

Departamento de Engenharia Electrotécnica e Computadores
Faculdade de Engenharia da Universidade do Porto
1993

Relatório do Prodep

MÉTODOS DE CONTROLO ÓPTIMO APLICADOS A MANIPULADORES ROBÓTICOS

Jorge Manuel Mendes Silva Gonçalves

4
6213/017.3/LEEC/372/GNNJ
25 09 09

ISR - Instituto de Sistemas e Robótica

DEEC - R. dos Bragas, 4099 Porto Codex, Portugal

Fax: 351 - 2- 319280

Tel: 351 - 2 - 2007505

REFERÊNCIA DO PROJECTO 4.3/1.60/92

Parecer Técnico

Acompanhei os trabalhos de estágio realizados no Pólo do Porto do Instituto de Sistemas e Robótica pelo Licº Jorge Manuel Mendes Silva Gonçalves, bem como apreciei os trabalhos produzidos e apresentados em Congressos Internacionais e o relatório oportunamente apresentado.

Com base no acompanhamento referido, considero que aqueles trabalhos reflectem o trabalho efectivamente realizado, revelando elevado empenhamento do estagiário.

Sou portanto de parecer que foram plenamente atingidos os objectivos inicialmente propostos no plano de trabalhos,

O Supervisor do Estágio no Instituto de Sistemas e Robótica



Prof. Doutor Jorge L. Martins de Carvalho



FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Departamento de Engenharia Electrotécnica e de Computadores

Rua dos Bragas, 4099 Porto Codex, PORTUGAL
Telef. 351-2-317105/107/412/457 · Telex 27323 FEUP P · Telefax 351-2-319280

Acção de Formação Prodep 4.3/7.04/92

Formando: Jorge Manuel Mendes Silva Gonçalves

Orientador Científico: Fernando Manuel Ferreira Lobo Pereira

Este relatório corresponde à apreciação científica das actividades desenvolvidas pelo Licº Jorge Manuel Mendes Silva Gonçalves durante o segundo semestre de 1992.

O trabalho desenvolvido foi de elevado mérito e executado com grande dedicação.

Os trabalhos realizados e o relatório apresentados pelo formando reflectem, com rigor a natureza dos trabalhos executados.

Além do empenhamento e das excelentes qualidades do estagiário concorreram para o sucesso deste trabalho, os meios disponibilizados.

O Supervisor do Estágio na FEUP

Fernando M. F. Lobo Pereira
Prof. Aux. da FEUP

Relatório de Estágio

Estágio para licenciatura no tema

Técnicas de Simulação, Controlo e Optimização para Sistemas Autónomos-
Aplicações na Robótica e em Processos de Manufatura

PRODEP Proc. N° 4.3/7.04/92

Autor

Jorge Manuel Mendes da Silva Gonçalves

Junho de 1993

Índice

1 Actividade desenvolvida.....	3
1.1 Estudos orientados efectuados.....	3
1.2 Disciplinas frequentadas (ver programa de trabalhos)	3
1.3 Participações em Conferências e Publicações em actas ou "proceedings" de congressos internacionais resultantes (comunicação em apêndice).....	3
2. Apêndices	4
2.1 Projecto Realizado.....	4
2.2 Comunicações Apresentadas.	4
2.3 Referências	

1 Actividade desenvolvida

1.1 Estudos orientados efectuados

Neste trabalho é proposta uma metodologia para encontrar estratégias de controlo óptimo para problemas onde se pretende minimizar o tempo de execução.

Serão apresentados dois algoritmos que permitem realizar a optimização do tempo que, os manipuladores demoram na execução das suas tarefas; complementados com um estudo sobre a sua convergência. Por fim aplicaram-se as conclusões obtidas a um exemplo concreto de optimização, sendo de realçar os bons resultados conseguidos.

1.2 Disciplinas frequentadas (ver programa de trabalhos)

Controlo Óptimo
Identificação e Modelização de Sistemas
Métodos de Optimização
Robótica
Análise de Sistemas Informáticos
Planeamento e Gestão da Produção

1.3 Participações em Conferências e Publicações em actas ou "proceedings" de congressos internacionais resultantes (comunicação em apêndice).

"Algoritmos de Tempo Mínimo para Problemas de Controlo em Robótica" apresentado na ENDIEL93. Uma comunicação sobre este trabalho foi também apresentada nas 3^{as} Jornadas Hispano-Lusas em Barcelona.

