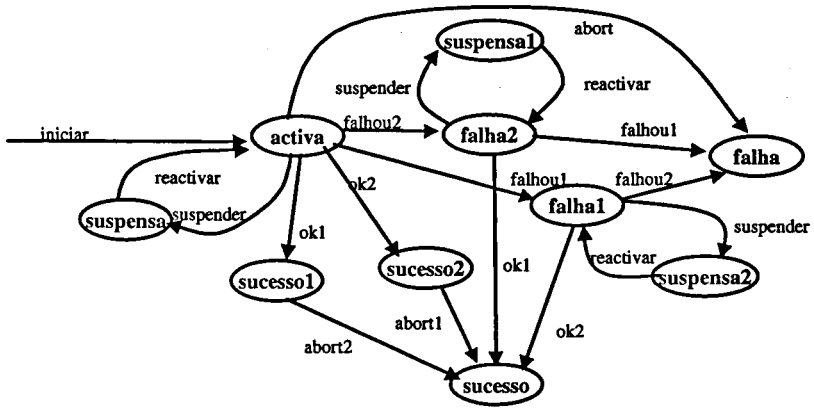


Figura 7.4-3

Diagrama do supervisor do operador "||".

Notar que a transição abort *não esta representada nos estados* ***falha1*** *e* ***falha2*** *simplesmente para facilidade de leitura do diagrama.*

O observador $S = (X, \Sigma, \xi, x_0)$, para o operador "||", é definido como se segue.

X é o conjunto de estados x, em que

$$X = \begin{Bmatrix} nao_activo, activa, sucesso1, sucesso2, falhou1, falhou2, \\ suspensaA, suspensaF1, suspensaF2, sucesso, falha \end{Bmatrix}$$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

ξ é a função transição de estado, $\xi : \Sigma \times X \rightarrow X$, tal que:

$\xi(\text{nao_activa}, inicio) = \text{activa}$ $\quad$ $\xi(\text{activa}, ok1) = \text{sucesso1}$
$\xi(\text{activa}, ok2) = \text{sucesso2}$ $\quad$ $\xi(\text{activa}, falhou1) = \text{falha1}$
$\xi(\text{activa}, falhou2) = \text{falha2}$ $\quad$ $\xi(\text{activa}, suspender) = \text{suspensaA}$
$\xi(\text{suspensoA}, reacti\,\text{var}) = \text{activa}$ $\quad$ $\xi(\text{sucesso1}, abortar2) = \text{sucesso}$
$\xi(\text{falha1}, suspender) = \text{suspensaF1}$ $\quad$ $\xi(\text{falha1}, falhou2) = \text{falha}$
$\xi(\text{falha2}, suspeender) = \text{suspensaF2}$ $\quad$ $\xi(\text{sucesso2}, abortal) = \text{sucesso}$
$\xi(\text{falha2}, falhou1) = \text{falha}$ $\quad$ $\xi(\text{falha1}, ok2) = \text{sucesso}$
$\xi(\text{falha2}, ok1) = \text{sucesso}$ $\quad$ $\xi(\text{suspensoF1}, reacti\,\text{var}) = \text{falha1}$
$\xi(\text{suspensoF2}, reacti\,\text{var}) = \text{falha2}$ $\quad$ $\xi(\text{x}, abortou) = \text{falha}\ \forall \text{x} \in \{\text{activa}, \text{falha1}, \text{falha2}\}$

A função $\phi : X \rightarrow \Gamma$ (realimentação) do nosso supervisor que define o operador "|" será então:

$\phi(\text{nao_activa}) = \{inicia\}$ $\quad$ $\phi(\text{activa}) = \Sigma_{nc}$
$\phi(\text{sucesso1}) = \{abort2\}$ $\quad$ $\phi(\text{sucesso2}) = \{abort1\}$
$\phi(\text{falha1}) = \Sigma_{nc}$ $\quad$ $\phi(\text{falha2}) = \Sigma_{nc}$
$\phi(\text{falha}) = \{\varepsilon\}$ $\quad$ $\phi(\text{sucesso}) = \{\varepsilon\}$
$\phi(\text{suspensoA}) = \Sigma_{nc}$ $\quad$ $\phi(\text{suspensoF1}) = \Sigma_{nc}$
$\phi(\text{suspensoF2}) = \Sigma_{nc}$

Estas entidades levam então à definição de um sistema em malha fechada S/I_c (definido como em 6.5).

7.4.6 Sequenciamento Forte

Este operador denotado pelo símbolo "&", indica que duas tarefas devem ser executadas uma após a outra. O sucesso desta sequência só ocorre se as duas forem bem sucedidas. A falha da tarefa que evolui inicialmente implica a falha da composição das tarefas sem haver possibilidade de iniciar a segunda. Caso a primeira tarefa evolua para um estado de sucesso então dá-se início à segunda tarefa e só o sucesso desta (após o sucesso da primeira) garante o sucesso da composição.

composição de duas tarefas em sequência

Considere-se a instalação $I = (Q, \Sigma, \delta, \text{nao_activa}, \{\text{falha}, \text{sucesso}\})$ controlável definida da seguinte maneira:

Q é o conjunto de estados q , em que $Q \subseteq Q_1 \cup Q_2 \cup \{\text{nao_activa}, \text{falha}, \text{sucesso}\}$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

δ é a função transição de estado, $\delta : \Sigma \times Q \rightarrow Q$, tal que:

Vamos acrescentar mais acontecimentos, que expressam as acções da coordenação ou das tarefas hierarquicamente superiores sobre a tarefa composição.

$\delta(q,\sigma) = \delta_1(q,\sigma)\, para\, todo\, \sigma \in \Sigma_1, e\, q \in Q_1$

$\delta(q,\sigma) = \delta_2(q,\sigma)\, para\, todo\, \sigma \in \Sigma_2, e\, q \in Q_2$

$\delta(\text{sucesso}_1, \varepsilon) = \text{nao_activa}_2$

$\delta(\text{falha}_1, \varepsilon) = \text{falha}$

$$\delta(q, abortar) = \text{falha}\ \forall q \in \begin{Bmatrix} \text{activa}_1, \text{suspensa}_1, \text{recupera}_1, \text{recupera_susp}_1 \\ \text{activa}_2, \text{suspensa}_2, \text{recupera}_2, \text{recupera_susp}_2 \end{Bmatrix}$$

$\delta(\text{nao_activa}, iniciar) = \text{nao_activa}_1,$

$\delta(\text{activa}_i, suspender) = \text{suspensa}_i, i = 1,2$

$\delta(\text{recupera}_i, reactiva) = \text{activa}_i, i = 1,2$

$\delta(\text{recupera}_i, suspender) = \text{recupera_susp}_i, i = 1,2$

$\delta(\text{recupera_susp}_i, reactiva) = \text{recupera}_i, i = 1,2$

supervisor sequenciamento forte

Tal como no caso anterior esta instalação (sequência de duas tarefas) vai ser objecto de acções de controlo $\Gamma \subseteq \{\gamma : \gamma \in 2^{\Sigma}\}$, pelo que a instalação controlada é então $I_c = (Q, \Gamma \times \Sigma, \delta_c, \text{nao_activa}, \{\text{falha}, \text{sucesso}\})$.

O supervisor para I_c será então definido como o mapa $f : L(I_c) \to \Gamma$. Sendo este realizado por $S := (S, \phi)$, em que $S = (X, \Sigma, \xi, x_0)$ é o observador da instalação série, e $\phi : X \to 2^{\Sigma}$ a função de controlo, para cada sequência de acontecimentos ω geradas por I_c,

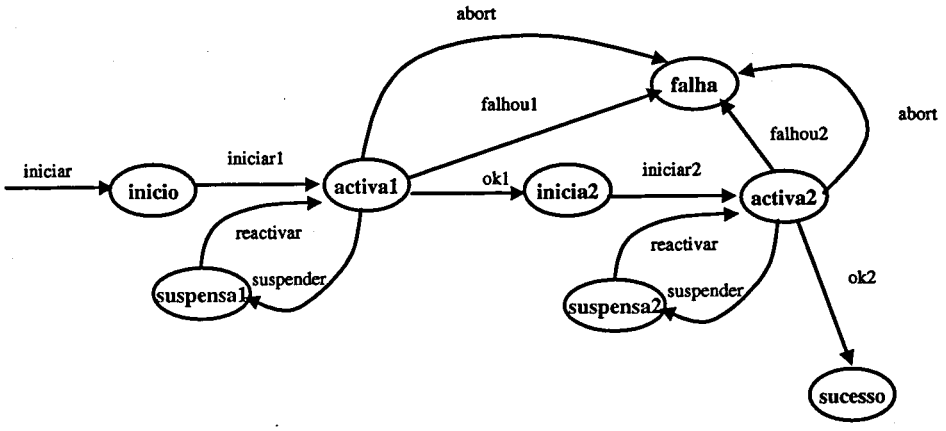


Figura 7.4-4

Diagrama do supervisor do operador "&".

O observador $S = (X, \Sigma, \xi, x_0)$, para o operador "&", é definido como se segue.

X é o conjunto de estados x, em que

$$X = \begin{Bmatrix} nao_activo, inicio, activa1, inicia2, activa2, \\ suspensa1, suspensa2, sucesso, falha \end{Bmatrix}$$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

ξ é a função transição de estado, $\xi : \Sigma \times X \rightarrow X$, tal que:

$\xi(\textbf{nao_activa}, \textit{iniciar}) = \textbf{inicio}$ $\quad$ $\xi(\textbf{activa1}, \textit{ok1}) = \textbf{inicio2}$

$\xi(\textbf{inicio}, \textit{iniciar1}) = \textbf{activa1}$ $\quad$ $\xi(\textbf{activa1}, \textit{falhou1}) = \textbf{falha}$

$\xi(\textbf{activa2}, \textit{falhou2}) = \textbf{falha2}$ $\quad$ $\xi(\textbf{activa1}, \textit{suspender}) = \textbf{suspensa1}$

$\xi(\textbf{suspenso1}, \textit{reacti}\,\text{var}) = \textbf{activa1}$ $\quad$ $\xi(\textbf{inicia2}, \textit{iniciar}) = \textbf{activa2}$

$\xi(\textbf{activa2}, \textit{ok2}) = \textbf{sucesso}$ $\quad$ $\xi(\textbf{suspensa2}, \textit{reacti}\,\text{var}) = \textbf{activa2}$

$\xi(\textbf{activa2}, \textit{suspeender}) = \textbf{suspensa2}$ $\quad$ $\xi(\textbf{x}, \textit{abortou}) = \textbf{falha}\ \forall \textbf{x} \in \{\textbf{activa1}, \textbf{activa2}\}$

A função $\phi : X \rightarrow \Gamma$ (realimentação) do nosso supervisor que define o operador "&" será então:

$\phi(\textbf{nao_activa}) = \{\textit{inicia}\}$ $\quad$ $\phi(\textbf{activa1}) = \Sigma_{nc}$

$\phi(\textbf{inicio}) = \{\textit{inicia1}\}$ $\quad$ $\phi(\textbf{activa2}) = \Sigma_{nc}$

$\phi(\textbf{inicia2}) = \{\textit{inicia2}\}$ $\quad$ $\phi(\textbf{suspenso1}) = \Sigma_{nc}$

$\phi(\textbf{falha}) = \{\varepsilon\}$ $\quad$ $\phi(\textbf{sucesso}) = \{\varepsilon\}$

$\phi(\textbf{suspenso2}) = \Sigma_{nc}$

Estas entidades levam então à definição de um sistema em malha fechada $\mathbf{S}/I_c$ (definido como em 6.5).

7.4.7 Sequenciamento Fraco

Este operador denotado pelo símbolo ";", indica que duas tarefas devem ser executadas uma após a outra. No entanto essa sequência não esta dependente do sucesso da primeira tarefa. A segunda tarefa deve ser executada após o fim da primeira, mesmo que esta tenha terminado sem sucesso. Para a composição das tarefas não falhar é necessário que a segunda tarefa chegue ao estado final **sucesso**.

composição de duas tarefas em sequência fraca

A instalação $I = (Q, \Sigma, \delta, \textbf{nao_activa}, F)$ controlável pode então ser definida da seguinte maneira:

Q é o conjunto de estados q , em que $Q \subseteq Q_1 \cup Q_2 \cup \{\textbf{nao_activa}, \textbf{falha}, \textbf{sucesso}\}$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{\textit{iniciar}, \textit{abortar}, \textit{suspender}, \textit{reacti}\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{\textit{abortar}, \textit{suspender}, \textit{reacti}\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{\textit{iniciar}\}$

δ é a função transição de estado, $\delta : \Sigma \times Q \rightarrow Q$, tal que:

$\delta(q,\sigma) = \delta_1(q,\sigma)\, para\, todo\, \sigma \in \Sigma_1, e\, q \in Q_1$

$\delta(q,\sigma) = \delta_2(q,\sigma)\, para\, todo\, \sigma \in \Sigma_2, e\, q \in Q_2$

$\delta(\mathbf{sucesso_1}, \varepsilon) = \mathbf{nao_activa_2}$

$\delta(\mathbf{falha_1}, \varepsilon) = \mathbf{nao_activa_2}$

$\delta(q, abortar) = \mathbf{falha}\ \forall q \in \begin{Bmatrix} \mathbf{activa_1}, \mathbf{suspensa_1}, \mathbf{recupera_1}, \mathbf{recupera_susp_1} \\ \mathbf{activa_2}, \mathbf{suspensa_2}, \mathbf{recupera_2}, \mathbf{recupera_susp_2} \end{Bmatrix}$

$\delta(\mathbf{nao_activa}, iniciar) = \mathbf{nao_activa_1}$,

$\delta(\mathbf{activa_i}, suspender) = \mathbf{suspensa_i}, i = 1,2$

$\delta(\mathbf{recupera_i}, reactiva) = \mathbf{activa_i}, i = 1,2$

$\delta(\mathbf{recupera_i}, suspender) = \mathbf{recupera_susp_i}, i = 1,2$

$\delta(\mathbf{recupera_susp_i}, reactiva) = \mathbf{recupera_i}, i = 1,2$

Notar que a função de transição apenas difere relativamente da função de transição do forte sequenciamento para o caso do estado primeira da tarefa atingir o estado falha, e neste operador a transição ocorre para dar início à segunda tarefa, não falhando a composição.

supervisor sequenciamento fraco

Tal como no caso anterior esta instalação (sequência de duas tarefas) vai ser objecto de acções de controlo $\Gamma \subseteq \{\gamma : \gamma \in 2^{\Sigma}\}$, pelo que a instalação controlada é então $I_c = (Q, \Gamma \times \Sigma, \delta_c, \mathbf{nao_activa}, \{\mathbf{falha}, \mathbf{sucesso}\})$.

O supervisor para I_c será então definido como o mapa $f : L(I_c) \rightarrow \Gamma$. Sendo este realizado por $\mathbf{S} := (S, \phi)$, em que $S = (X, \Sigma, \xi, x_0)$ é o observador da instalação série, e $\phi : X \rightarrow 2^{\Sigma}$ a função de controlo, para cada sequência de acontecimentos (ω) geradas por I_c,

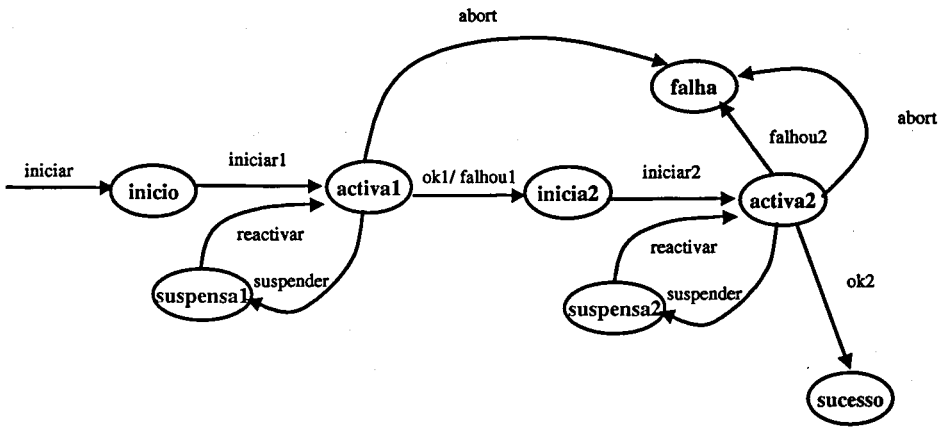


Figura 7.4-5

Diagrama do supervisor do operador ";".

O observador $S = (X, \Sigma, \xi, x_0)$, para o operador ";" é bastante idêntico ao apresentado anteriormente, e é definido como se segue.

X é o conjunto de estados x, em que

$$X = \begin{Bmatrix} nao_activo, inicio, activa1, inicia2, activa2, \\ suspensa1, suspensa2, sucesso, falha \end{Bmatrix}$$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_u = \Sigma_{1_u} \cup \Sigma_{2_u} \cup \{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

ξ é a função transição de estado, $\xi : \Sigma \times X \rightarrow X$, tal que:

ξ(**nao_activa**, *iniciar*) = **inicio**
ξ(**activa1**, *ok1*) = **inicio2**
ξ(**inicio**, *iniciar1*) = **activa1**
ξ(**activa1**, *falhou1*) = **inicio2**
ξ(**activa2**, *falhou2*) = **falha**
ξ(**activa1**, *suspender*) = **suspensa1**
ξ(**suspenso1**, *reacti* var) = **activa1**
ξ(**inicia2**, *iniciar*) = **activa2**
ξ(**activa2**, *ok2*) = **sucesso**
ξ(**suspensa2**, *reacti* var) = **activa2**
ξ(**activa2**, *suspender*) = **suspensa2**
ξ(**x**, *abortou*) = **falha** $\forall$**x** $\in$ {**activa1**, **activa2**}

A função $\phi : X \rightarrow \Gamma$ (realimentação) do nosso supervisor que define o operador ";", é igual ao definido para a tarefa anterior, será então:

ϕ(**nao_activa**) = {*inicia*}
ϕ(**activa1**) = Σ_{nc}
ϕ(**inicio**) = {*inicia1*}
ϕ(**activa2**) = Σ_{nc}
ϕ(**inicia2**) = {*inicia2*}
ϕ(**suspenso1**) = Σ_{nc}
ϕ(**falha**) = {ε}
ϕ(**sucesso**) = {ε}
ϕ(**suspenso2**) = Σ_{nc}

Estas entidades levam então à definição de um sistema em malha fechada $\mathbf{S}/I_c$ (definido como em 6.5).

Notar que a diferença entre este operador e o sequenciamento forte reside no observador e na modelização da composição.