2. Apêndices

2.1 Projecto Realizado

2.2 Comunicações Apresentadas.

Departamento de Engenharia Electrotécnica e Computadores
Faculdade de Engenharia da Universidade do Porto
1993

Trabalho Realizado

MÉTODOS DE CONTROLO ÓPTIMO APLICADOS A MANIPULADORES ROBÓTICOS

Jorge Manuel Mendes Silva Gonçalves

Índice

1. INTRODUÇÃO	4
2. PROBLEMA DE CONTROLO ÓPTIMO	5
2.1 Descrição do problema	5
2.2 Princípio do máximo	5
3 ALGORITMO DE RESOLUÇÃO	9
3.1 Introdução	9
3.2 Descrição do Algoritmo	9
3.3 Calculo da Estratégia Inicial	10
3.4 Integração das e.d.o. Que Definem a Variável Adjunta	11
3.5 Teste de optimalidade	12
3.5.a Caso com restrições no controlo	12
3.5.b Caso sem restrições no controlo	12
3.6 Actualização do controlo e contador de iterações	12
3.6.a Método do gradiente conjugado	12
3.6.b Caso sem restrições no controlo	13
4. MANIPULADORES ROBÓTICOS	17
4.1 Introdução	17
4.2 Formulação matemática do problema	17
4.3 Forma canónica do problema de controlo óptimo	18
4.4 Princípio do máximo	19
4.5 Algoritmo de resolução	21
4.6 Exemplo. Manipulador com dois graus de liberdade	21
4.6.A Inicialização e estratégia inicial	24
5. TRANSFORMAÇÃO DO PROBLEMA DE TEMPO LIVRE NUM	
PROBLEMA DE TEMPO FIXO	29
5.1 Introdução	29
5.2 Mudança de variável	29
5.3 Princípio do Máximo	30
5.4 Algoritmo de resolução	30
5.5 Exemplo de aplicação	32

CONTROLO EM TEMPO MÍNIMO APLICAÇÕES À ROBÓTICA

1. INTRODUÇÃO

Ultimamente tem sido dispendida uma quantidade considerável de trabalho no desenvolvimento de técnicas numéricas para a solução de problemas de controlo óptimo.

As técnicas de optimização vem de encontro às necessidades da indústria moderna, que devido à grande competitividade actual, se vê obrigada a fazer elevados investimentos de teor tecnológico (por exemplo células flexíveis, manipuladores robóticos e outros sistemas de manuseamento de materiais, etc), tornando imperiosa a sua rentabilização que só é possível com uma utilização optimizada. Esta é especificada através de estratégias de gestão e controlo de sistemas dinâmicos (acontecimentos discretos ou contínuos) que melhor modelizem os sistemas referidos.

Os métodos de trabalho utilizados neste projecto são os de optimização dinâmica que, pelo facto de serem abstractos, endereçam classes muito vastas de problemas identificadas apenas pelas características das estruturas matemáticas presentes quer na função objectivo quer nas equações ou inclusões que definem as restrições dinâmicas e estáticas.

Para além destes exemplos tem sido reconhecido que os sistemas robóticos estão a começar a desempenhar um papel fundamental em várias aplicações, que vão desde a manufactura até à exploração do espaço. Para aumentar a produtividade geral e a eficiência da operação do sistema, um dos aspectos importantes consiste em minimizar o tempo total em que o manipulador robótico está em movimento.

Uma vantagem importante da utilização de manipuladores numa linha de produção de um sistema de manufactura consiste no facto de eles possuírem a flexibilidade necessária para operarem mesmo quando os locais para onde os componentes ou materiais se deslocam dependerem do resultado de operações anteriores (de inspecção por exemplo) que, embora conhecidos à priori, poderão variar por períodos com baixa frequência.

Tendo isto em consideração foi incluída no projecto aqui descrito uma demonstração de como utilizar o algoritmo de tempo mínimo descrito para encontrar estratégias de controlo óptimo de um manipulador robótico a desempenhar tarefas repetitivas numa linha de produção de um sistema de manufactura.

Convém para além disso referir que a estratégia de controlo do manipulador deve ser obtida como solução de um problema de tempo mínimo sujeito não só às restrições dinâmicas, bem como a restrições estáticas associadas ao espaço livre para movimentação e limites máximos de velocidade. A trajectória assim obtida constituirá a referência para o controlador robótico estável e robusto (a perturbações e incertezas).

Sendo assim, a adopção de estratégias de controlo em tempo mínimo, para o sistema de manuseamento e transporte de um processo produtivo traduz-se numa diminuição do tempo de permanência dos produtos semiacabados no processo produtivo. Ora isto revela-se num menor WIP (Work in process) médio o que tem consequências não só no abaixamento dos custos de operação, mas também uma maior rapidez e flexibilidade de resposta às solicitações do mercado.

2. PROBLEMA DE CONTROLO ÓPTIMO

2.1 Descrição do problema

Seja o seguinte sistema que em cada instante t pode ser descrito por $x(t) \in \mathfrak{R}^n$ e controlado por $u(t) \in \Omega$ ($\Omega(t) \in \mathfrak{R}^m$ é o conjunto dos valores possíveis de controlo a aplicar em cada instante) e seja $f(t, x(t), u(t))$ com $f(\cdot, \cdot, \cdot): \mathfrak{R} \times \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$ a sua dinâmica. Sabendo que o estado actual é x_0 , pretendemos levar o sistema deste estado ate ao estado x_f em tempo mínimo.

Assim, o problema pode escrito da seguinte forma

$$(P_1) \quad \begin{array}{ll} \text{Minimizar} & t_f \\ \text{Sujeito a:} & \begin{cases} \dot{x}(t) = f(t, x(t), u(t)) \\ x(0) = x_0 \\ x(t_f) = x_f \\ u(t) \in \Omega(t) \end{cases} \end{array}$$

Por forma a retirar uma restrição ao problema, este pode escrito de uma outra forma

$$(P) \quad \begin{array}{ll} \text{Minimizar} & t_f + K \cdot \|x(t_f) - x_f\|^2 \\ \text{Sujeito a:} & \begin{cases} \dot{x}(t) = f(t, x(t), u(t)) \\ x(0) = x_0 \\ u(t) \in \Omega(t) \end{cases} \quad \text{q.s. em } [0, 1] \end{array}$$

sendo K uma constante positiva escolhida suficientemente grande de modo a penalizar o erro com que se atinge o estado final pretendido.

2.2 Princípio do máximo

O princípio do máximo para o problema de tempo mínimo pode ser escrito da seguinte forma

Seja $(t_f, x(t), u(t))$ um processo minimizante.

Então $\exists p: [0, t_f] \rightarrow \mathfrak{R}^n$ tal que

$$\begin{aligned} -\dot{p}^T(t) &= p^T(t) \cdot f_x(t, x(t), u(t)) \\ p(t_f) &= -g_x(x(t_f)) = -2K \cdot (x(t_f) - x_f) \\ \text{e } u(t) &\text{ maximiza } u \rightarrow p^T(t) \cdot f(t, x(t), u) \quad \text{q.s. em } [0, 1]. \end{aligned}$$

Sendo $H(t, x(t), u(t), p(t))$ definido por

$$H(t, x(t), u(t), p(t)) = p^T(t) \cdot f(t, x(t), u(t))$$

então

$$H(t_f, x(t_f), u(t_f), p(t_f)) = 1$$

Demonstração

Temos, então, o seguinte problema

$$(P) \quad \text{Min } t_f + g(x(t_f))$$

$$\begin{cases} \dot{x}(t) = f(t, x(t), u(t)) \\ u(t) \in \Omega \\ x(0) = x_0 \end{cases} \quad \text{q.s. } -[0, t_f]$$

Seja (u^*, x^*, t^*) dado e solução de (P); construa-se um Problema Auxiliar de tempo fixo (Paux).

Escolhendo τ como a nova variável de tempo, o estado e o controlo podem ser escritos como

$$\tilde{x}(\tau) = (x \circ t)(\tau)$$

$$\tilde{u}(\tau) = (u \circ t)(\tau)$$

sendo $(f \circ g)$ a composta de f após g .

O novo problema tem o seguinte aspecto

$$(Paux) \quad \text{Min } t(1) + g(\tilde{x}(1))$$

$$\begin{cases} \dot{t}(\tau) = v(\tau) & \text{com } v(\tau) > 0 \\ \dot{\tilde{x}}(\tau) = f(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)) \cdot v(\tau) \\ \tilde{x}(0) = x_0, \quad \tilde{u}(\tau) \in \Omega \\ t(0) = 0 \end{cases}$$

onde $t(\tau)$ é uma nova componente da variável de estado.