7.4.8 Sequenciamento condicional

O operador "sequenciamento condicional", funciona na lógica inversa do "forte sequenciamento", ou seja, a segunda tarefa só se inicia se a primeira tarefa não obtiver sucesso. Este operador denotado pelo símbolo "**?**", é idêntico ao "forte sequenciamento", em que se atribui sinónimos inversos aos estados falha e sucesso da primeira tarefa. A segunda tarefa só deve ser executada após o fim da primeira, caso esta falhe. Se o estado da primeira tarefa for sucesso então a tarefa composição atingiu o estado de sucesso, mas caso a primeira falhe é necessário que a segunda tarefa chegue ao estado final **sucesso**.

composição de duas tarefas em sequência condicional

Considere a instalação $I = (Q, \Sigma, \delta, \textbf{nao_activa}, F)$ controlada definida da seguinte maneira:

Q é o conjunto de estados q, em que $Q \subseteq Q_1 \cup Q_2 \cup \{\textbf{nao_activa}, \textbf{falha}, \textbf{sucesso}\}$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{abortar, suspender, reacti\,var\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

δ é a função transição de estado, $\delta : \Sigma \times Q \rightarrow Q$, tal que:

Notar a ligeira alteração na função de transição relativamente aos casos anteriores.

$$\delta(q,\sigma) = \delta_1(q,\sigma)\, para\, todo\, \sigma \in \Sigma_1, e\, q \in Q_1$$
$$\delta(q,\sigma) = \delta_2(q,\sigma)\, para\, todo\, \sigma \in \Sigma_2, e\, q \in Q_2$$
$$\delta(\mathbf{sucesso}_1, \varepsilon) = \mathbf{sucesso}$$
$$\delta(\mathbf{sucesso}_2, \varepsilon) = \mathbf{sucesso}$$
$$\delta(\mathbf{falha}_1, \varepsilon) = \mathbf{nao_activa}_2$$
$$\delta(q, abortar) = \mathbf{falha}\ \forall q \in \begin{Bmatrix} \mathbf{activa}_1, \mathbf{suspensa}_1, \mathbf{recupera}_1, \mathbf{recupera_susp}_1 \\ \mathbf{activa}_2, \mathbf{suspensa}_2, \mathbf{recupera}_2, \mathbf{recupera_susp}_2, \end{Bmatrix}$$
$$\delta(\mathbf{nao_activa}, iniciar) = \mathbf{nao_activa}_1,$$
$$\delta(\mathbf{activa}_i, suspender) = \mathbf{suspensa}_i, i = 1,2$$
$$\delta(\mathbf{recupera}_i, reactiva) = \mathbf{activa}_i, i = 1,2$$
$$\delta(\mathbf{recupera}_i, suspender) = \mathbf{recupera_susp}_i, i = 1,2$$
$$\delta(\mathbf{recupera_susp}_i, reactiva) = \mathbf{recupera}_i, i = 1,2$$

supervisor sequenciamento condicional

Tal como no caso anterior esta instalação (sequência de duas tarefas) vai ser objecto de acções de controlo $\Gamma \subseteq \{\gamma : \gamma \in 2^{\Sigma}\}$, pelo que a instalação controlada é então $I_c = (Q, \Gamma \times \Sigma, \delta_c, \mathbf{nao_activa}, \{\mathbf{falha}, \mathbf{sucesso}\})$.

O supervisor para I_c será então definido como o mapa $f : L(I_c) \rightarrow \Gamma$. Sendo este realizado por $\mathbf{S} := (S, \phi)$, em que $S = (X, \Sigma, \xi, x_0)$ é o observador da instalação série, e $\phi : X \rightarrow 2^{\Sigma}$ a função de controlo, para cada sequência de acontecimentos ω geradas por I_c,

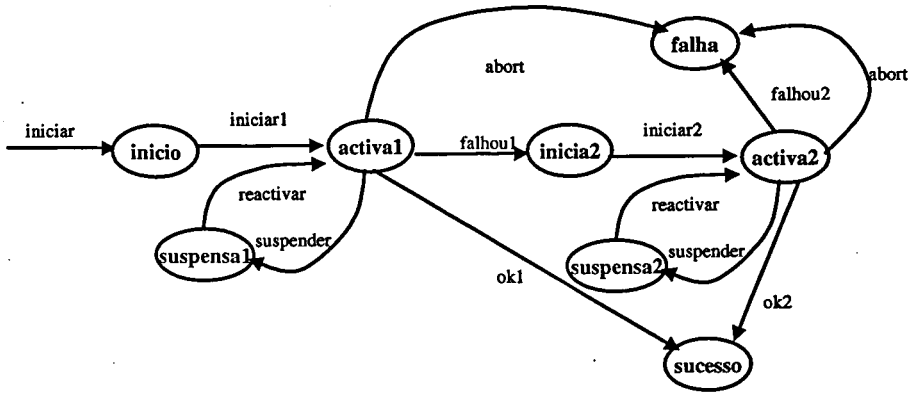


Figura 7.4-6

Diagrama do supervisor do operador ";".

Então o observador $S = (X, \Sigma, \xi, x_0)$, para o operador **"?"** é idêntico ao apresentado anteriormente e, definido como se segue.

X é o conjunto de estados x, em que

$$X = \begin{Bmatrix} nao_activo, inicio, activa1, inicia2, activa2, \\ suspensa1, suspensa2, sucesso, falha \end{Bmatrix}$$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

ξ é a função transição de estado, $\xi : \Sigma \times X \rightarrow X$, tal que:

ξ(**nao_activa**, *iniciar*) = **inicio**	ξ(**activa1**, *ok*1) = **sucesso**
ξ(**inicio**, *iniciar*1) = **activa1**	ξ(**activa1**, *falhou*1) = **inicio2**
ξ(**activa2**, *falhou*2) = **falha**	ξ(**activa1**, *suspender*) = **suspensa1**
ξ(**suspenso1**, *reacti* var) = **activa1**	ξ(**inicia2**, *iniciar*) = **activa2**
ξ(**activa2**, *ok*2) = **sucesso**	ξ(**suspensa2**, *reacti* var) = **activa2**
ξ(**activa2**, *suspender*) = **suspensa2**	ξ(**x**, *abortou*) = **falha** $\forall$**x** $\in$ {**activa1**,**activa2**}

A função $\phi : X \rightarrow \Gamma$ (realimentação) do nosso supervisor que define o operador ";", é igual ao definido para a tarefa anterior, será então:

ϕ(**nao_activa**) = $\{inicia\}$	ϕ(**activa1**) = Σ_{nc}
ϕ(**inicio**) = $\{inicial\}$	ϕ(**activa2**) = Σ_{nc}
ϕ(**inicia2**) = $\{inicia2\}$	ϕ(**suspenso1**) = Σ_{nc}
ϕ(**falha**) = $\{\varepsilon\}$	ϕ(**sucesso**) = $\{\varepsilon\}$
ϕ(**suspenso2**) = Σ_{nc}	

Estas entidades levam então à definição de um sistema em malha fechada S/I_c (definido como em 6.5).

Notar mais uma vez que a entre este operador e os operadores sequenciais reside no observador e na modelização da composição.

7.4.9 Composição cíclica condicional

O operador "composição cíclica condicional" denotado "!", foi definido no sentido de permitir implementar ciclos de funcionamento condicionais. Este operador contraria o princípio de funcionamento adoptado, nos operadores anteriores. Enquanto os anteriores não tinham caminho para uma trajectória em ciclo, este operador faz a instalação entrar ciclo caso a sequência das duas a tarefas falhe. O objectivo desta tarefa é garantir que, caso uma tarefa suceda a seguinte obrigatoriamente tem de suceder, e na situação da segunda falhar então temos de voltar a executar a primeira tarefa. No entanto, caso a primeira tarefa falhe então a composição **falha**.

composição cíclica condicional de duas tarefas

A instalação $I = (Q, \Sigma, \delta, \text{nao_activa}, F)$ controlável pode então ser definida da seguinte maneira:

Q é o conjunto de estados q , em que $Q \subseteq Q_1 \cup Q_2 \cup \{\text{nao_activa}, \text{falha}, \text{sucesso}\}$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

δ é a função transição de estado, $\delta : \Sigma \times Q \to Q$, tal que:

Notar as semelhanças entre as composições dos operadores sequenciais.

$\delta(\mathbf{q},\sigma) = \delta_1(\mathbf{q},\sigma)\, para\, todo\, \sigma \in \Sigma_1, e\, \mathbf{q} \in Q_1$

$\delta(\mathbf{q},\sigma) = \delta_2(\mathbf{q},\sigma)\, para\, todo\, \sigma \in \Sigma_2, e\, \mathbf{q} \in Q_2$

$\delta(\mathbf{sucesso_1}, \varepsilon) = \mathbf{inicia_2}$

$\delta(\mathbf{sucesso_2}, \varepsilon) = \mathbf{sucesso}$

$\delta(\mathbf{falha_1}, \varepsilon) = \mathbf{falha}$

$\delta(\mathbf{falha_2}, \varepsilon) = \mathbf{naso_activa_1}$

$\delta(\mathbf{q}, abortar) = \mathbf{falha}\ \forall \mathbf{q} \in \begin{Bmatrix} \mathbf{activa_1}, \mathbf{suspensa_1}, \mathbf{recupera_1}, \mathbf{recupera_susp_1}, \\ \mathbf{activa_2}, \mathbf{suspensa_2}, \mathbf{recupera_2}, \mathbf{recupera_susp_2}, \end{Bmatrix}$

$\delta(\mathbf{nao_activa}, iniciar) = \mathbf{nao_activa_1}$,

$\delta(\mathbf{activa_i}, suspender) = \mathbf{suspensa_i}, i = 1,2$

$\delta(\mathbf{recupera_i}, reactiva) = \mathbf{activa_i}, i = 1,2$

$\delta(\mathbf{recupera_i}, suspender) = \mathbf{recupera_susp_i}, i = 1,2$

$\delta(\mathbf{recupera_susp_i}, reactiva) = \mathbf{recupera_i}, i = 1,2$

supervisor composição cíclica

Tal como no caso anterior esta instalação vai ser objecto de acções de controlo $\Gamma \subseteq \{\gamma : \gamma \in 2^{\Sigma}\}$, pelo que a instalação controlável é então $I_c = (Q, \Gamma \times \Sigma, \delta_c, \mathbf{nao_activa}, \{\mathbf{falha}, \mathbf{sucesso}\})$.

O supervisor para I_c será então definido como o mapa $f : L(I_c) \to \Gamma$. Sendo este realizado por $\mathbf{S} := (S, \phi)$, em que $S = (X, \Sigma, \xi, x_0)$ é o observador da instalação série, e $\phi : X \to 2^{\Sigma}$ a função de controlo, para cada sequência de acontecimentos ω geradas por I_c,

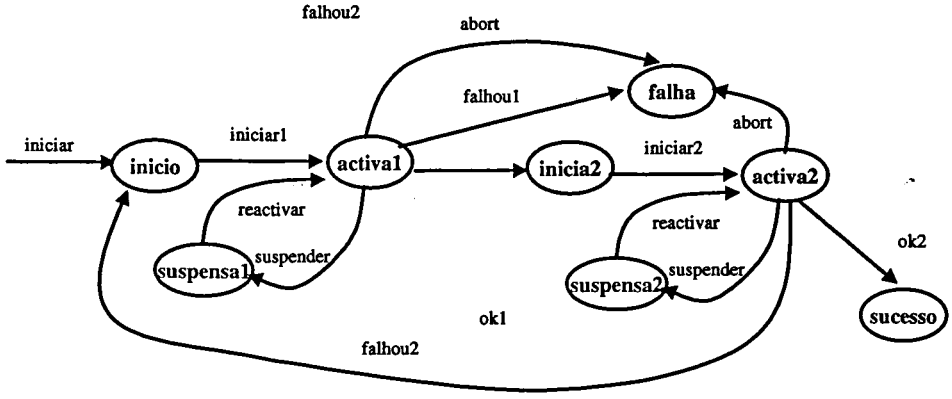


Figura 7.4-7

Diagrama do supervisor do operador "!".

Então o observador $S = (X, \Sigma, \xi, x_0)$, para o operador "!" é bastante idêntico ao apresentado anteriormente e, é definido como se segue.

X é o conjunto de estados x, em que

$$X = \begin{Bmatrix} nao_activo, inicio, activa1, inicia2, activa2, \\ suspensa1, suspensa2, sucesso, falha \end{Bmatrix}$$

$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{iniciar, abortar, suspender, reacti\,\text{var}\} \cup \{\varepsilon\}$, notar que $\Sigma_1 \cap \Sigma_2 = \varnothing$

sendo que $\Sigma_{nc} = \Sigma_{1_{nc}} \cup \Sigma_{2_{nc}} \cup \{abortar, suspender, reacti\,\text{var}\}$, $\Sigma_c = \Sigma_{1_c} \cup \Sigma_{2_c} \cup \{iniciar\}$

ξ é a função transição de estado, $\xi : \Sigma \times X \rightarrow X$, tal que:

ξ(**nao_activa**, *iniciar*) = **inicio**	ξ(**activa1**, *ok*1) = **sucesso**
ξ(**inicio**, *iniciar*1) = **activa1**	ξ(**activa1**, *falhou*1) = **falha**
ξ(**activa2**, *falhou*2) = **inicio**	ξ(**activa1**, *suspender*) = **suspensa1**
ξ(**suspenso1**, *reacti* var) = **activa1**	ξ(**inicia2**, *iniciar*2) = **activa2**
ξ(**activa2**, *ok*2) = **sucesso**	ξ(**suspensa2**, *reacti* var) = **activa2**
ξ(**activa2**, *suspender*) = **suspensa2**	ξ(**x**, *abortou*) = **falha** $\forall$**x** $\in$ {**activa1**,**activa2**}

A função $\phi : X \rightarrow \Gamma$ (realimentação) do nosso supervisor que define o operador ";", é igual ao definido para a tarefa anterior, será então:

ϕ(**nao_activa**) = {*iniciar*}	ϕ(**activa1**) = Σ_u
ϕ(**inicio**) = {*inicia*1}	ϕ(**activa2**) = Σ_u
ϕ(**inicia2**) = {*inicia*2}	ϕ(**suspenso1**) = Σ_u
ϕ(**falha**) = {ε}	ϕ(**sucesso**) = {ε}
ϕ(**suspenso2**) = Σ_u	

Estas entidades levam então à definição de um sistema em malha fechada S/I_c (definido como em 6.5).

Notar novamente que a diferença nos operadores sequências se dá ao nível do observador.

Esta página foi intencionalmente deixada em branco

PARTE IV

Ferramentas para a Síntese de Problemas de Controlo

Esta página foi intencionalmente deixada em branco

8 FORMULAÇÃO DOS PROBLEMAS DE CONTROLO

8.1 Introdução

O século vinte testemunhou desenvolvimentos notáveis da teoria de controlo, tendo sido consolidado uma gama de conceitos e resultados quer para sistemas lineares quer não lineares. Com a cada vez maior disponibilidade de capacidade computacional e o seu papel cada vez mais generalizado na concepção de sistemas, tornou premente os desenvolvimentos correntes dos sistemas híbridos. A evolução deste tipo de sistemas processa-se através de uma íntima ligação entre dois tipos de componentes: uma em que o estado evolui no tempo e outra em que a transição de estado se deve à ocorrência de acontecimentos discretos. Para a emergência de uma extensão da teoria de controlo para sistemas híbridos tem-se considerado um quadro de desenvolvimento que combina conceitos e métodos quer da teoria do controlo para sistemas dinâmicos convencionais quer das ciências de computação. Neste capítulo, serão descritos de forma sucinta o núcleo de conceitos fundamentais relevantes para o controlo de sistemas dinâmicos.

Adopta-se a seguinte notação: um sistema dinâmico evolui ao longo do tempo ($t \in T$) produzindo sinais de saída, $y(t) \in Y$, em resposta aos sinais de entrada, $u(t) \in U$, e dependendo do estado $x(t) \in X$. A natureza destas funções e a sua relação com o tempo determinam as várias possibilidades no que respeita à classificação dos diversos tipos de modelos de sistemas de controlo que é possível utilizar.

A evolução da variável de estado num dado intervalo de tempo, $x: T \rightarrow X$, designa-se normalmente de trajectória, "história" ou "trace" do sistema e desempenha um papel fundamental como objecto da análise destes sistemas. A evolução da variável de estado em função da entrada pode ser descrita pela equação transição de estado $x(t) = \phi(t, t_0, x(t_0), u(.))$ ou pela equação diferencial $x'(t) = f(t, x(t), u(t))$, sendo a equação de saída dada por $y(t) = g(t, x(t), u(t))$.

Neste quadro, a especificação do problema de controlo de um sistema dinâmico consiste em definir o conjunto de trajectórias no espaço de estados que são admissíveis através da consideração de diversos tipos de restrições envolvendo as variáveis de controlo, de estado, e controlo e estado que especificam objectivos e requisitos comportamentais. Adicionalmente, a especificação de requisitos comportamentais poderá obrigar à optimização de

um critério para a avaliação do desempenho do sistema na síntese da estratégia de controlo.

Controlar um sistema dinâmico é pois escolher uma estratégia de controlo que determina, do conjunto de comportamentos possíveis, um que satisfaz as restrições impostas para o sistema de forma a que a trajectória do estado pertença ao conjunto objectivo. De acordo com o tipo ou natureza do subconjunto das trajectórias objectivo várias estratégias de controlo podem ser desenvolvidas.

8.1.1 Estratégias de controlo

Uma classificação genérica permite catalogar os tipos de problemas de controlo e deste modo as estratégias de controlo associadas, nas seguintes classes:

- Seguimento (ou "tracking"). O objectivo da acção de controlo consiste em aproximar a trajectória executada de uma dada uma trajectória de referência do estado ($z \in H_x$). Tipicamente, define-se uma tolerância na forma de um conjunto objectivo $\{h_x : |h_x - z| \leq \varepsilon, \varepsilon > 0\}$ no qual a trajectória do estado é mantida pela estratégia de controlo.
- Um caso particular da estratégia anterior é o controlo em posição ou regulador, cujo o critério é compensar as perturbações por forma a manter o estado no ponto ($x = x_0$, $t > t_k$). Deste modo é garantida a imunidade do sistema a perturbações que poderão ter a sua génese em ruído e ou nas incertezas do modelo do processo a controlar.
- Evitar um conjunto. Dado um conjunto de estados $Q \subset X$, o conjunto objectivo é definido como $\{h_x \forall t . h_x(t) \notin Q\}$, ou seja, nenhuma trajectória deverá atingir um estado em $Q \subset X$.

Em muitas aplicações práticas, as estratégias a serem geradas pelos controladores envolvem a combinação das propriedades inerentes aos reguladores e aos servos complementando com as características necessárias ao "state-avoidance problem".

Uma vez caracterizado o conjunto objectivo, há que sintetizar os controladores por forma a gerar sinais de controlo que circunscrevem os comportamentos possíveis do sistema a trajectórias admissíveis. Este é o papel desempenhado pelas leis de controlo. Por forma a garantir a robustez a

perturbações na forma de ruído ou devidas a incertezas na modelização, as leis de controlo que estamos interessados analisar e sintetizar são aquelas que utilizam a realimentação, ou seja, aquelas que, dado o valor para o estado e/ou saída correntes, determinam o valor da entrada a aplicar ao sistema.

São já clássicos na Teoria de Controlo uma vasta gama de resultados e métodos que permitem a análise da existência de leis de controlo que garantem certos tipos de comportamento. No que respeita, o atingir um dado objectivo, a noção de controlabilidade é essencial. Um ponto no espaço de estados $(t_i, x) \in T \times X$ é controlável face ao conjunto objectivo $O \subset T \times X$ se existir uma sequência de acções de controlo γ que num instante $t > t_i$ colocam o estado (t_i, x') no interior do conjunto $O \subset T \times X$, ou seja: $(t_i, x) \xrightarrow{\cdots\gamma\cdots} (t, x'), (t, x') \in O$. Um sistema diz-se completamente controlável se e só se é possível conduzir o sistema de um dado estado arbitrário $x \in X$ até um outro qualquer em tempo finito. Uma vez que nem sempre o estado está disponível, surge uma outra questão complementar que se denomina observabilidade e que consiste na possibilidade de determinar o estado do sistema a partir da saída deste.

8.1.2 Caracterização de acções de controlo

Nesta secção, referimos algumas propriedades gerais relevantes para estratégias de controlo de sistemas de locomoção num contexto suficientemente abstracto para abranger quer os sistemas discretos quer em tempo contínuo. Estas são a Invariância e Atingibilidade.

Dado um par {sistema de controlo, conjunto}, diz-se que satisfaz a propriedade de Invariância se existirem estratégias de controlo às quais correspondem trajectórias que se mantêm no conjunto dado. A invariância poderá classificar-se de fraca se pelo menos uma ou forte se todas as estratégias de controlo satisfizerem a referida propriedade.

Dado um par {sistema controlado, conjunto}, diz-se que satisfaz a propriedade de Atingibilidade significa que existe uma estratégia de controlo à qual corresponde uma trajectória que atinge o conjunto dado num intervalo de tempo finito. Obviamente, se o estado inicial do sistema pertencer ao conjunto dado, a definição fica trivial.

Importa também referir que, no contexto dos sistemas de controlo com acontecimentos discretos, a formulação para supervisor e a noção de controlabilidade apresentadas em [Ramadge 87a] permitem concluir que o supervisor é passível de, entre outras características, uma interpretação análoga à das acções do tipo proporcional e integral dos sistemas em tempo contínuo.

Diz-se que um sistema tem capacidade antecipativa se, em qualquer estado "não proibido", for possível ocorrer um acontecimento controlável antes de qualquer outro não controlável que promova a transição do sistema para outro estado "não proibido". Por "estado proibido" entende-se qualquer estado que, uma vez atingido, significa o não cumprimento das especificações definidas para o funcionamento do sistema.

8.2 Controladores para a execução de Manobras

Uma das principais tarefas da plataforma móvel será a da movimentação. Como já referimos nos capítulos de análise e de especificação, a síntese do movimento estruturar-se-á de forma hierárquica. Nos níveis superiores da hierarquia está a decisão da manobra activa. Como referimos, o movimento está fortemente associado às restrições físicas. Dessa forma, definimos padrões de movimento e, consequentemente, para cada padrão de movimento, uma manobra de forma a que o sistema evolua conforme a especificação e satisfazendo as restrições físicas. Deixando para secções posteriores o endereçamento da síntese de manobras e das estratégias de controlo elementar, vamos agora tratar da síntese do sistema que decide a manobra activa.

O controlador de execução de manobras recebe como referência um grafo de manobras que une o estado inicial ao final. Caso a organização do movimento ocorra "off-line" tal como no planeamento da missão, este grafo toma a forma de uma sequência. Se a sequência de restrições for definida "on-line" por um dado critério, o grafo toma a forma de múltipla escolha. Esta referência especifica o controlo em malha aberta. A saída deste controlador é um conjunto de manobras parametrizadas com as respectivas restrições topológicas. O controlador de manobras deverá ser completo, ou seja, o processo de engenharia de sistemas subjacente à definição do conjunto de manobras deverá ser levado a cabo de forma exaustiva.

Uma vez que é necessário concatenar as manobras, o controlador precisa de, em cada instante, ter em conta a manobra correntemente activa, a prévia e a seguinte para ser possível determinar como receber o veículo e o "entregar". O mapa de "realimentação" é definido recorrendo a um observador híbrido/discreto – autómato híbrido em que os acontecimentos são todos não controláveis – que determina a topologia corrente com base nos padrões definidos. Com base na referência corrente da malha aberta e a topologia que resulta do observador, é seleccionada a manobra que se adapta à topologia corrente e à referência de malha aberta.

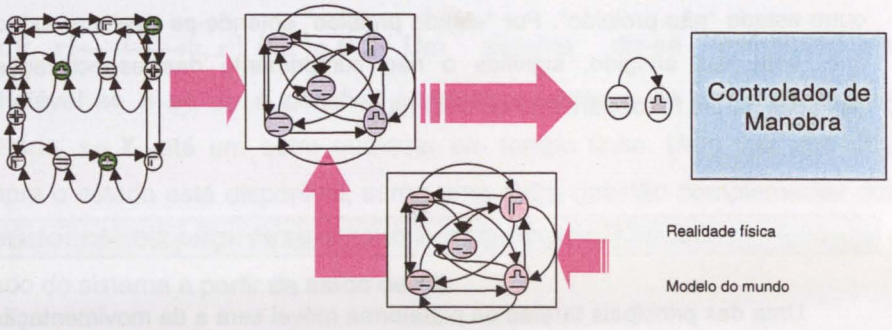


Figura 8.2-1

Organização de manobras e observadores híbridos.

O observador permite a determinação da topologia do padrão de restrições geométricas a considerar no período temporal corrente. No caso do ambiente autónomo industrial, o número de padrões é baixo, e a caracterização resulta fundamentalmente dos obstáculos que possam existir que impeçam a movimentação do sistema.

Notar que o carácter hierárquico da estrutura de controlo de movimento, permite gerir de forma eficiente do ponto de vista computacional o processo de escolha da manobra activa.

Um mecanismo para a implementação de estratégias de controlo em "tempo real" que permite entrar em consideração a, eventualmente elevada, variabilidade do ambiente consiste no "Horizonte Deslizante".

Este mecanismo consiste em definir a estratégia de controlo para um dado horizonte temporal fixo T de acordo com o mecanismo de síntese apropriado que tipicamente selecciona a estratégia de melhor desempenho de entre aquelas que satisfazem os critérios de correcção adoptados. Esta estratégia é aplicada durante o primeiro subintervalo T' de T, findo o qual é avaliada uma observação da evolução, incluindo o estado e o ambiente, do sistema. Se desta observação constatar um desvio relativamente ao comportamento previsto acima de uma certa tolerância, então, o deverá ser efectuada uma nova síntese – mais uma vez seleccionando de entre as

estratégias verificadas aquela com melhor desempenho – tendo em conta o estado observado para um intervalo de tempo T mas agora para aplicar durante um subintervalo T' a partir do momento actual, ou seja no fim do período T' considerado anteriormente.

Doutro modo, ou seja, se constatar que a execução está de acordo com a evolução esperada, então continuar-se-á a aplicar a estratégia de controlo previamente calculada por mais um subintervalo T'.

Em qualquer caso volta-se a observar a evolução do sistema e as operações descritas repetem-se até ao cumprimento integral da missão.

8.3 Controladores Híbridos para a definição de uma manobra

Um aspecto crítico do controlador híbrido consiste na criteriosa integração da gestão dos acontecimentos discretos relevantes sob o ponto de vista lógico com a evolução do sistema devida à necessidade de satisfazer as restrições físicas. Esta última refere-se, de forma abreviada, por evolução em tempo contínuo enquanto que à primeira se designa por evolução discreta. Assim a ideia subjacente à síntese de controladores híbridos assenta na definição "off-line" da estrutura de cada manobra em função da classe das restrições físicas associadas por forma a garantir a admissibilidade física. Seguidamente, caracterizam-se um conjunto de leis de controlo tendo em vista uma gama de objectivos de interesse e cujo domínio de validade inclui a região de operação a ser utilizada na manobra. É nesta fase que recorre a uma gama muito vasta de resultados e métodos da teoria do controlo que se garante a admissibilidade das restrições físicas e tecnológicas pois, neste processo, os modelos físicos permitem de forma realista entrar com a evolução do sistema no que respeita à satisfação das restrições e à quantificação do desempenho. Resta ainda uma etapa de articulação a qual define um conjunto de condições – pré-condições, co-condições e pós-condições – que se traduzem na configuração das transições (invariantes e guardas). Este conjunto é definido "off-line" no planeamento do movimento, sendo o elemento apropriado deste conjunto seleccionado "on-line" através de mecanismos de detecção da satisfação ou não das condições para a ocorrência das transições. Desse mapa de realimentação resulta a manobra de realimentação contínua a activar.

8.3.1 Mecanismos de síntese

Nesta secção descrever-se-á de forma sucinta e num contexto abstracto o mecanismo nuclear referente ao processo de síntese de estratégias de controlo. O carácter geométrico e o elevado nível de abstracção considerado na descrição deste mecanismo permite a sua consideração para sistemas cuja evolução está referida quer ao tempo quer aos acontecimentos discretos. Para a definição deste são necessários dois ingredientes:

- um conjunto fechado no espaço dos estados.
- uma estrutura que define o modo como evolui o estado.

Este último é tipicamente definido por uma equação ou uma inclusão diferencial e permite especificar o modo como a actuação, definida pelo valor da variável de controlo, afecta a evolução da variável de estado. Este dispositivo poderá ser de natureza muito variada: desde constituir uma descrição de um objecto em termos funcionais baseada nas leis da física, até traduzir uma evolução lógica subjacente à concepção do objecto em apreço, passando por caracterizações mistas. Em todo o caso, é, de um modo geral, possível associar a este dispositivo um mapa de transição de estado que é, naturalmente dotado da propriedade de semi-grupo [Kalman 69].

O primeiro item, o conjunto do espaço dos estados, é definido através dos ingredientes que constituem o ambiente do sistema e ou os objectivos para a acção de controlo. Assim o conjunto pode ser definido através de equações que determinam os pontos do espaço dos estados em que, por exemplo, o desempenho do sistema é máximo, o sistema tem a estabilidade garantida, a resposta do sistema satisfaz determinados requisitos, etc.

A relação entre o estado do sistema e o conjunto dada determina o tipo de síntese pretendida. Essencialmente, existem duas possibilidades que permitem enquadrar uma gama extremamente variada de problemas de controlo.

- Atingibilidade. O estado está no complementar do conjunto dado e pretende-se sintetizar uma estratégia de controlo que o conduza o estado ao interior do conjunto dado num intervalo de tempo finito.
- Invariância. O estado está no interior do conjunto dado e pretende-se sintetizar uma estratégia de controlo que mantenha o estado dentro do conjunto.

Em todo o caso, um mecanismo genérico de síntese do sinal de actuação "proximal aiming" poderá ser utilizado em quaisquer destas classes genéricas de classes de controlo e vai ser apresentado em diferentes contextos nesta secção.

8.3.2 Atingibilidade

Existe uma classe muito significativa de questões ligadas ao projecto e verificação de sistemas que podem ser formuladas como um problema de atingibilidade. A Atingibilidade consiste em verificar se um determinado ponto do estado pertence ou não ao conjunto de estados atingíveis de um determinado sistema dinâmico. Os estados atingíveis $R(t;t_0,x_0)$ são o conjunto de pontos de $\Re^n$ que podem ser atingidos de (t_0,x_0) no tempo t.

O problema da verificação nos sistemas dinâmicos pode ser formulado da seguinte maneira:

Seja $C(\cdot):[t_0,\infty)\rightarrow P(\Re^n)$ um mapa fechado que toma como valor conjuntos em $\Re$ representando uma propriedade de segurança de uma determinada manobra. Uma das possibilidades para verificação da propriedade é mostrar que existe um mapa aberto $A(\cdot)$ com um grafo conexo que $\forall t \geq t_0$, satisfaz $A(t) \subset C(t) \cap R(t;t_0,x_0)$.

A importância da questão de atingibilidade é demonstrada pela vasta literatura da Teoria do Controlo e Ciências de Computação que endereça a caracterização dos conjunto de estados atingíveis em sistemas dinâmicos. A caracterização dos estados atingíveis tem sido desenvolvida ao longo das seguintes direcções:

1) Formas integrais dadas por secções temporais dos tubos de atingibilidade que representam a evolução do espaço de estados atingíveis num determinado intervalo de tempo contínuo. Estes são definidos em tempo positivo por mapas "funnel" e, em tempo negativo, por mapas de solubilidade dos conjuntos de evolução. A propriedade de semigrupo destes mapas que caracterizam o domínio da solução é bastante conveniente para a síntese de estratégias de controlo realimentado [Kurzhanskii 97], para problemas de controlo com objectivo finito. Esta caracterização da síntese pode ser facilmente aplicada a sistemas dinâmicos com incertezas e imprecisões e quer no controlo em malha aberta, quer no controlo em malha fechada. Uma outra caracterização é possível em termos de funções de valor recorrendo à evolução em tempo quer positivo quer negativo.

2) As formas diferenciais são caracterizadas por desigualdades do tipo de Hamilton-Jacobi-Bellman, que têm sido objecto de extensa investigação quer no contexto da teoria de controlo quer no contexto da teoria dos jogos. Em [Clarke 96], a dinâmica do sistema é dada por

$$\dot{x}(t) \in F(t, x(t)), \qquad x(t_0) \in C$$

O conjunto do espaço de estados pode ser caracterizado por um proximal Hamilton-Jacobi-Bellman,

$$\theta + H(t, x, \varsigma) = 0$$

$\forall(\theta, \varsigma) \in \partial_p R(t,x), \forall(t,x) \in [t_0, \infty) \times \Re^n$ em que as condições fronteira $\lim_{t \downarrow t_0} R(t; t_0, C) = C$ existem por $R := \{(t,x) : x \in R(t : t_0, C)\}$, e o Hamiltoneano $H(t, x, p)$ é dado por $Max\{p \bullet v : v \in F(t,x)\}$ e $\partial_P S(s)$ o conjunto proximal normal ao conjunto S em s.

Uma abordagem construtiva é garantida pela formula exponencial deduzida em [Wolenski 90],

$$R(T;0,\xi) = \lim_{N \to \infty} R_N(T;0,\xi)$$

em que $R_N(T;0,\xi) = (I + \frac{T}{N} F)^N(\xi)$, I é o operador identidade, a potência $I + \frac{T}{N} F$ é no sentido da composição de multifunções e o limite é considerado no sentido de Kuratowski.

Do ponto de vista da complexidade, o problema da atingibilidade é não decidível para uma classe generalizada de sistemas híbridos [Deshpande 95]. As principais técnicas de analise de atingibilidade pode ser organizado em duas categorias [Graettinger 91], [Grossman 93]:

- Uma, utilizando métodos puramente simbólicos, baseada na existência de soluções analíticas para as equações diferenciais e na representação do espaço de estados no quadro da teoria decidível dos números reais.
- Outra baseada em métodos que combinam a integração numérica de equações diferenciais e a representação simbólica da aproximação do conjunto do espaço de estados por poliedros ou elipsoides.

A caracterização geométrica bem definida do conjuntos dos estados atingíveis dos sistemas dinâmicos lineares [Varaya 72] favoreceu o desenvolvimento de um grande número de técnicas construtivas para a computação das aproximações poliédricas [Graettinger 91], [Pereira 00], [Varaya 98]. A ideia básica fundamenta-se em estratégias "proximal" [Krasovski 88]], ou seja, a fronteira do conjunto dos estados atingíveis é determinada escolhendo em cada ponto o vector velocidade, que coloca a variável de estado no instante de tempo final o mais longe possível. Esta mesma ideia fundamenta o algoritmo apresentado em [Pereira 00] que combina a programação dinâmica e a fórmula exponencial, gerando as aproximações poliédricas interiores e exteriores ao conjunto de estados atingíveis de sistemas não lineares.

Em [Kurzhanski 97] são utilizadas equações (evolução elipsodal) sobre conjuntos e adoptados os cálculos respectivos. Do ponto de vista computacional, esta abordagem apresenta-se como vantajosa relativamente às abordagens poliédricas, contudo a aproximação pode ser mais conservativa.

8.3.3 Invariância

A invariância é uma noção clássica na teoria dos sistemas dinâmicos. Tendo em vista explicitar a sua importância no projecto de sistemas de controlo, consideremos o seguinte problema: um veículo é controlado por forma a que o estado do sistema se mantenha numa determinada região. No sistema Po-Robot podia considerar este se devia manter afastado de qualquer obstáculo físico por forma a poder executar a manobra seguinte sem entrar em colisão.

A interpretação dos conjuntos de nível da função valor em termos do conjunto dos estado atingíveis fornece, no contexto dos sistema híbridos, um enquadramento geométrico bastante conveniente para acomodar as proposições lógicas, que surgem das especificações.

Para cada manobra, podemos construir uma função de Lyapunouv que define ao nível dos conjuntos S_0, S_1, S_2 e S_3 (estes são compactos) no espaço $(t \times x)$ (em que $x \in \Re^n$ e denota a variável de estado), respectivamente.

- O conjunto óptimo, no sentido minmax, também designado "stable bridge" [Krasovski 88].

- O conjunto final, que é o conjunto de todos os pontos onde terminam as trajectórias que correspondem à manobra concluir com sucesso.
- O conjunto inicial que corresponde aos pontos que satisfazem as condições requeridas para iniciar a manobra.
- O conjunto estável que corresponde ao conjunto de pontos, que permite a esta concluir através de uma lei de controlo realimentado de baixo nível.

Notar que, se, por construção, as relações $S_0 \subset S_1 \subset S_2 \subset S_3$ são satisfeitas, então podemos garantir que os respectivos controladores satisfazem a correspondente relação de restrictividade.

Um subconjunto do espaço de estados é dito de invariante relativamente a um sistema dinâmico, se, caso contenha o valor da trajectória num determinado instante então contêm todos os valores da trajectória. A qualificação de positiva e negativa é considerada, em função da direcção do tempo. Complementarmente relativamente a um sistema dinâmico um subconjunto invariante também pode ser classificado como forte ou fraco, se respectivamente contém todas as trajectórias ou pelo menos uma trajectória. Este último conceito coincide com a noção de viabilidade [Aubin 91].

A [Aubin 84], [Aubin91], [Clarke 96b] a condição tangencial

$$F(x) \cap T_S^B(x) \neq 0,$$

em que $T_S^B(x)$ é o cone tangente de Bouligand relativo a S em x, é a condição necessária e suficiente para a existência da invariância fraca, ou para a viabilidade do conjunto S relativamente ao sistema dinâmico $x \in F(x)$. Adicionais caracterizações do conjunto de estados atingíveis pode ser observado em [Clarke 98]. Em particular, é demonstrado que a invariança fraca do par (F,S) é equivalente a qualquer uma das seguintes condições:

$h(x,\varsigma) \leq 0, \forall x \in S, \forall \varsigma \in N_S^P(x)$; e

$\forall x_0 \in S, \forall \varsigma > 0, S \cap R(t_0 + \delta; t_0, x_0) \neq 0$, para algum $\delta \in (0,\varepsilon)$.