Então:

$(\tilde{u}^*, v^*, \tilde{x}^*, t^*)$ é Sol. de (Paux) sse (x^*, u^*, t_f^*) é Sol. de (P)

$$e \quad \begin{cases} \tilde{x}^*(\tau) = (x^* \circ t^*)(\tau) \\ \tilde{u}^*(\tau) = (u^* \circ t^*)(\tau) \\ t^*(1) = t_f^* \\ t^*(0) = 0 \\ \dot{t}^*(\tau) = v^*(\tau) \end{cases}$$

onde $(f \circ g)(t) = f(g(t))$.

Princípio do máximo para (Paux)

Seja $(\tilde{u}^*, v^*, \tilde{x}^*, t^*)$ uma Sol. de (Paux)

então

$\exists (\tilde{p}, \tilde{q}): [0, 1] \rightarrow \mathbb{R}^n \times \mathbb{R}$ tal que

$$\begin{cases} -\dot{\tilde{p}}^T(\tau) = \tilde{p}^T(\tau) \cdot f_x(t^*(\tau), \tilde{x}^*(\tau), \tilde{u}^*(\tau)) \cdot v(\tau) \\ -\dot{\tilde{q}}(\tau) = \tilde{p}^T(\tau) \cdot f_t(t^*(\tau), \tilde{x}^*(\tau), \tilde{u}^*(\tau)) \cdot v(\tau) \end{cases}$$

com

$$\begin{cases} -\tilde{p}(1) = g_x(\tilde{x}^*(1)) \\ -\tilde{q}(1) = 1 \end{cases}$$

e $(\tilde{u}^*(\tau), v^*(\tau))$ maximiza $(\tilde{u}, v) \rightarrow [\tilde{p}^T(\tau) \cdot f(t^*(\tau), \tilde{x}^*(\tau), \tilde{u}) + \tilde{q}(\tau)] \cdot v$ em $\Omega \times \mathbb{R}$.

Seja

$$H(t(\tau), \tilde{x}(\tau), \tilde{p}(\tau), \tilde{q}(\tau), \tilde{u}(\tau), v(\tau)) = \tilde{p}^T(\tau) \cdot f(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)) \cdot v(\tau) + \tilde{q}(\tau) \cdot v(\tau).$$

Se $v^*(\tau) \in \mathbb{R}^+$, então $v^*(\tau)$ maximiza $\bar{H}(\tilde{u}(\tau), v)$ o que implica que

$$0 = \frac{d}{dv} \bar{H}(\tilde{u}(\tau), v) = \tilde{p}^T(\tau) \cdot f(t^*(\tau), \tilde{x}^*(\tau), \tilde{u}^*(\tau)) + \tilde{q}(\tau) \quad (1)$$

$$\text{onde } \bar{H}(\tilde{u}(\tau), v) = H(t(\tau), \tilde{x}(\tau), \tilde{p}(\tau), \tilde{q}(\tau), \tilde{u}(\tau), v(\tau))$$

Como $v^*(\tau) > 0$, qualquer que seja τ , então, tem-se que

$$\tilde{u}^*(\tau) \text{ maximiza } \tilde{u} \rightarrow [\tilde{p}^T(\tau) \cdot f(t^*(\tau), \tilde{x}^*(\tau), \tilde{u}) + \tilde{q}(\tau)] \cdot v^*(\tau)$$

implica que

$$\tilde{u}^*(\tau) \text{ maximiza } \tilde{u} \rightarrow \tilde{p}^T(\tau) \cdot f(t^*(\tau), \tilde{x}^*(\tau), \tilde{u}) \quad \text{em } \Omega \quad \text{em } [0, 1]$$

Para exprimir as condições necessárias de optimalidade nos dados do problema original vamos efectuar a mudança inversa de variável independente. t tem inversa uma

vez que é injectiva. Esta propriedade pode deduzir-se do facto de $t(\cdot)$ ser estritamente monótona crescente.

$$\dot{t}(\tau) > 0 \Rightarrow t(\cdot) \text{ é monótona} \Rightarrow t(\cdot) \text{ é injectiva}$$

$$\forall \tau \in [0, 1]$$

Logo existe t^{-1} .