Em que $N_S^P(x)$ e $h(x,\varsigma)$ são respectivamente, o cone proximal normal de S e o mais pequeno Hamiltoniano de F. As definições, propriedades e condições suficientes para a existência de invariância forte

estão desenvolvidas em [Clarke 98]. A prova dessas condições envolve a construção da função proximal de realimentação, $f_p(x) \in F(x)$, para qualquer ponto do espaço de estados ou seja

$$f_p(x) \text{ minimiza } v \rightarrow v \cdot (x - s),$$

para algum $s \in S$ tal que $x - s \in N_S^P(s)$, sendo $N_S^P(s)$ o cone gerado pela proximal normal a S em s, o que mostra o papel da invariância a síntese do controlo.

A noção de invariância é endereçada no contexto dos sistemas híbridos em [Aubin 00] pela formulação de um problema de controlo óptimo. A ideia principal é restabelecer a viabilidade de um conjunto arbitrário K, através da reinicializarão da trajectória do estado por um novo mapa sempre que um conjunto final seja atingido. Todavia uma trajectória típica apresenta descontinuidades associadas às transições discretas do autómato híbrido. Neste trabalho, a estabilidade no sentido de Lyapounov resulta do teorema da viabilidade e da caracterização da função valor que obtida como numa extensão dos resultados correspondentes no contexto da teoria da viabilidade.

Uma análise do papel da invariância, e em particular do o associado à invariância positiva, no domínio do controlo é apresentada em [Blanchini 99], onde os tópicos abordados são: o controlo com restrições; a análise de robustez; a síntese do controlo; a rejeição de perturbações e a optimização; e métodos construtivos baseados nas aproximações poliédricas ou elipsoidais na determinação dos conjunto invariante para a síntese do controlo.

8.3.4 Controlo por realimentação

Os problemas como a regulação, seguimento e atenuação das perturbações, são tipicamente endereçados por técnicas de síntese de controlo realimentado que condicionam o comportamento do sistema dinâmico, de tal forma que uma função de desempenho é optimizada (controlo realimentado óptimo), ou o estado é mantido num determinado conjunto (controlo de posição). Em ambos os casos a incerteza pode ser incorporada enquadrando o problema na teoria dos jogos [Kurzhanskii 97].

Os métodos de Lyapounov, a linearização exacta por realimentação, técnicas lineares aplicadas a sistemas dinâmicos auxiliares obtidos por linearização da dinâmica na vizinhança do ponto de operação, e o "gain-

scheduling" de controladores lineares para a utilização em vários pontos de funcionamento, são algumas das principais abordagens para as estratégias de controlo com objectivos semelhantes ao controlo de posição. Focaremos simplesmente os tópicos principais na análise deste problema de controlo.

Consideremos o problema de estabilização de um sistema dinâmico um $\dot{x} = f(x,u), x(0) = x_0$ assumindo que este possui um ponto de equilíbrio x*. O objectivo consiste em escolher uma função de controlo $u(t) \in \Omega$ como função da variável de estado, ou seja $u = k(x)$, por forma a que a trajectória convirja assimptoticamente para o ponto de equilíbrio apesar da presença de perturbações. Para tal necessitamos da definição de função de Lyapounov.

Uma função regular $V : \Re^n \rightarrow \Re$ é uma função de controlo de Lyapounov, se dotada das seguintes propriedades.

- positiva definida – $V(x) \leq 0 \;\; e \;\; V(x) = 0 \;\; se \;\; x = 0$
- própria – $\lim_{|x| \rightarrow \infty} V(x) = \infty$, e
- decresce infinitesimalmente, ou seja, existe uma função W contínua positiva em $\Re^n - 0$ tal que $\min_{u \in \Omega} \{\nabla V(x) \cdot f(x,u)\} \leq -W(x), \forall x \neq 0$

O método de projecto de Lyapounov envolve dois estágios:

- encontrar a função de controlo de Lyapounov.
- seleccionar a função de controlo realimentado $u = k(x)$, que satisfaz a condição $\nabla V(x) \cdot f(x,u) \leq -W(x), \forall x \in \Re^n$.

A existência de uma função regular de Lyapounov, a regularidade da função contínua de realimentação, o conceito da solução de uma equação diferencial ordinária descontínua, e a robustez aos erros no contexto da descontinuidade são algumas de um grande número de questões que têm alimentado a activa investigação que nesta área se tem desenvolvido.

As condições de Brockett [Rudie 95] determinam que um sistema dinâmico $\dot{x} = f(x,u)$ é estável por uma realimentação contínua, se e só se $f(N,\Omega)$ contêm a origem para uma qualquer vizinhança de N em 0. Como contra exemplo, pode-se apresentar o famoso exemplo do integrador não holonómico apesar de ser estável assimptóticamente. No entanto, em [Clarke 97] é mostrado que o integrador não holonómico, possui uma lei de

realimentação estável descontínua, em que o conceito da solução é definido da seguinte forma, $x := \lim_{N\to\infty} x^N$ em que x^N satisfaz

$$\dot{x}^N = f(x^N(t), k(x_i^N)) \qquad \forall t \in [t_i, t_{i+1}),$$

sendo $x_i^N = x^N(t_i)$ com $\pi_N = \{t_0, t_1, ..., t_N\}$ é a partição to intervalo de tempo definida tal que $\max_i |t_i - t_{i-1}| \to 0$ como $N \to \infty$ [Clarke 98]

Em [Sontag 83] demonstra-se que o integrador não holonómico possui sempre um função contínua de Lyapounov que satisfaz a seguinte condição

$$\inf_{u\in\Omega} DV(x, f(x,u)) \le -W(x) < 0 \qquad \forall x \neq 0$$

onde $DV(x; f)$ representa a derivada de Dini em x na direcção de f (ver [Clarke 98]). Em [Clarke 00], é mostrado que existe sempre uma função local Lipschitz Lyapounov que "praticamente" estabiliza o sistema, ou seja garante a existência de função de realimentação que conduz o sistema a uma vizinhança de um ponto de equilíbrio em tempo finito e, em [Rifford 00] é mostrado que esta função existe sempre para o integrador não-holonómico.

A robustez aos erros de medida é garantido para funções de Lyapounov regular, sendo a robustez nos outros casos conseguida apenas se for considerada a estabilização dinâmica.

A construção da função de Lyapounov continua a ser uma questão extremamente crítica no projecto de controladores. Ainda não existe um método sistemático de o fazer a não ser quando existe uma interpretação física do problema clara, ou então a estrutura particular da dinâmica do sistema de controlo o permite. Nesta última classe de problemas, assentam as metodologias de "backstepping", e o "sliding-modes".

8.3.5 Controlo óptimo

A optimalidade é uma consideração frequente na definição de estratégias de controlo. Assim, dois exemplos típicos e muito significativos são:

- Garantir um comportamento agressivo o que, tipicamente, corresponde a definir manobras ou a gerar trajectórias em tempo mínimo.

- Maximizar a duração da missão de um veículo autónomo num sistema de vigilância, o que se consegue minimizando o consumo de energia.

A optimização dinâmica é precisamente uma área da teoria do controlo em que a estratégia de controlo é sintetizada tendo em vista optimizar um critério de desempenho e simultaneamente satisfazer as restrições estáticas e dinâmicas. Estas restrições reflectem a realidade física dos dispositivos que constituem o sistema a ser controlado bem como a especificação da missão que se deseja realizar. Mais ainda, dado que a formulação do problema de controlo óptimo inclui ingredientes que reflectem objectivos e restrições, as técnicas de optimização também poderão ser utilizadas para efectuar a coordenação de sistemas dinâmicos independentes.

Em ambos as classes de sistemas endereçadas nesta tese, mas com extrema pertinência para os veículos submarinos autónomos, a escassez de recursos impõe que a estratégia de controlo seja escolhida de forma a que também seja optimizada uma certa função. Esta deverá ser criteriosamente escolhida para que, no decurso da execução das missões previstas, se verifique pelo menos uma boa gestão de recursos utilizados. A estratégia de controlo é a solução do problema de controlo óptimo, de que uma versão simples é:

(P) Minimizar $g(x(T))$

satisfazendo as restrições:

$$dx/dt = f(t,x,u),$$

$$(x(0),x(T)) \in C,$$

$$u(t) \in \Omega.$$

Embora, exista uma vasta classe de questões fundamentais para o endereçamento das aplicações, as duas classes de caracterizações das soluções do problema (P) mais significativas são:

• Condições necessárias de optimalidade (ver [Clarke 98], [Vinter 00]), das quais a melhor conhecida é o Princípio do Máximo, que permitem identificar os pares trajectória e estratégia de controlo candidatos a óptimo. Assim se (u^*,x^*) é solução do problema de controlo óptimo (P), então existe um número positivo λ e uma função p satisfazendo

$$- dp/dt = p\, \nabla_x f(t,x^*(t),u^*(t)),$$

$$(p(0),-p(T) \in N_C(x^*(0),x^*(T)) + \lambda\, (0, g_x(x(T)))$$

$$u^*(t) \text{ maximiza } v \rightarrow H(t,x^*(t),p(t),v) \text{ em } \Omega,$$

onde $H(t,x,u,p)=\langle p,f(t,x,u)\rangle$.

A aproximação computacional da solução é obtida por eliminação sucessiva de trajectórias candidatas de custo decrescente.

• Condições suficientes de optimalidade (ver [Clarke 98], [Vinter 00]), tipicamente, dadas na forma da equação às derivadas parciais de Hamilton-Jacobi,

$$\nabla_t V(t,x) + \nabla_x V(t,x) f(t,x,u) \leq 0,$$

$$v(T,x(T))=g(x(T)),$$

na qual assenta o método de resolução da programação dinâmica. Uma vez obtida a função valor como solução da equação de Hamilton-Jacobi, a trajectória óptima corresponde à função de controlo em malha fechada $u(t)=k(t,x(t))$ que maximiza $H(t,x,u,p)$ quando $(t,x(t),p(t))= (t,x(t),-\nabla_x V(t,x(t)))$.

8.3.6 Coordenação de sistemas dinâmicos

A coordenação de sistemas dinâmicos pode ser executado quer pela partilha das restrições no controlo, quer pela partilha das funções objectivo.

Enderecemos então estas duas abordagens. As necessárias condições de optimalidade, foram derivadas para várias classes de problemas que são do interesse dos sistemas híbridos.

Multiprocessos. Esta classe de problemas de optimização dinâmica já foi endereçada em [Clarke 89a] e consiste na consideração de um número finito de sistemas dinâmicos em que as estratégias de controlo foram determinadas tendo em conta a *minimização de uma função custo global*, e em que as restrições temporais e de estado foram satisfeitas. De facto desta abordagem resulta o controlo coordenado de vários sistemas dinâmicos através da articulação das restrições dos pontos finais de cada sistema. Em [Clarke 89b], o princípio é utilizado quer no contexto do controlo e das inclusões diferenciais.

Optimização em dois níveis. Problemas de optimização hierárquica já foram considerados por vários autores [Sandell 78], [Varaya 97]. Em [Ye 97], é demonstrada uma aplicação do princípio do máximo a um problema de controlo em dois níveis. O controlo de alto nível é realizado no sentido de *minimizar uma função global* de custo, *o controlo no nível hierárquico mais baixo é*

realizado minimizando a sua própria função custo. A coordenação neste paradigma é realizada através das funções de custo. Os problemas de coordenação que endereçamos nesta tese podem ser resolvidos recorrendo aos paradigmas descritos anteriormente, e pode ser formulado da seguinte maneira:

$$(P_H)\ Minimize \quad J_H(\{e^i\}, u, \{v^i\})$$

$$sujeito\ a \quad u \in U \text{ e as soluções de}$$

$$(e^i, v^i) \text{ a } (P_L^i(u, \{e^j\} j \neq i))$$

Em que o problema $(P_L^i(u, \{e^j\} j \neq i)), i = 1,..., N$, é definido por

$$Minimize \quad J_L^i(e^i, u, v^i)$$

$$sujeito\ a \quad \dot{x}^i(t) = f^i(t, x^i(t), u(t), v^i(t))$$

$$\lfloor t_0^i, t_1^i \rfloor - a.e.$$

$$w^i \in W^i \text{ e}$$

$$\{e^i\} \in \Lambda,$$

Em que

$e^i = col(t_0^i, x^i(t_0^i), t_1^i, x^i(t_1^i)) \subset \Re^{2n_i+2}$,

$\{e^i\}$ representa o produto cartesiano $e^1 \times \cdots \times e^N$,

$J_H(\{e^i\}, u, \{v^i\}) := f_0(\{e^i\})$, $J_L^i(e^i, u, v^i) := g^i(e^i)$,

$U := L^2([t_0, t_1; \mathrm{M}]$, ω^i é o conjunto das funções mensuráveis no intervalo $\lfloor t_0^i, t_1^i \rfloor$, tomando valores no conjunto limitado $W^i \subset \Re^m$, sendo $\Lambda \subset \Re^\alpha$ (com $\alpha = \sum_{i=1}^N (2n_i + 2)$), M, e W^i conjuntos limitados, e para $i = 1,..., N$, $\lfloor t_0^i, t_1^i \rfloor \subset [t_0, t_1]$.

Notar que a coordenação resulta quer da função custo quer das restrições associadas aos pontos finais. Se as funções de custo $(P_L^i(u, \{e^j\}_{j \neq i}))$ $i = 1,..., N$ são ignoradas e o objectivo é reduzido a encontrar um controlo para o conjunto dos sistemas dinâmicos que minimiza o custo

global, então o problema reduz-se ao à abordagem multiprocessos [Clarke 89b].

Esta formulação endereça somente o caso descentralizado pois é necessário ter disponível toda a informação para a síntese da coordenação.

8.4 Exemplo de utilização da estrutura oferecida pela arquitectura para a implementação de uma manobra

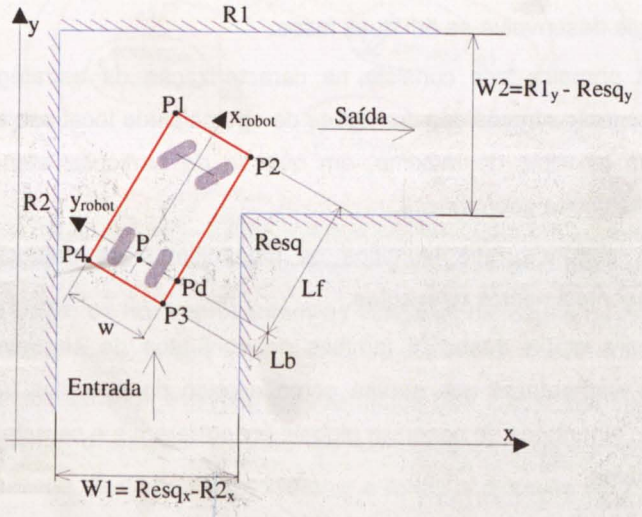
Vamos agora exemplificar como é possível desenvolver e implementar uma lei de controlo híbrida, bem como aplicar a metodologia proposta por [Almeida 97] para o projecto desse mesmo controlador utilizando a arquitectura por nós proposta no capítulo 7.

8.4.1 Formulação do Problema – Uma Curva em 'L'

O problema associado à manobra de uma Curva em 'L' referenciado no capítulo 4 cuja a origem se encontra em [Almeida 97] consiste na síntese da solução de controlo para uma plataforma móvel sujeita às restrições geométricas exemplificadas na figura seguinte, por forma a que esta consiga, partindo de uma configuração inicial (a configuração pertence ao conjunto das possíveis configurações na zona de entrada), atingir uma configuração final (pertence a um conjunto de configurações na zona da saída), preservando a propriedade em que a restrição caracterizada pela esquina não afecta o movimento de saída.

Figura 8.4-1

Problema Curva em L



O veículo considerado é não holonómico e o controlo da direcção do tipo rodas direccionais. Este último factor tem como consequência restrições no raio de curvatura que o veículo pode efectuar.

Na análise e síntese da manobra [Almeida 97] considerou o seguinte modelo cinemático:

$$\begin{cases} \dot{x} = v.\cos(\theta) \\ \dot{y} = v.\sin(\theta) \\ \dot{\theta} = v.c \end{cases}$$

Onde ***v*** e ***c*** são os controlos satisfazendo as restrições:

$$v \in [-v_{\max}, v_{\max}] \quad \text{e} \quad c \in [-c_{max}, c_{max}]$$

e onde,

- (x, y, θ) é o estado do sistema, onde o par (x, y) representa as coordenadas do ponto central no eixo traseiro do veículo, ponto este que corresponde à origem do referencial de coordenadas do veiculo, correspondendo θ à diferença angular entre o referencial do veículo e o global.
- v é a velocidade angular, e
- c é a curvatura instantânea a que veiculo esta sujeito, que é dada pelo o inverso do raio instantâneo de curvatura.

8.4.2 Síntese do controlador híbrido

Em [Almeida 97] é proposta uma metodologia para o desenho da organização do controlador híbrido, através da síntese das estratégias de movimentação elementares e da determinação dos acontecimentos relevantes. Esta metodologia desenvolve-se em duas fases:

- A primeira fase consiste na caracterização da estratégias de controlo através das condições de optimalidade local sob a forma de principio do máximo, em que se dá particular atenção às restrições geométricas,
- A segunda fase identifica os mecanismos de detecção dos acontecimentos relevantes.

Da primeira etapa, resultam famílias de conjuntos de segmentos de movimentação elementares que servirá como espaço de pesquisa para, na segunda etapa, algoritmos de pesquisa globais encontrarem a e caracterizarem a trajectória óptima.

Estes procedimentos de pesquisa permitem a definição de conjuntos de condições que vão constituir os mecanismos para detectar os acontecimentos relevantes que vão fazer despoletar as transições entre os segmentos de movimentação elementares.

Deste modo, são definidas leis de realimentação contínua e discreta que permitem um encadeamento de segmentos de movimentação elementares, constituindo uma trajectória possível da configuração corrente até ao conjunto de configurações de destino e que optimize o critério definido.

No trabalho referenciado, esta metodologia é utilizada como guia para síntese e organização das leis de controlo e dos mecanismos de detecção dos eventos relevantes.

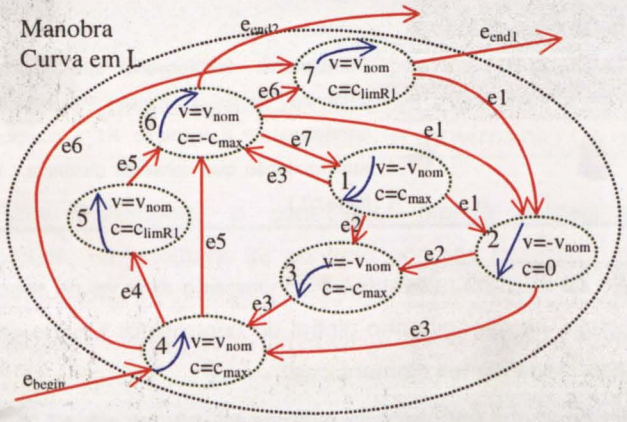
A existência de erros de execução e modelização, bem como de incerteza no posicionamento do veículo e das restrições, foram considerados na síntese do controlador híbrido conduzindo à introdução de flexibilidade nas situações críticas para assegurar a satisfação das restrições e evitando configurações que sejam sensíveis à perda de desempenho.

8.4.3 Organização do controlador híbrido

A organização global do controlador híbrido proposta está representada na figura seguinte.

Figura 8.4-2

Controlador híbrido para a manobra curva em L



Onde, os nós representam as várias modalidades de controlo elementar e os arcos representam as transições entre modalidades que ocorrem quando são detectados os eventos representados nos arcos. Em cada nó, o número representa o estado discreto do controlador e que corresponde a uma modalidade de controlo elementar e a forma e direcção do fluxo imposto pela

lei de controlo associada é representada por uma seta a azul. Junto a cada arco está identificado o número do acontecimento que vai despoletar a transição no controlador híbrido.

Na tabela seguinte encontram-se resumidas, as leis de controlo elementar desenvolvidas por [Almeida 97] onde se constata que existem dois tipos bem distintos de modalidades elementares. Um primeiro, onde o controlo contínuo é em malha aberta e assume valores de controlo extremos (modalidades 1, 3, 4 e 6). E outro, onde existe uma realimentação com base em certas restrições geométricas, tendo a lei de controlo o objectivo de manter as restrições quase activas, isto é, mover o veículo tangencialmente às restrições (modalidades 2, 5 e 7).

Tabela 8.4-1

Lista de leis de controlo para modalidades de controlo elementares.

Controlador Elementar	*Lei de Controlo*	*Descrição*
1	$f_{lim_w}(-v_{nom}, c_{max})$	
2	$f_{back_tan_Resq}(X_{robot}, R_{esq})$	Minimiza distância($\overline{P2Pd}$, R_{esq} – folga c2)
3	$f_{lim_w}(-v_{nom}, -c_{max})$	
4	$f_{lim_w}(v_{nom}, c_{max})$	
5	$f_{clim}(X_{robot}, R2)$	Componente feedforward que impõem uma dinâmica para o ponto P1 movimentar-se paralelamente à restrição R2 + componente de realimentação que minimiza distância (P1, R2 + folga c5)
6	$f_{lim_w}(v_{nom}, -c_{max})$	
7	$f_{clim}(X_{robot}, R1)$	Componente feedforward que impõem uma dinâmica para o ponto P1 movimentar-se paralelamente à restrição R1 + componente de realimentação que minimiza distância (P1, R1 – folga c7)

A detecção de eventos relevantes é conseguida através de mecanismos que avaliam o custo no desempenho global do controlador híbrido provocado por transições nas modalidades elementares.

Pode se verificar que enquanto os eventos e1, e3, e4, e6, e7, e_{end1} e e_{end2} foram desenhados por forma a anteciparem a activação de restrições geométricas na modalidade corrente, i.e., a violação dos invariantes dessa modalidade (logo com um horizonte temporal relativamente curto), os eventos e2 e e5, estão associados à detecção de acontecimentos optimizantes. Estes

eventos resultam da previsão envolvendo um horizonte temporal mais longo, e utilizando modelos do veículo bem como do próprio controlador híbrido.

Os eventos e1 a e7 e e_{end1} e e_{end2} são definidos na seguinte tabela.

Tabela 8.4-2

Lista de eventos para as modalidades de controlo elementares.

Eventos	Condição a verificar
e1	Activação da restrição geométrica R_{esq} pelo segmento $\overline{P2Pd}$
e2	Minimiza θ no início da próxima modalidade 2
e3	Activação da restrição R2 pelo ponto do veículo P4
e4	Activação da restrição R2 pelo ponto do veículo P1
e5	Optimiza o parâmetro de coordenação com uma próxima manobra caso o conjunto final possa ser atingido, ou Minimiza θ no início da próxima modalidade 2
e6	Activação da restrição R1 pelo ponto do veículo P1 e θ≤θini
e7	Activação da restrição R1 pelo ponto do veículo P1
e_{end1}	A restrição R_{esq} não pode restringir o movimento para a frente & θ∈ θfinal_set
e_{end2}	θ≤θlim

Onde *θini* and *θlim são definidos como*:

$$\theta_{ini} = arctg\left(\frac{L_f}{\frac{w}{2} - R_{min}}\right) \quad \text{e} \quad \theta_{lim} = \arcsin\left(\frac{Lf \cdot W2 - w \cdot \left(Xf^2 + w^2 - W2^2\right)}{Xf^2 + w^2}\right)$$

8.4.4 Implementação no enquadramento do controlador de recurso de movimento da classe 1C1SnA

Para implementar o controlador híbrido apresentado nos pontos precedentes, na estrutura de controlo por nós proposta (ver capítulo 7), é necessário definir as actividades dos estados representados à cor azul e definir as transições correspondentes aos arcos desenhados a vermelho.

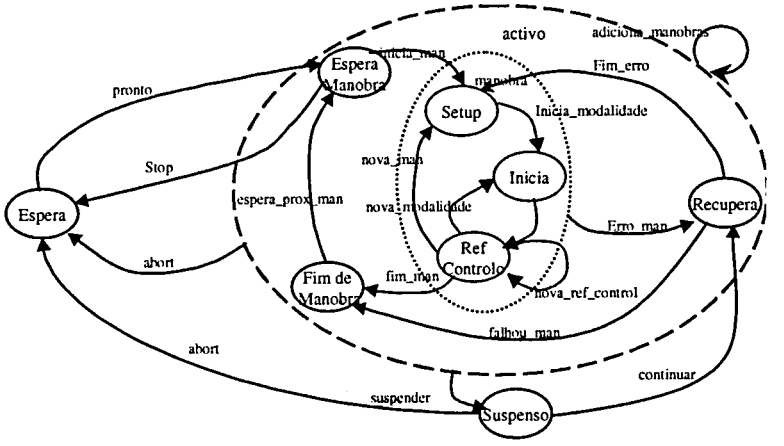


Figura 8.4-3

Implementação do controlador de recurso.

Os restantes estados e arcos são utilizados para actividades estruturais tais como, interface com a coordenação e encadeamento de manobras.

A actividade denominada ***setup*** garante o recrutamento de recursos do sistema, com especial destaque para os recursos sensoriais necessários à execução da manobra. Ou seja para cada manobra, quem realiza a implementação têm de indicar que algoritmos estimação vão estar activos, para a respectiva alocação física dos recursos.

Na actividade denominada ***Inicia*** são parametrizados as lei de controlo elementar e os mecanismos de processamento e de alocação sensorial. Complementarmente, esta actividade executa uma tarefa estrutural responsável pela indicação da lei de controlo de movimento activa e pela indicação dos acontecimentos relevantes que podem ocorrer. Essa tarefa passa simplesmente pelo preenchimento das variáveis do processo que gere a locomoção (uma fotografia das variáveis podem ser vista na figura seguinte) com o conteúdo da tabela seguinte.

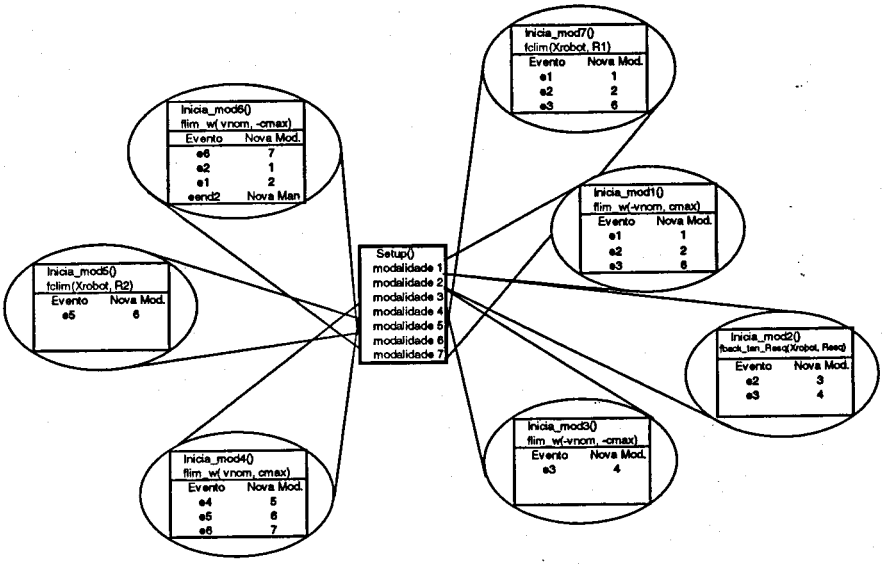
Controlador Elementar	*Lei de Controlo*	*Acontecimento a detectar / Modalidade seguinte*			
1	$f_{lim_w}(-V_{nom}, C_{max})$	e1 / 2	e2 / 3	e3 / 6	
2	$f_{back_tan_Resq}(X_{robot}, R_{esq})$	e2 / 3	e3 / 4		
3	$f_{lim_w}(-V_{nom}, -C_{max})$	e3 / 4			
4	$f_{lim_w}(V_{nom}, C_{max})$	e4 / 5	e5 / 6	e6 / 7	
5	$f_{clim}(X_{robot}, R2)$	e5 / 6			
6	$f_{lim_w}(V_{nom}, -C_{max})$	e6 / 7	e7 / 1	e1 / 2	e_{end2}
7	$f_{clim}(X_{robot}, R1)$	e1 / 2	e_{end1}		

Tabela 8.4-3

Transições ente modalidades de controlo elementares.

Figura 8.4-4

Implementação das manobras.



Por último a actividade ***Ref Controlo*** assegura que a lei elementar activa é a indicada e verifica as condições de disparo dos acontecimentos. Fazendo transitar a lei de controlo corrente para a lei de controlo seguinte ou fazendo transitar para uma nova manobra.

Esta página foi intencionalmente deixada em branco

9 CONCLUSÕES

Esta dissertação endereçou o desenvolvimento de sistemas–solução para duas classes de aplicações, o transporte industrial num ambiente semi-estruturado e a movimentação no ambiente subaquático. Estes distinguem-se de forma essencial no que respeita aos mecanismos de percepção e na forma de coordenação das actividades desenvolvidas pelos diversos dispositivos. Foi demonstrada a viabilidade de uma abordagem unificada para o projecto de ambas estas classes de sistemas substancialmente distintas, tendo no caso do sistema Po-Robot sido comprovado o sucesso desta. Importa referir que neste último cenário aplicacional o sistema desenvolvido foi validado por um utilizador final.

No desenrolar deste esforço, foi colocada a ênfase nos aspectos de carácter metodológico que, partindo do processo de engenharia de sistemas, ver [IEEE 98], refinou tendo em vista os aspectos específicos das classes de sistemas endereçadas. Essencialmente, estes envolvem a integração de todo um conjunto de ingredientes diversificados que incluem: o conceito de operação, garantia de propriedades dos modelos de forma construtiva, uniformidade da representação nas diversas fases do projecto,

Foi refinada uma arquitectura de controlo que permite uma gestão adequada da relação complexidade – expressividade que depende fortemente do contexto da execução. Mais ainda, sendo modular e escalável, esta arquitectura adequa-se bem ao controlo coordenado de múltiplos veículos ou dispositivos autónomos. Por outro lado, a generalização da ideia de utilizar a reconfigurabilidade da arquitectura como elemento da estratégia de controlo permite embeber esta actividade num contexto integrado de gestão dos recursos disponibilizados para a execução da missão. Em particular, foi apresentada um dos componentes.

A arquitectura implementável constitui um dos componentes fundamentais para o quadro de projecto de sistemas autónomos que permite a articulação coerente dos problemas de controlo de diversos tipos.

Foram identificados dois quadros formais de teoria de controlo, um no contexto de sistemas discretos e outro no da optimização dinâmica que melhor preenchem os requisitos que foram identificados. Em particular, o carácter geométrico associado aos resultados de optimização dinâmica, permitem uma fácil e natural formulação dos problemas de controlo colocados.

Paralelamente identificamos um cenário (fontes de água doce submarina) de grande importância na estrutura do desenvolvimento humano e sustentabilidade da terra, em que o recurso à intervenção de veículos

submarinos autónomos pode incrementar o alcance da monitorização e estudo destes fenómenos. Das singularidades deste cenário e dos requisitos operativos e técnicos que colocamos resultou uma inovadora estratégia de pesquisa. È de salientar que um dos aspectos relevantes para a solução foi a utilização de técnicas de movimentação em formação para superar as dificuldades de comunicação no ambiente submarino.

No que respeita ao trabalho futuro, importa considerar as seguintes perspectivas imediatas do trabalho até agora realizado:

- Formular e sintetizar os problemas de controlo especificados
- Implementar as soluções da robótica submarina, nas futuras aplicações no âmbito LSTS [LSTS 00], e, em particular no projecto PISCIS [PISICIS 02].
- Automatização das fases de projecto desde a especificação conceptual do problema até à geração automática de código.
- Conceber linhas orientadoras para a síntese integrada de hardware/software e estratégias de controlo.
- Concepção e desenvolvimento de dispositivos específicos para a percepção e navegação.
- Desenvolvimento de uma "test-bed" para a simulação, análise e síntese de sistemas distribuídos com "hardware-in-the-loop"

Quanto à investigação de carácter conceptual, importa referir que o conceito de autonomia e as características de comportamento condicionadas pela satisfação de restrições rigorosas em conjugação com a juventude da área têm vindo a proporcionar cenários de investigação aliciantes. Contudo, vamos limitarmo-nos a apresentar algumas sugestões directamente relacionadas com o trabalho.

- Formulação e síntese do problema de controlo coordenado de diversos veículos e dispositivos autónomos.
- Formulação e solução do problema de estimação descentralizada e distribuída.
- Extensão deste quadro para o caso de sistemas autónomos envolvendo um conjunto de veículos com uma elevada heterogeneidade.

Esta página foi intencionalmente deixada em branco

REFERÊNCIAS

[Agre 90] Agre, P. and Chapman, D.: "What Are Plans for?", Robotics and Autonomous Systems 6, pp. 17-34, 1990

[Alami 91] Alami, R., Chatila, R. and Freeman, P.: "Task Level Teleprogramming for Intervention Robots", Procs of Mobile Robots for Subsea Environments; IARP, Monterey, CA, pp 119-136, 1991

[Albus 88] Albus, J.: "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles", NIST Tech. Note 1251, Washington, DC, Sept., 1988

[Albus 90] Albus, J. and Quintero R.: "Towards a Reference Model Architecture for Real-Time Intelligent Control Systems (ARTICS)", Robotics and Manufacturing, New York, ASME, vol.3, 1990

[Albus 91] Albus J.: "Outline for a Theory of Intelligence," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, nº. 3, pp. 473-509, May/June 1991

[Albus 94] Albus J.: "A Reference Model Architecture for Intelligent Systems Design", NISTIR 5502, National Institute of Standards and Technology, Gaithersburg, MD, Sept. 1994

[Alegria 95] Alegria, P.: "Navegação Acústica em Ambientes Estruturados", Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, Porto, Mar. 1995

[Almeida 96] Almeida, J. M. "Controlo Híbrido de Robots Não-Holonómicos", Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, Porto, 1996

[Almeida 97] Almeida, J., Pereira, F., Sousa, J.: "A Hybrid System Approach to Feedback Control of a Non-Holonomic Car-like Vehicle", 5th IEEE Mediterranean Conference on Control and Systems, Cyprus, July, 1997

[Alur 93] Alur, R., Courcoubetis, C., Henzinger, T. and Ho, P.: "Hybrid Automata: An Algorithm Approach to the Specification and Verification of Hybrid Systems", Hybrid Systems, R. Grossman, A. Nerode, R. Ravn and H. Rischel (eds), Lect. Notes in Comp. Sci. nº 736, pp. 209-229, Springer Verlag, 1993

[Arkin 87] Arkin, R.: "Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behaviour", IEEE International Conference on Robotics and Automation, pp. 264-271,March 1987

[Arkin 97] Arkin, R. and Balch, T.: "AuRA: Principles and Practice in Review", Journal of Experimental and Theoretical Artificial Intelligence, 1997

[Aubin 84] Aubin, J-P. and Cellina, A.: Differential Inclusions: Set-valued Maps and Viability Theory, Springer-Verlag, 1984

[Aubin 91] Aubin, J-P, Viability Theory, Birkhauser, 1991

[Aubin 00] Aubin, J-P.: "Inpulse Differential Inclusions and Hybrid Systems: A Viability Approach", Lecture Notes, University of California at Berkeley, 2000

[Beer 90] Beer, R., Chiel and H., Sterling, R.:" A Biological Perspective on Autonomous Agent Design", Robotics and Autonomous Systems 6, 1990

[Beer 92] Beer, R., Chiel, H., Quinn, R., Espenchied, K. and Larsson, P.:" A Distributed Neural Network for Hexapod Robot Locomotion", Neural Computation 4, 1992

[Berns 95] Berns, K., Dillman, R. and Piekenbrok, S.:" Neural Networks for the Control of a Six-legged Walking Machine", Robotics and Autonomous Systems 14, 1995

[Blanchini 99] Blanchini, F: "Set Invariance in Control", Automatica, vol 35, pp.1747-1767, 1999

[Branicky 98] Branicky, M., Borkar, V. and Mitter, S.:"A Unified Framework for Hybrid Control: Model and Optimal Control Theory", IEEE Trans. on Automatic Control, 43:31-45, 1998

[Brooks 86a] Brooks , R., "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics and Automation, vol. 2, no. 1, March 1986.

[Brooks 86b] Brooks R., "Achieving Artificial Intelligence Through Building Robots," M.I.T. Artificial Intelligence Laboratory, Memo 899, May 1986

[Chatila 92] Chatila, R. Alami, Degallaix and H. Haruelle: "Integrated Planning and Execution Control of Autonomous Robot Actions", Procs. IEEE Int. Conf. on Robotics and Automation, Nice, France, 1992

[Chatila 95] Chatila, R., Ingrand, F. and Alami, R.: "Mission Planning and Execution Control for Intervention Robots", Procs. of the International Program Development in Undersea Robotics & Intelligent Control - a joint US/Portugal Workshop, March, 2-3, Lisboa, Portugal, pp. 38-43, 1995

[Clarke 89a] Clarke, F and Vinter, R.:"Applications of Optimal Multiprocesses", SIAM J. on Control and Optimization, vol. 27, no. 5, pp. 1048-1071, 1989

[Clarke 89b] Clarke, F and Vinter, R.:"Optimal Multiprocesses", SIAM J. on Control and Optimization, vol. 27, no. 5, pp. 1072-1091, 1989

[Clarke 96] Clarke, F.:"A Proximal Characterization of the Reachable Set", Systems and Control Letters, vol 27, pp.195-197, 1996

[Clarke 96b] Clarke, F. and Wolenski, P.:"Control of Systems to Sets and Their Interiors", Journal of Optimization Theory and Applications, vol. 88, no. 1, pp. 3-23, 1996

[Clarke 97] Clarke, F., Ledyaev, Y. Sontag, E. and Subbotin, A.:"Asymptotic Controllability Implies Feedback Stabilization", IEEE Trans. on Automatic Control, vol. 42, no 10, pp.1394-1407, 1997

[Clarke 98] Clarke, F., Ledyaev, Y. Stern, R. and Wolenski, P., "Nonsmooth analysis and control theory", Graduate Texts in Mathematics Vol. 178, Springer-Verlag, 1998

[Clarke 00] Clarke, F., Ledyaev, Y. and Stern, R.: "Feedback Stabilization and Lyapunov Functions", SIAM J. Control and Optimization, vol 39, no 10, pp. 25-48, 2000

[Cruz 99] Cruz, N.,Matos, A, Martins, A., Silva, J., Santos, D., Boutov, D., Ferreira, P. and Pereira, F. "Estuarine environment studies with Isurus, a REMUS class AUV", Procs. MTS/IEEE Oceans'99, Seatle, WA, USA, 1999

[Cruz 00] Cruz, N. , Madureira, L., Matos, A., Pereira, F. Lobo, "A versatile Acoustic Beacon for Navigation and Remote Tracking of Multiple Underwater Vehicles

[Daws 96] Daws, C. Olivero, A., Tripakis and S. Yovine, S., "The tool kronos", in Hybrid Systems III, Verification and Control, Lecture Notes in Computer Science 1066, Springer-Verlag, 1996

[Deshpande 95] Deshpande, A. and Varaya, P.:"Viable Control of Hybrid Systems", Hybrid Systems II, pp. 128-147, Springer, 1995

[Deshpande 97] Deshpande, A: "The DIADEM System for Real-Time Dynamic Event Management", Proceedings of the 1997 NATO Workshop on Discrete Event and Hybrid Systems, 1997

[Edwards 97] Edwards, S., Lavagno, L., Lee, E. and Sangiovanni-Vicentelli, A.: "Design of Embedded Systems: Formal Models, Validation and Synthesis", Proceedings IEEE, vol 85, no. 3, pp. 366-390, 1997

[Espiau 95] Espiau, B., Simon, D. and Kapellos, K.: "Formal Verification of Missions and Tasks", pre-print, INRIA, 1995

[Estrela 01] Estrela, J.:" Software para Controlo e Gestão de Missões de Veículos Autónomos", Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, 2001

[Fletcher 99] Fletcher, Barbara, "Chemical Plume Mapping with an Autonomous Underwater Vehicle",

[Fleury 96] Fleury, Sara: "Architecture de Contrôle Distribuée Pour Robots Mobiles Autonomes: Principes, Conception et Applications", PhD thesis, Université Paul Sabatier de Toulouse, Rapport LAAS N° 96156, 1996

[Fossen 94] Fossen, T.:Guidance and Control of Ocean Vehicles, John Wiley & Sons, 1994

[Franzemeier 00] Franzmeier, Mara," Groundwater discharge focus of Kahana Bay environmental", //www.soest.hawaii.edu/SEAGRANT/Makai/july00/3.html

[Gat 92] Gat, E.: "Integrating Planning and Reacting in a Heterogenous Asynchronous Architecture for Controlling Real-world Mobile Robots", in AAAI-92 Proceedings, AAAI Press, 1992

[Gat 94] Gat, E. and Dorais, G.: "Robot Navigation by Conditional Sequencing", IEEE International Conference on Robotics and Automation, pp. 1293-1299, 1994

[Giralt 79] Giralt, G., Sobek, R. P. and Chatila, R.: " A Multi-Level Planning and Navigation System for a Mobile Robot: a First Approach to HILARE", Procs. 6th IJCAI, Tokyo, Japan, August 1979

[Giralt 83] Giralt, G., Chatila, R. and Vaisset, M., "An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots", Procs 1st Int. Symp. on Robotics Research, M.I.T., Michael Brady, Richard Paul, Eds, 1983

[Golaszewski 87] Golaszewski, C., H., and Ramadje, P., "Control of Discrete Event Process with Forced Events", Procs 26 Conference on Decision and Control, Los Angels, Dec. 1987

[Graettinger 91] Graettinger, T. and Krogh, B.:"Hyperplane Method for Reachable Set Estimation for Linear Time-Invariant Systems", Journal of Optimization Theory and Applications, vol. 69, pp. 555-588, 1991

[Grossman 93] Grossman, R., Nerode, A., Ravn, A. and Rishel H. (eds), Hybrid Systems. Lecture Notes in Computer Science, 736, Springer-Verlag, 1993

[Harel 87] Harel, D., " Statecharts: A Visual Formalism for Complex Systems", Sci. Computer Programming, pp.231-274, July 1987.

[Harmon 92] Harmon, S. Y.: "Architectures: Designers versus Implementors", in Procs. of the Workshop On Architectures for Intelligent Control Systems , IEEE Int. Conf. on Robotics and Automation, Nice, France, May 1992, pp. 1-6.

[Healey 96] Healey, A., Marco, D. and McGhee, R.: "Autonomous Underwater Vehicle Control Coordination Using a Tri-Level Hybrid Software Architecture", IEEE Robotics and Automation Conference, Minniapolis, MI., pp. 2149-2159, 1996.

[Henzinger 93] Henzinger, T., Manna, Z., and Pnueli, A.: "Towards Refining Temporal Specifications into Hybrid Systems", in: Hybrid Systems, Grossman, R., Nerode, A., Ravn, A., and Rischel, H., eds., Lect. Notes in Comp. Sci. nº 736, Springer Verlag, 1993, pp. 60-76.

[Henzinger 94] Henzinger, T., Ho, P.: "Model Checking Strategies for Linear Hybrid Systems", 7th Int. Conf. Indust. & Eng. Applic. of A.I. & Expert Syst., Austin, TX, May 1994

[Henzinger 95] Henzinger, T., Ho, P. and Wong-Toi, H.:"Hytech: the Next Generation", Procs. IEEE Real-Time Systems Symposium, PTSS'95, pp.169-183, 1995

[Hoare 85] Hoare, C.A.R., "Communicating sequential processes", Prentice Hall International, Series in Computer Science, UK, 1985

[IEEE 98] Standard for Application and Management of The Systems Engineering Process, IEEE series 1220, IEEE Standards Association, 1998

[Inan 89] Inam, Kemal M., Varaiya, Pravin P.,"Algebras of Discrete Eveny Models", in Proc. IEEE, vol.77, no 1, pp. 24-38, Jan. 1989

[Inan 92] Inan, Kemal M."An Algebric Approach to Supervisory Control", in Mathematics of Control, signals, and Systems, Springer-Verlag, N°5, 1992, pag151-164, 1992

[ISePorto 01] http://lsa.isep.ipp.pt/iseporto

[Kalman 69] Kalman, R, Falb, P and Arbib, M.:Topics in Mathematical System Theory, McGraw-Hill, 1969

[Koseká 96] Koseca, Jana, "A framework for modeling and verifying visually guided agents: Design, Analysis, Experiments", Thesis for the degree of Doctor of Philosophy, at the University of Pennsylvania, Mar., 1996

[Kraovski 88] Krasovski, N. and Subbotin, A.: Game-theoretical Control Problems, Springer-Verlag, 1988

[Kurzhanskii 97] Kurzanskii, A. and Valyi, I.: "Ellipsoidal Calculus for Estimation and Control", Birkhauser 1997

[Kurzhanskii 00] Kurzanskii, A. and Varaya, P.: "Ellipsoidal Techniques for Reachability Analisys", Computation and Control, Lynch, N. and Krogh, B. (eds), Lecture Notes in Computer Science, pp. 202-214, Springer-Verlag, 2000

[Laengle 97]- Laengle, TH. And Lueth, T.C.: "Decentralized Control of Distributed Intelligent Robots and Subsystems", 1997

[Leewen 90] Leeween, J.: Handbook of theoretical computer science, Elsevier, 1990

[Lima 95] Lima, P.: "Feedback-Based Performance Improvement of Intelligent Control Systems", Procs. Int. URIC Program Development, US/Portugal Workshop, March, 2-3, 1995, pp. 105-110

[LSA 01] http://lsa.isep.ipp.pt

[LSTS 00] http://www.fe.up.pt/~lsts

[Lygeros 98] Lygeros, J, Godbole, D. and Sastry, S: "Verified Hybrid Controllers for Automated Vehicles", IEEE Transactions on Automatic Control, vol. 43, no. 4, pp. 522-539, 1998

[Lueth 94] Lueth, T.C. and U. Rembold: "Extensive Manipulation Capabilities and Reliable Behaviour at Autonomous Robot Assembly", IEEE Int. Conference on Robotics and Automation, San Diego, CA, 1994

[Manheim 01] MANHEIM, Frank T., KRANTZ, David E, SNYDER, Donald S., Zonge, MADSEN, John A., STURGIS, Brian, "Streaming Resistivity Surveys and Core DrillingDefine groundwater Discharge into costal Bays of The Delmarve Peninsula", GSA Annual Meeting, November 5-8, Boston,

Massachusetts 200, url: http://gsa.confex.com/gsa/2001AM/finalprogram/

[Marco 96] Marco, D.: "Autonomous Control of Underwater Vehicles and Local Area Maneuvering", PhD. Thesis, Naval Postgraduate School, Monterey, 1996

[Martins 96] Martins, A., "Modelização e controlo de um veículo autónomo terrestre", Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, 1996.

[Martins 99] Martins, A., Matos, A., Cruz, N. and F.L. Pereira, "IES: an open system for underwater inspection", Procs MTS/IEEE Oceans'99, Seattle, WA, USA, 1999

[Martins 01] Martins, A., Patacho, A., Silva, E., Lima, L., Baptista, J. and Almeida, J.:" Do Projecto à Implementação da Plataforma Móvel Utopia", 2001 Robotic Festival Conference, Guimarães, Portugal, April 2001

[Martins 02] Martins, A., Almeida, J., Silva, E. and Baptista, J.:"Design of the ISePorto Robocup Middle-Size League Robotic Soccer Team: Control, Localisation and Coordination", Procs. Mediterranean Conference on Control, Lisbon, Portugal, July 2002

[Mathworks 97] Real-Time Workshop User's Guide version 2.1, The Mathworks, 1997

[Mobies 01] http://www.rl.af.mil/tech/programs/MoBIES/

[Noreils 95] Noreils, F. and Chatila, R.: "Plan Execution Monitoring and Control Architecture for Mobile Robots", IEEE Trans. Rob. & Autom., vol.11, no.2, pp. 255-266, April 1995

[Ostroff 89] Ostroff, J.S.,"Synthesis of controllers for real-time discrete event systems", in Proceedings of the 28th Conference on Decision and Control, Tampa, Florida, pp. 138-144 Dec 1989

[Ostroff 90] Ostroff, J.S. and Wonham W., M. "A framework for real-time discrete event control", in IEEE Transactions on Automatic Control, vol 35, no. 4, pp. 386-397, April, 1990

[Ostroff 90] Ostroff, J.S. and Wonham W., M.: "A Framework for Real-Time Discrete Event Control", in IEEE Transactions on Automatic Control, vol. 35, no. 4, pp. 386-397, April 1990

[Ostroff 91] Ostroff, J. S.. "Verification of Safety Critical Systems Using TTM/RTTL ", in Proceedings of REX Workshop on Real-Time: Theory in Pratice, 1991

[Parker 00] Parker, L., "Current State of the Art in Distributed Autonomous Mobile Robotics", Distributed Autonomous Robotic Systems, Parker L., Barhen J., Bekey G. (eds) , Springer Verlag, 2000

[Payton 90] Payton, D.: "Internalized plans: A representation for action resources", in Designing Autonomous Agents, Maes, P. (ed), pp.89-104, The MIT Press, 1990

[Pereira 00] Pereira, F. and Sousa, J.: "On the Approximation of the Reachable Set Boundary", Procs. of Controlo 2000, Guimarães, IFAC, 2000

[Pereira 01] Pereira, F.:"Control Design for Autonomous Vehicles: A Dynamic Optimization Perspective", European Journal of Control, vol 7, pp.178-202, 2001

[Peterson 81] Peterson, J., L., "Petri Net Theory and the Modeling Systems", Englewood Cliffs, Prentice-Hall, 1981

[PISCIS 02] Project PISCIS proposal Report, LSTS-FEUP, 2001

[Po-Robot 95] "Multi-Purpose Portuguese Flexible Mobile Robot, Working Plan Proposal", Institute of Systems and Robotic, Porto, Portugal, 1995

[Po-Robot 97a] "Functional System": Po-Robot Sub-Task ST2/T3 Report, Institute of Systems and Robotic, Porto, Portugal, 1997

[Po-Robot 97b] Silva, E., Pereira, "Nível Físico do Sistema Romeu", Relatorio Po-Robot, ISR- Porto, 1997

[Po-Robot 98b]. Almeida, J., Martins, A., Sousa, D., Silva, E. Pereira, Alegria, P. and Silva, E. Pereira, "Sistema de Navegação da Plataforma Romeu", Relatorio Po-Robot, ISR, 1998

[Po-Robot 98c] Almeida, J., Alegria, P. and Silva, E. Pereira "Sistema de Navegação da Plataforma Romeu", Relatoirio Po-Robot, ISR, 1998

[Po-Robot 98e] Cunha, P. and Alegria, P, "Estação de Transferência do Sistema Romeu", Relatório Po-Robot, ISR, Porto, 1998

[Po-Robot 99a] Silva, E., Almeida, J, Martins, A., Alegria, P., Dinis, S., Silva, A., Cunha, P. and Pereira, F.: "Multi-Tast Real Time environment for An Autonomous Mobile Platform", Po-Robot Project Final Report, NATO, 1999

[Po-Robot 99b] Almeida, J, Silva, E., Martins, A., Alegria, P., Dinis, S., Silva, A., Cunha, P. and Pereira, F.: "Functional Layer of the Control Architecture", Po-Robot Project Final Report, NATO, 1999

[Puri 94] Puri, A. and Varaya., P."Decidability of hybrid systems using abstractions". In Proc. CAV. LNCS 818,Springer-Verlag, 1994

[Puri 95] Puri, Anuj, "Theory of Hybrid Systems and Discrete Event Systems", Thesis for the degree of Doctor of Philosophy, at the University of Berkeley California, 1995

[Ramadge 87a] Ramadge, P.J. and Wonham, W.M., "*Supervisory* Control of a class of discrete event processes", in SIAM Journal on Control and Optimization, 25(1): 206-230, 1987

[Ramadge 87b] Ramadge, P.J. and Wonham, W.M., "Modular Feedback Logic for Discrete Event Systems", in SIAM Journal on Control and Optimization, vol. 25, no. 5, pp.1202-1218, 1987

[Ramadge 89] Ramadge, P.J. and Wonham, W.M., "The Control of Discrete Event Systems", in Proceedings IEEE, vol. 77, no. 1, pp. 81-98, Jan. 1989

[Ramos 00] Cruz, N., Madureira, L., Matos, A., Pereira, F. Lobo, "Monitoring an Ocean Outfall using AUV", Url:www.fe.up.pt/~lsts

[Ravn 93] Ravn, A., Rischel, H. and Hansen, K.: "Specifying and Verifying Requirements for Real-Time Systems", IEEE Trans. Soft. Eng., vol.18, no.1, pp. 41-55, Jan. 1993

[Report 00] http://www.jhu.edu/~scor/pubs.htm, Scientific Comittee on Oceanic Research.

[Rifford 00] Rifford, L.: "Existence of Lipschitz and Semiconcave Control-Lyapunov Functions",SIAM J. Control and Optimization, vol. 39, no. 4, pp. 1043-1064, 2000

[Rosenblatt 95] Rosenblatt, J. and Thorpe, C: "Combining Multiple Goals in a Behaviour-based Architecture", International Conference on Integrated Robots and Systems (IROS), Aug. 1995

[Rosenblatt 97] Rosenblatt, J, "DAMN: A Distributed Architecture for Mobile Navigation", Ph.D. Thesis, The Robotics Institute, Carnegie Mellon University, Pittsburg, PA, 1997

[RTI 02] Joshi, R., Bose, A.and Breneman, S.,:"Software Platforms for Unmaned Systems", White Paper, Real Time Innovations Inc., Sunnyvale, CA, 2002

[Rudie 95] Rudie, K. and Willems, J.: "The Computational Complexity of Decentralized Discrete Event Control Problems", IEEE Trans. on Automatic Control, vol. 40, no. 7, pp. 1313-1319, 1995

[Saffioti 95] Saffioti, A., Konolige., and Ruspini, E.: "A Multivalued logic approach to integrating planning and control", Artificial Intelligence 76, pp.481-526, March 1995

[Samson 90] Samson, C, Espiau, B and Le Borge, M.: "Robot Control: The Task Function Approach", Oxford University Press, 1990

[Sandell 78] Sandell, N, Athans, M, Varaya, P. and Safonov, M.: "Survey of Decentralized Control Methods for Large Scale Systems", IEEE Trans on Automatic Control, vol. 23, no. 2, pp. 108-128, 1978

[Saridis 84]. Saridis, G. and Graham, J.: "Linguistic Decision Schemata for Intelligent Robots," Automatica, vol. 20, no 1, pp. 12-126, 1984

[Saridis 85] Saridis, G.: "Foundation of the Theory on Intelligent Control," Procs. IEEE Workshop on Intelligent Control, Rensselaer Polytechnique Institute, Troy, N.Y, pp. 23-28, 1985

[Saridis 89] Saridis, G.: "Analytic Formulation of the Principle of Increasing Intelligent with Decreasing Precision for Intelligent Machines," Automatica, vol. 25, no. 3, pp.461-467, 1989

[Sastry 99] Sastry, S. "Hierarchical Hybrid System Design of Flight Management Systems for Unmanned Aerial Vehicles", Notes from the Arrabida Summer Course in Advanced

Control Systems: System Theory on the Eve of the 21st Century, Lisbon, Portugal, 1999

[Schneider 95] Schneider, S., Chen, V. and Pardo-Catellote, G.: "The ControlShell Components -Based Real-Time, Programming System, IEEE Int. Conference on Automation, Nagoya, Japan, 1995

[Silva 94] Silva, E., Sousa, J., Pereira, F., Sequeira, J. and Ribeiro, I.: "On the Design of the Po-Robot System", Procs. of the IEEE Intelligent Vehicles Symposium 94, Paris, France, 1994

[Silva 98] Silva, E., and Pereira F.: "Verificação de Operadores de Composição de Tarefas", Relatório Interno ISR, Nov. 1998

[Silva 99] Silva, E. , Pereira, F. and Sousa, J. : "On the Design of a Control Architecture For an Autonomous Mobile Robot", em Advances In Intelligent Autonomous Systems, editado S.G. Tzafestas, Kluwer Academic Publishers, 1998

[Simon 93] Simon, D., Espiau, B., Castillo, C. and Kapellos, K.: "Computer-Aided Design of a Generic Robot Controller Handling Reactivity and Real-Time Control Issues", IEEE Trans. on Control Systems Technology, vol. 1, no. 4, pp. 213- 229, Dec. 1993

[Simon 95] Simon, D., Kapellos, K. and Espiau, B.: "Formal Verification of Mission and Tasks Application to Underwater Robotics", pre-print, Workshop on Hybrid Systems, Grenoble, France, Sept. 1995

[Smart 01] "Embedded Software Analysis and Generation", SmartVehicle Baseline Report, Ford Motor Co., 2001

[Smith 00] Smith, S. V., David, L. et al, "Loicz Tools: Quantifying GroundWater Flow ", http://www.coastal-fluxes.slt.lk.

[Sontag 83] Sontag, E.: "A Lyapunov-like Characterization of Asymptotic Controllability", SIAM J. Control and Optimization, vol. 21, pp. 462-471, 1983

[Sousa 94] Sousa, J., Pereira, F. and Silva, E.: "A Dynamically Configurable Architecture for the Control of an AUV", Procs. of the OCEANS 94 Conf., Brest, France, Sept., pp 131-136, 1994

[Sousa 95] Sousa, J. and Pereira, F.: "A generalized vehicle-based control architecture for multiple AUVS ", Procs. of the OCEANS 95, pp 131-136, 1995

[Sousa 97] Sousa. J.: "On the Control of Remus Vehicle", Remus Project Report, Institute of Systems and Robotics, Porto, Portugal, 1997

[Sousa 01] Sousa, J., Fraga, S, Martins, A. and Pereira, F.:"A Control Framework for a Remotely Operated Vehicle", Procs. International Conference on Engineering for Ocean and Offshore Structures, Singapore, Dec. 2001

[Sousa 02] Sousa, J., Matos, A. and Pereira, F.: "Dynamic Optimization in the Coordination and Control of Autonomous Underwater

Vehicles", LSTS internal report submited to CDC, Las Vegas, EUA, 2002

[Stewart 96] Stewart, D. and Khosla, P.: "The Chimera Methodology: Designing Dynamically Reconfigurable and Reusable Real-time Software using Port-Based Objects: Journal of Software Engineering and Knowledge Engineering, pp.249-27, June 1996

[Stewart 96] Stewart, D. and Arora, G.:" A Tool for Analyzing and Fine-Tuning the Real-Time Properties of an Embedded System", submitted to IEEE Trans. On Software Engineering, June 2000

[Swarzenski 02] Swarzenski, Peter W., U.S Geological Survey Center for Coastal Geologie, http;//sofia.usgs.gov.

[Tripakis 96] Tripakis, S. e Yovine, S. . "Analysis of timed systems based on time-abstracting bisimulations", In Proceedings of 8th Computer-Aided Verification, Rutgers, NJ, Lecture Notes in Computer Science 1102, Springer-Verlag, July 1996

[Varaya 72] Varaya, P.: Notes on Optimization, Van Nostrand Reinhold Co., 1972

[Varaya 97] Varaya, P.:"Towards a Layered View of Control", Procs. 36th CDC, pp.1187-1190, 1997

[Varaya 98] Varaya, P.: "Reach Set Computation Using Optimal Control", Procs. KIT Workshop on Verification of Hybrid Systems, Verimag, Grenoble, France, 1998

[Varaya 99] Varaiya, P: "A Question About Hierarchical Systems", pre-print, T. Djaferis and I. Schick (eds), System Theory: modeling, analysis and control, Kluwer, 1999

[Venugopal 92] Venugopal, K, Sudahakar R. and Pandaya A.: "On-line Learning Control of Autonomous Underwater Vehicles Using Feedforward Neural Networks", IEEE Journal of Oceanic Engineering, vol 17, no 4, Oct. 1992

[Vinter 00] Vinter, R.:" Optimal Control", Systems and Control, Birkhauser, 2000. [Wang 90] Wang, F., Saridis G.: "A Coordination Theory for Intelligent Machines", Automatica, vol. 20, no 5, pp. 833-844, 1990

[Wang 91] Wang, F., Kyriakopoulos, K., A. Tsolkas and G. Saridis.: "A Petri-net for an Intelligent Mobile Robot", IEEE Trans. on Systems, Man and Cybernetics, vol. 21, no. 4, pp. 777-789, July/August 1991

[Wolenski 90] Wolenski, P.: "An exponential Formula for the Reachable Set of a Lipschitz Differntial Inclusion", SIAM J. Control and Optimization, vol. 28, pp. 148-1161, 1990

[Wong 96a] Wang, K., C., Wonham, W. M. "Hierarchical Control of Discrete-Event Systems", Discrete Event Dynamic Systems, Kluwer Academic Publishers, pp. 241-273, 1996,

[Wong 96b] Wang, K., C., Wonham, W. M. "Hierarchical Control of Timed Discrete-Event Systems", Discrete Event Dynamic Systems, Kluwer Academic Publishers, pp. 275-306, 1996

[Wong 96c] Wang, K., C., Wonham, W. M. "Modular Control and Coordination of Discrete- Event Systems", internal report, University Toronto, 1996

[Wonham 87] Wonham, W.M e Ramadge, P.J., "On supremable controllable sublanguage of a givem language", in SIAM Journal on Control and Optimization, vol. 25, no 3, pp. 637-659, 1987

[Ye 97] Ye, J.:" Optimal Strategies for Bilevel Dynamic Problems", SIAM J. Control and Optimization, vol. 35, no. 2, pp. 512-531,1997

[Ying 94] Zhang Ying, "A Foundation for the Design and Analysis of Robotic Systems and Behaviors", Thesis for the degree of Doctor of Philosophy, at the University of British Columbia 1994

[Ying 94] Zhang Ying, "A Foundation for the Design and Analysis of Robotic Systems and Behaviors", Thesis for the degree of Doctor of Philosophy at the University of British Columbia 1994

[Yorger 86] Yoerger, D., Newman, J., Slotine: J., "Supervisory Control System for the JASON ROV", IEEE J. Ocean. Engeneering, vol. 11, no. 3, pp. 392-399, July 1986

[Yuh 90] Yuh, J.: "A Neural Net Controller for Underwater Robotic Vehicles", IEEE Journal of Oceanic Engineering, vol 15, no 3, July 1992

[Zeeman 62] Zeeman, E.: "The Topology of the Brain and Visual Perception", Topology of 3-Manifolds, Fort, M. (ed), pp. 240-256, 1962

ANEXO: PRELIMINARES ALGÉBRICOS

Estas notas foram organizadas seguindo a proposta de [Wonham 99] com base nas notas das aulas e teses de doutoramento que orientou no período de 1988 a 1999.

esta página foi deixada em branco intencionalmente

A.1 Posets

Os "***Posets***"(conjuntos parcialmente ordenados), desempenham uma função muito importante na teoria dos sistemas, pois estas estruturas são fundamentais ao desenvolvimento e ao suporte de certos conceitos de optimização.

Seja X um conjunto. Uma relação binaria em X é um subconjunto de $X \times X$. Seja o operador $\leq$ uma relação binária em X, e $x, y \in X$, escrever $x \leq y$, significa que o par ordenado $(x, y) \in X \times X$ pertence à relação $\leq$.

A relação $\leq$ é uma p.o (do inglês "partial order", talvez traduzir por relação de ordenação parcial) em X se esta é:

$$\begin{aligned} &\textit{reflexiva}: && (\forall x \in X)\ x \leq x \\ &\textit{transitiva}: && (\forall x, y, z \in X)\ x \leq y \,\&\, y \leq z \Rightarrow x \leq z \\ &\textit{antisimétrica}: && (\forall x, y \in X)\ x \leq y \,\&\, y \leq x \Rightarrow x = y \end{aligned}$$

Dois elementos $x, y \in X$, são comparáveis se $x \leq y$ ou $y \leq x$. Uma ***p.o.*** é totalmente ordenada se os elementos de qualquer par de X são comparáveis. Nem sempre dois elementos arbitrários de X são comparáveis. Se $x, y \in X$ não são comparáveis então escreve-se $x <> y$

Se $\leq$ é uma ***p.o.*** em X, então o par $(X, \leq)$ é um ***poset*** (ou conjunto parcialmente ordenado).

se o "***meet***" de x e y existe denota-se por $x \wedge y$

Seja $x, y \in X$, e X um ***poset***. Um elemento $a \in X$ é um limite inferior para x e y sse $a \leq x$ e $a \leq y$. Um elemento $l \in X$ é o maior limite inferior ("***meet***") para x e y sse

$$\begin{aligned} & l \leq x \,\&\, l \leq y \\ & \quad \&(\forall a \in X)\, a \leq x \,\&\, a \leq y \Rightarrow a \leq l \end{aligned}$$

se o "***join***" de x e y existe denota-se por $x \vee y$

Da mesma forma, $b \in X$ é um limite superior para x e y sse $x \leq b$ e $y \leq b$. Um elemento $u \in X$ é o menor limite superior para ("***join***") x e y sse

$$\begin{aligned} & x \leq u \,\&\, y \leq u \\ & \quad \&(\forall b \in X)\, x \leq b \,\&\, y \leq b \Rightarrow u \leq b \end{aligned}$$

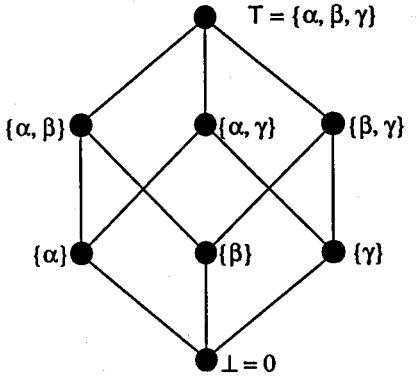
Considera-se que $\perp \in X$ é o elemento "***bottom***" (caso exista), $\perp \leq x\ \forall x \in X$. De forma semelhante considera-se $T \in X$ o elemento de "***top***", $T \geq x\ \forall x \in X$.

A.2 Diagramas de Hasse

Um diagrama de Hasse de um ***poset*** $(X, \leq)$, é um grafo directo cujos os nós correspondem aos elementos de $x \in X$, e os lados aos pares (x, y). Os lados são normalmente desenhados para a vizinhança de x e y

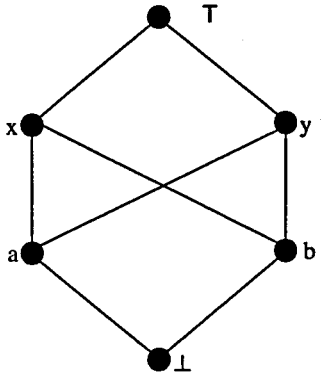
Exemplo

Seja $A = \{\alpha, \beta, \gamma\}$ e $X = Pwr(A)$, o diagrama de Hasse é o seguinte.



Consideremos outro exemplo

$$X = \{T, x, y, a, b, \bot\}$$



A.3 Lattices

Um ***Lattice*** é um ***poset*** L,em que quaisquer dois elementos possuem "***meet***" e "***join***". Mais formalmente as operações $\wedge$ e $\vee$ definem as seguintes funções:

$\wedge : LxL \rightarrow L;$
$\vee : LxL \rightarrow L.$

Se $x, y, z \in L$, e considerando que $*$ denota consistentemente os operadores $\wedge$ e $\vee$ então:

$x * x = x$ (o operador * é idempotente)

$x * y = y * x$ (o operador * é comutativo)

$(x * y) * z = x * (y * z)$ (o operador * é associativo)

Então para qualquer k podemos escrever, sem ambiguidade, e nomeadamente "***meet***" e "***join***" estão definidos para qualquer subconjunto finito de elementos.

Podem ainda ser verificadas as seguintes relações:

$x \wedge (x \vee y) = x \vee (x \wedge y) = x$ (absorção);

$x \leq y \; sse \; x \wedge y = x \; e \; sse \; x \vee y = y$ (consistência).

$y \leq z \Rightarrow (x \wedge y \leq x \wedge z) \& (x \vee y \leq x \vee z)$ ($\wedge$ e $\vee$ são "isotone")

$x \wedge (y \vee z) \geq (x \wedge y) \vee (x \wedge z)$
$x \vee (y \wedge z) \geq (x \vee y) \wedge (x \vee z)$ (desigualdades distributivas)

$x \leq z \Rightarrow x \vee (y \wedge z) \leq (x \vee y \wedge z)$ (desigualdade modular)

Se num dado ***lattice*** as desigualdades distributivas são sempre igualdades, então o ***lattice*** é distributivo. Se a expressão do lado direito da desigualdade modular é sempre uma igualdade, então o ***lattice*** é modular. Todo o ***lattice*** distributivo é modular.

Seja $(L, \wedge, \vee)$ um ***lattice*** e S um subconjunto não vazio (possivelmente infinito) de L. Para generalizar a noção de "***meet***" de elementos de S, definamos $l = \inf(S)$, significa que l é um elemento de L, com as seguintes propriedades:

$(\forall y \in S), l \leq y \quad \& \quad (\forall z)(\forall y \in S) \; z \leq y) \Rightarrow z \leq l$ (não é requisito l pertencer a S).

De forma equivalente é possível generalizar a definição de "***join***", considerando $u = \sup(S)$. Se S é finito, então $\inf(S)$ e $\sup(S)$ são o "***meet***" e o "***join***" como foi definido anteriormente paro um L finito, e existem sempre, porque L é um ***lattice***. Mas caso S seja infinito não é verdade que $\inf(S)$ e $\sup(S)$ existam. O ***laticce*** L é ***completo*** se para qualquer subconjunto S não vazio de L, os $\inf(S)$ e $\sup(S)$ existem.

Quer L seja ou não seja completo, se o $\sup(L)$ (ou $\mathbf{T}$) existe, então por convenção define-se para o subconjunto vazio $S=0\subseteq L$, $\inf(0)=\sup(L)$. De forma semelhante, se $\inf(L)$ (ou $\perp$) existe, então podemos definir $\sup(0)=\inf(L)$.

A.4 Relações de Equivalência

Seja X, um conjunto não vazio e $E\subseteq XxX$ uma relação binária em X. E é uma relação de equivalência se:

$$\begin{array}{ll}(\forall x\in X)\ xEx & E\ é\ reflexiva\\ (\forall x,x'\in X)\ xEx'\Rightarrow x'Ex & E\ é\ simétrica\\ (\forall x,x',x''\in X)\ xEx'\ \&\ x'Ex''\Rightarrow xEx'' & E\ é\ transitiva\end{array}$$

Muitas das vezes denota-se xEx' por $x\equiv x'$

Considere-se $x\in X$, $[x]$ é o subconjunto de elementos x' que são equivalentes a x:

$$[x]:=\{x'\in X\mid x'Ex\}\subseteq X$$

O subconjunto $[x]$ é um ***coset*** (ou classe de equivalência) de x referida à relação de equivalência E. Pela propriedade reflexiva todos os ***cosets*** são não vazios.

Proposição 1.

Quaisquer dois ***cosets*** ou são coincidentes, ou são disjuntos.

$(\forall x,y\in X)\ ou\ [x]=[y]\ ou\ [x]\cap[y]=0$

Prova

Seja $x,y\in X$ e $u\in[x]\cap[y]$, pelo que uEx e uEy. Consideremos $[x]\subseteq[y]$. Se $x'\in[x]$ então $x'Ex$. Dado que xEu (pela simetria) temos que $x'Eu$ (por transitividade) e da mesma forma temos $x'Ey$. Resultado equivalente seria obtido considerando $[y]\subseteq[x]$, pelo que $[x]=[y]$.

Seja P a família de subconjuntos de X com índice α pertencente a um conjunto A qualquer:

$$\mathrm{P}=\{C_\alpha\mid\alpha\in A\},\ C_\alpha\subseteq X$$

Uma família P é uma partição de X se:

$(\forall \alpha \in A) C_\alpha \neq 0$ *(todos os subconjuntos são não vazios)*

$(\forall x \in X)(\exists \alpha \in A)\, x \in C_\alpha$ *(cada x pertence a um C_ε)*

$(\forall \alpha, \beta \in A)\alpha \neq \beta \Rightarrow C_\alpha \cap C_\beta = 0$ *(subconjuntos com diferentes índices são disjuntos)*

Aos subconjuntos C_α chama-se células de P.

desta observação infere-se o termos *"relação de equivalência"* e *partição* têm significados equivalentes.

Então, pode afirmar-se que para uma dada classe de equivalência, o conjunto de diferentes ***cosets*** $[x]$ é uma partição de X. Inversamente para uma dada partição P de X, pode-se definir uma classe de equivalência E em X por:

$xEy \; sse \; (\exists \alpha \in A) x \in C_\alpha \;\&\; y \in C_\alpha$;

ou seja x e y são equivalentes se pertencem à mesma célula de P.

Seja $\Omega(X)$, ou simplesmente Ω, o conjunto todas as relações de equivalência (ou partições) em X. Definimos uma ***p.o.*** em Ω de forma a que Ω seja um ***lattice*** completo do seguinte modo:

$(\forall E_1, E_2 \in \Omega) E_1 \leq E_2 \; sse \; (\forall x, y \in X) x E_1 y \Rightarrow x E_2 y$

Foram adoptados os termos fino e grosseiro para os termos em inglês finer e coarser respectivamente.

Significa que todo e qualquer ***coset*** E_1 é um subconjunto (e só de um) ***coset*** de E_2. Se $E_1 \leq E_2$ então pode dizer-se que E_1 refina E_2, ou E_1 é mais fino que E_2, ou ainda que E_2 é mais grosseiro que E_1.

Proposição 2.

No ***poset*** $(\Omega, \leq)$, o ***meet*** $E_1 \wedge E_2$ dos elementos E_1 e E_2 existe sempre e é dado por

$$(\forall x, x' \in X)\, x \equiv x', (\mathrm{mod}\, E_1 \wedge E_2)$$
$$sse \; x \equiv x' (\mathrm{mod}\, E_1) \;\&\; x \equiv x' (\mathrm{mod}\, E_2)$$

Prova

Considere-se $E := E_1 \wedge E_2$ como definimos (no enunciado da proposição). Então E é uma classe de equivalência em X, pelo que $E \in \Omega$. Demonstra-se que $E \leq E_1$ e que $E \leq E_2$. Por fim se $F \in \Omega$ e $F \leq E_1, F \leq E_2$ então $F \leq E$.

Pode então dizer-se que o "***meet***" é a partição mais grosseira que é mais fina do que E_1 e E_2,

Proposição 3.

No ***poset*** $(\Omega,\leq)$, o ***join*** $E_1 \vee E_2$ dos elementos E_1 e E_2 existe sempre e é dado por

$$(\forall x, x' \in X)\, x \equiv x', (\mathrm{mod}\, E_1 \vee E_2)$$
$$sse\, (\exists k_{\mathrm{int}eiro} \geq 1)(\exists x_0, x_1, x_2 \ldots\ldots x_k \in X) x_0 = x \,\&\, x_k = x'$$
$$\&\, (\forall i) 1 \leq i \leq k \Rightarrow [x_i \equiv x_{i-1} (\mathrm{mod}\, E_1)\, or\, x_i \equiv x_{i-1} (\mathrm{mod}\, E_2)]$$

Pela proposição de ***join*** observamos que x e x' podem ser conectados por uma sequência de elementos auxiliares $x_1,\ldots,x_{k-1}$, em que cada "conexão" representa uma outra das classes de equivalência. O ***join*** $E_1 \vee E_2$ pode ser então descrito como a mais fina partição, das partições mais grosseiras que contêm quer E_1 quer E_2.

Desta forma é estabelecido um ***lattice*** $(\Omega,\leq)$ de relações de equivalência (ou partições).

Vamos agora verificar que $(\Omega,\leq)$ é um ***lattice*** completo.

Proposição 4.

Seja $F \subseteq \Omega$ uma colecção não vazia de relações de equivalência em X. Então o $\inf(F)$ existe. Vejamos

$$(\forall x, x' \in X)\; x(\inf(F))x' \; sse\, (\forall f \in F)\; xfx'$$

Também o $\sup(F)$ existe, vejamos então que:

$$(\forall x, x' \in X)\, x(\sup(F))x'$$
$$sse\; (\exists k_{\mathrm{int}eger} \geq 1)(\exists F_1,\ldots,F_k \in F)(\exists x_0,\ldots,x_k \in X)$$
$$x_0 = x \;\&\; x_k = x' \;\&\; (\forall i) 1 \leq i \leq k \Rightarrow x_i \equiv x_{i-1}\, (\mathrm{mod}\, F_i)$$

Tal como foi definido o $\inf(F)$ e o $\sup(F)$ também são relações de equivalência e como tal possuem as mesmas propriedades requeridas pelas definições de supremo e ínfimo já apresentadas anteriormente.

A definição de $\inf(F)$ afirma que x e x' são equivalentes respectivamente a cada uma das relações de equivalência da colecção F. A definição de $\sup(F)$ afirma que x e x' podem ser ligados por uma sequência elementos através de sucessivas relações de equivalência $(\mathrm{mod} f)$ $f \in F$.

Particularmente $\perp = \inf(\Omega)$ e $T = \sup(\Omega)$ existem e são dados por:

$x \equiv x' (\mathrm{mod}\perp)\ sse\ x = x', x \equiv x' (\mathrm{mod} T)\ sse\ VERDADE$

O $\perp$ é a mais fina partição possível de encontrar, enquanto T é a maior partição possível de encontrar (ou seja todo o conjunto X.

Pelo que vimos anteriormente então o ***lattice*** das relações de equivalência em X é completo.

Os elementos de $\Omega(X)$ pode ser interpretados como estruturas de informação em X. Dessa forma, seja $E \in \Omega(X)$ uma relação de equivalência conhecida, e exactamente conhecido o elemento $y \in X$, o significado da afirmação "$x \equiv y(\mathrm{mod}\, E)$" pode ser traduzido por "x é conhecido relativamente à propriedade da partição de y". Consequentemente uma partição fornece mais informação que outra mais "grosseira". O elemento $\perp \in \Omega$ significa informação "perfeita" (ou ignorância nula), enquanto T significa "informação nula" (ou ignorância completa).

Se $E, F\, \Omega(X)$ então $E \wedge F$ pode ser interpretado como representando a informação que é contributo da cooperação de E e F. Dualmente $E \vee F$ pode ser interpretado como a informação que E e F produzem em comum.

A.5 Núcleo de Equivalência e Factorização Canónica

Seja X e Y conjuntos. Neste ponto denotaremos os elementos de $\Omega(X)$ por π, ρ etc, e vamos pensar nos elementos de $\Omega(X)$ como partições de X. Seja $\pi \in \Omega(X)$ e denotaremos por $\overline{X}$ o conjuntos das células distintas de π, ou alternativamente $\overline{X}$ denota o conjunto dos índices das células da partição π. Frequentemente também se escreve X/π (lê-se $X \bmod \pi$) para representar $\overline{X}$. Complementarmente denotaremos por P_π a função "surjectiva" que mapeia qualquer x no respectivo ***coset***:

$P_\pi : X \to X/\pi : \ x \to [x].$

Chamaremos a P_π projecção canónica associada a π.

Consideremos $f: X \to Y$. Frequentemente associa-se a f, uma ***função induzida*** $f_*: Pwr(X) \to Pwr(Y)$, que para cada subconjunto em X o transforma numa imagem em Y por f:

$$f_*(A) := \{f(x) | x \in A\}, \quad A \in Pwr(X), \quad (i.e.\ A \subseteq X).$$

Consequentemente $f_*(A) \subseteq Y$. Particularmente $f_*(X)$ é a imagem de f, denotada por $\mathrm{Im}(f)$. Usualmente na notação não se distingue f e f_*, escrevendo $f(A)$ por $f_*(A)$.

Associa-se à função $f: X \to Y$ a função imagem inversa $f^{-1}: Pwr(Y) \to Pwr(X)$ de acordo com

$$f^{-1}(B) := \{x \in X \mid f(x) \in B\}, \quad B \in Pwr(Y), \quad (i.e.\ B \subseteq Y)$$

Assim $f^{-1}(B) \subseteq X$

Nota:
f^{-1} é sempre uma função "bem definida", mesmo que f não tenha função inversa no sentido comum.
Caso f seja bijectiva, então a função inversa no sentido comum, coincide com a inversa da função imagem f^{-1}.

Associe-se agora a $f: X \to Y$, uma relação de equivalência em $\Omega(X)$ chamada o núcleo de equivalência de f denotada por $\ker f$ definida como se segue:

$$(\forall x, x' \in X)\ x \equiv x' \ (\mathrm{mod}\ \ker f)\ sse\ f(x) = f(x')$$

$\ker P_\pi = \pi$

Os ***cosets*** do $\ker f$ são os subconjuntos de X em que f assume valores distintos em Y, e por vezes chamados de "fibers" de f.

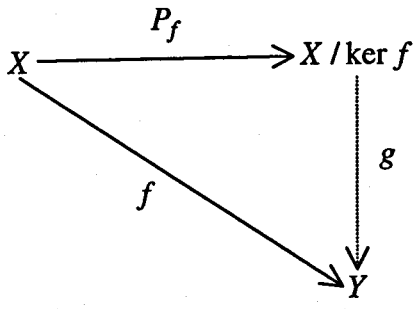
Exemplo: Considere a função $f: S^2 \to R$ que mapeia um ponto da superfície da terra, na respectiva altura relativamente ao nível do mar (medida em metros). Então $f^{-1}(\{100\})$ é o ***coset*** do $\ker f$ que contêm os pontos cuja a elevação são 100 metros.

A partição correspondente ao $\ker f$ consiste na subfamília de subconjuntos não vazios de X formada da família de subconjuntos

$$\{f^{-1}(\{y\}) | y \in Y\}$$

Consideremos novamente $f: X \to Y$, e seja $P_f: X \to X/\ker f$ a projecção canónica. Então existe uma única função $g: X/\ker f \to Y$ tal que $f = g o P_f$.

De facto, seja $z \in X/\ker f$ e $z = P_f(x)$ para algum $x \in X$, define-se $g(z) := f(x)$. Esta definição é inequívoca, porque $z = P_f(x')$ então $P_f(x') = P_f(x)$, e, portanto uma vez $\ker P_f = \ker f$ então $f(x') = f(x)$. Desta forma obtemos a factorização canónica de f através do núcleo de equivalência.



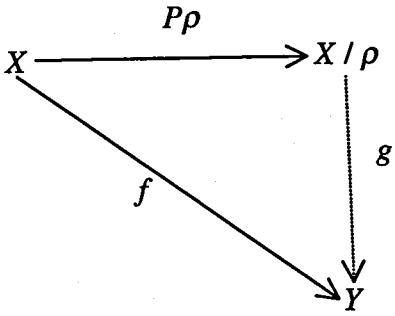
Na factorização canónica $f = go P_f$, o factor g é sempre injectivo. Suponhamos que $z, z' \in X / \ker f$ e $g(z) = g(z')$. Se $z = P_f(x)$ e $z' = P_f('x')$ então $f(x) = go P_f(x) = g(z) = g(z') = go P_f(x') = f(x')$

nomeadamente $x \equiv x' (\text{mod} \ker f)$, então $x \equiv x' (\text{mod} \ker P_f)$, *i.e.*

$$z = P_f(x) = P_f(x') = z'$$

Proposição 5.

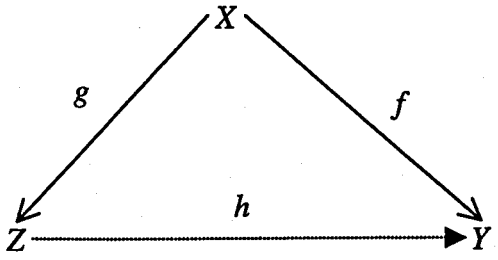
Suponha-se $f : X \to Y$ e seja $\rho \in \Omega(X)$ com $\ker f \geq \rho$. Existe um mapa único $g : X / \rho \to Y$ tal que $f = go P_\rho$.



$\ker f \geq \rho$

Proposição 6.

Suponha-se $f : X \to Y$ e $g : X \to Z$ e seja $\ker f \geq \ker g$. Então existe um mapa $h : Z \to Y$ tal que $f = ho g$. Além disso, h é único definido na imagem $g(X)$ em Z, ou seja a restrição $h \mid g(X)$ é única.

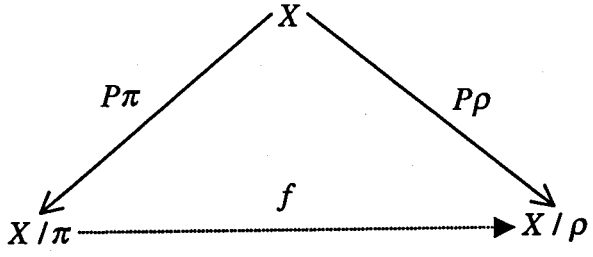


$\ker f \geq \ker g$

Nesta situação f é chamado *factor através de* g. Intuitivamente o significado desta proposição é "g preserva informação suficiente para calcular f através de h.

Proposição 7.

Se $\pi, \rho \in \Omega(X)$ e $\pi \leq \rho$, então existe uma única função $f: X/\pi \to X/\rho$ tal que $P_\rho = foP_\pi$.



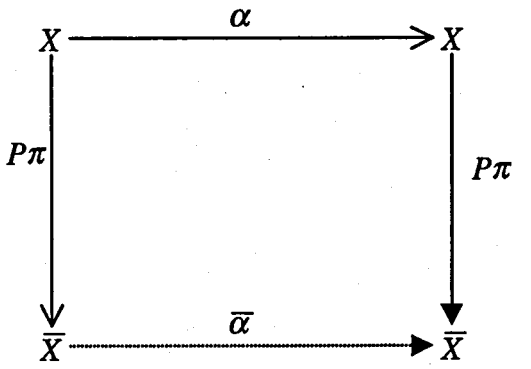
$\rho \geq \pi$

Exemplo: Congruência de um sistema dinâmico

Um sistema dinâmico num conjunto X é um mapa $\alpha: X \to X$, com seguinte interpretação. Os elementos $x \in X$ são os "estados" do sistema, e α é a função de transição de estado. Escolhido um estado inicial $x_0 \in X$, deixando o sistema evoluir sucessivamente pelos estados $x_1 = \alpha(x_0)$, $x_2 = \alpha(x_1)$, ... A sequência $\{\alpha^k(x_0) \mid k \in N\} \in X^N$ é o caminho de (X, α) com estado inicial x_0.

escreve-se α^k para a k-ésima composição de α ($\alpha^0 = identidade$, $\alpha^1 = \alpha$)

Seja $\pi \in \Omega(X)$, com a projecção canónica $P_\pi: X \to \overline{X} := X/\pi$. Diz-se que a relação de equivalência π é congruente com α se existe um mapa $\overline{\alpha}: \overline{X} \to \overline{X}$ tal que $\overline{\alpha} o P_\pi = P_\pi o \alpha$.



Proposição 8.

π é congruente relativamente a α sse

$$\ker P_\pi \leq \ker(P_\pi o \alpha),$$

nomeadamente

$$(\forall x, x')(x, x') \in \pi \Rightarrow (\alpha(x), \alpha(x')) \in \pi$$

Prova

pela proposição 1, fazendo as seguintes atribuições $(Y, f, \rho, g) = (\overline{X}, P_\pi o \alpha, \pi, \overline{\alpha})$.

A.6 "Sprays" e "Covers"

Veremos como a factorização canónica de uma função através do núcleo de equivalência pode ser generalizado a um conjunto de relações.

Seja X e Y conjuntos. Uma relação entre X e Y é um subconjunto $f \subseteq X \times Y$. Se o par $(x, y) \in f$, diz-se que y é $f-relacionado$ a x. A relação f de X para Y é um "***spray***" se

$$(\forall x \in X)(\exists y \in Y)(x, y) \in f,$$

ou seja cada elemento de X possui pelo menos um elemento de Y $f-relacionado$. Então escreve-se $f : X --| Y$. Um "***spray***" é funcional se para todo o elemento de X existe um e um só elemento de Y tal que $(x, y) \in f$ (neste caso a notação adoptada para a relação deve ser a convencional). Seja $f : X --| Y$, o "***cover***" de f ,denotado por $\text{cov}(f)$, é a família de subconjuntos de X dado por

$$\{f_y \mid y \in Y\}\, f_y := \{x \in X \mid (x, y) \in f\}$$

Os subconjuntos $f_y \subseteq X$ são células do $\mathrm{cov}(f)$. Então para um dado "***spray***" f é verdade que

$$X = \cup\{f_y \mid y \in Y\}$$

Um "***spray***" $f : X --| Y$ é surjectivo se

$(\forall y \in Y)(\exists x \in X)(x, y) \in f$, ou seja, para todo o elemento de Y, $f_y \neq 0$.

Se $f : X --| Y$ e $g : Y --| Z$ então a composição fog é um "***spray***" definido por

$$gof = \{(x, z) \in X \times Z \mid (\exists y \in Y)(x, y) \in f \,\&\, (y, z) \in g)\};$$

Ou seja, para todo o $z \in Z$

$$(gof)_z = \{x \in X \mid (\exists y \in Y) x \in f_y \,\&\, y \in g_z)\}$$

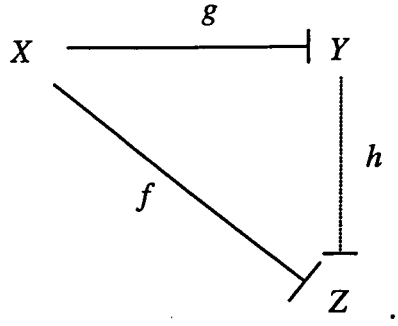
Pela proposição seguinte pode ser observado como um "***spray***" pode ser factorizado através de outro.

Proposição 9.

Sejam $f : X --| Z$ e $g : X --| Y$ "***sprays***". Existe um "***spray***" $h : Y --| Z$ tal que $hog = f$ se:

1. $(\forall y \in Y)(\exists z \in Z) g_y \subseteq f_z$
2. $(\forall z \in Z)(\exists Y' \subseteq Y) f_z = \cup\{g_y \mid y \in Y'\}$

Consequentemente f é factorizado por g se cada (todos) elemento do $\mathrm{cov}(g)$ está contido num qualquer membro do $\mathrm{cov}(f)$, e cada (todos) o elemento do $\mathrm{cov}(f)$ é a união de alguns elementos do $\mathrm{cov}(g)$



Prova

Suponhamos que (1) e (2) é verdade. defina-se $h: Y --| Z$ pela regra: dado $y \in Y$, seja $(y,z) \in h$ *sse* $g_y \subseteq f_z$; pelo menos um z existe por (1), e consequentemente h é um "***spray***". No sentido de verificar $f = hog$, suponha-se que $(x,z) \in f$, e nomeadamente $x \in f_z$. Por (2) existe um $W \subseteq Y$ tal que:

$$f_z = \cup\{g_w \mid w \in W\}$$

Visto que $x \in f_z$, existe $w \in W$ em que $x \in g_w$. Uma vez que $g_w \subseteq f_z$, segue então que $(w,z) \in h$. Também $(x,w) \in g$ pelo que consequentemente $(x,z) \in hog$. como é requerido. Para a prova da afirmação inversa, façamos $(x,z) \in hog$, então para algum $y \in Y$ é verdade que $x \in g_y$ e $y \in h_z$. Por definição de h, $g_y \subseteq f_z$, uma vez que $x \in f_z$, momeadamente $(x,z) \in f$ como é requerido.

Notar que a função da condição (1) é garantir que a relação h é um "***spray***".

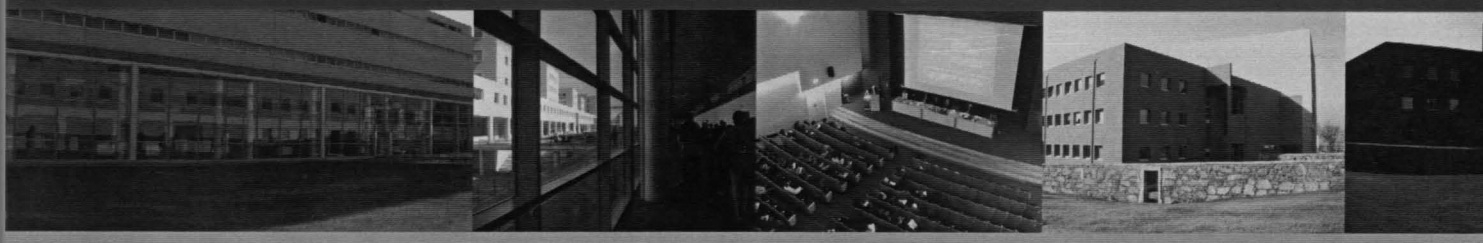
Para os "***sprays***" dados na proposição anterior, na generalidade não é verdade que o "***spray***" h é unicamente determinado pelo requisito $f = hog$, mesmo que g seja "surjectivo". Observemos um exemplo: Seja $X = \{x\}$, $Y = \{y_1, y_2\}$ e $Z = \{z_1, z_2\}$, com $f = \{(x,z_1),(x,z_2)\}$ e $g = \{(x,y_1),(x,y_2)\}$. Então $f = hog = h'og$ onde $h = \{(y_1,z_1),(y_2,z_2)\}$ e $h' = \{(y_1,z_2),(y_2,z_1)\}$. Contudo como subconjuntos de $Y \times Z$, os "***sprays***" $h: Y --| Z$ são parcialmente ordenados, e com respeito a esta ordenação das famílias de "***sprays***" h que satisfazem a condição $f = hog$ possuem um único elemento maximal, dito h^*, definido como na prova da proposição 1:

$$(y,z) \in h^* \; sse \; g_y \subseteq f_z$$

Um "***cover***" de X é uma família C de subconjuntos X (células de C) tal que:

$$X = \cup\{X' \mid X' \in C\}$$

Seja $\mathbf{C}(X)$ a colecção de todos os "***covers***" de X. Se $C \in \mathbf{C}(X)$ seja Y qualquer conjunto em correspondência bijectiva com C, com $y(X')$ o elemento de Y que indexa a célula X' de C. Claramente C define um "***spray***" $f : X --| Y$, com $f_{y(X')} = X'$ e $\mathrm{cov}(f) = C$. Caso não haja células de C vazias o "***spray***", f é "surjectivo".



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000073443