Efectuando as seguintes mudanças de variável inversa

$$\begin{cases} p(t) = (\tilde{p} \circ t^{-1})(t) \\ u^*(t) = (\tilde{u} \circ t^{-1})(t) \\ x^*(t) = (\tilde{x} \circ t^{-1})(t) \\ t^*(1) = t_f \end{cases}$$

e tendo em conta que

$$\dot{p}(t(\tau)) = \frac{dp(t(\tau))}{d\tau} = \frac{dp}{dt} \cdot \frac{dt}{d\tau} = \dot{p}(t) \cdot v(\tau)$$

obtemos

$$\begin{cases} -\dot{p}^T(t) = p^T(t) f_x(t, x^*(t), u^*(t)) \\ -p(t_f^*) = g_x(x^*(t_f^*)) \end{cases}$$

Se em (1) fizer $\tau = 1$ fico com

$$1 = -\tilde{q}(1) = \tilde{p}^T(t_f^*) \cdot f(t_f^*, \tilde{x}^*(t_f^*), u^*(t_f^*)) \quad \text{em } \Omega \text{ em } [0, 1]$$

e

$$u^*(t) \text{ maximiza } u \rightarrow p^T(t) \cdot f(t, x^*(t), u) \quad \text{em } \Omega \text{ em } [0, t_f^*]$$

3 ALGORITMO DE RESOLUÇÃO

3.1 Introdução

O algoritmo que aqui apresentamos é constituído por quatro fases: inicialização e obtenção da estratégia inicial, integração das equações diferenciais ordinárias da variável adjunta, teste de optimalidade e obtenção de um novo controlo.

3.2 Descrição do Algoritmo

Comecemos por ver o que é que cada fase do algoritmo faz em seguida estudarmos com detalhe os pormenores com interesse dessas fases.

Fase A

Inicialização

Aqui começa-se por definir:

- o instante inicial t_0 ;
- as condições iniciais $x(t_0)$;
- os valores do estado no instante final $x(t_f)$;
- e o instante final T (suficientemente grande).

Definição da estratégia inicial

Escolhe-se uma estratégia de controlo u_0 admissível, ou seja, $u_0(t) \in \Omega$. Não basta escolher um controlo admissível qualquer. Esse controlo tem que garantir que o sistema tem que ficar "perto" (valor da função objectivo não muito alto) do estado final de modo a que o algoritmo não convirja posteriormente para mínimos locais longe da solução.

Depois de obtido o controlo inicial $u_0(t)$, estamos em condições de poder integrar as equações diferenciais ordinárias (e.d.o.) que definem a dinâmica do sistema ($\dot{x}(t) = f(t, x(t), u(t))$) em $[t_0, T]$ usando as condições iniciais x_0 , obtendo, assim, a trajectória $x(t)$ em $[t_0, T]$.

O valor do instante (t_f) é definido como o valor mais baixo que a função objectivo toma no intervalo $[t_0, T]$.

Fase B. Obtenção da variável adjunta

Nesta fase, sabendo $p(t_f) = -2K \cdot (x(t_f) - x_f)$ estamos em condições de integrar as e.d.o. que definem o vector adjunto ($-\dot{p}^T(t) = p^T(t) f_x(t, x(t), u(t))$) no intervalo $[t_0, T]$.

Fase C. Teste de Optimalidade

Nesta fase verifica-se se o $u(t)$, $x(t)$ e t_f são óptimos, ou seja, se não existe outro $u'(t)$ que dê origem um $x'(t)$ e a um t_f' tal que $J(u) > J(u')$, em que $J(u)$ é a função objectivo ($J(u) = t_f + g(x(t_f))$).

Se $u(t)$, $x(t)$ e t_f são estratégias óptimas, então o algoritmo pára. De outro modo continua para a fase seguinte (fase D).

Fase D. Obtenção de novo controlo

O novo controlo (que leva obrigatoriamente a um custo mais baixo) pode ser determinado da seguinte forma

$$u_{i+1} = u_i + \delta_i \cdot H_u \in \Omega$$

sendo

$$H(t, x(t), p(t), u(t)) = p^T(t) \cdot f(t, x(t), u(t))$$

e, consequentemente,

$$H_u^T(t, x(t), p(t), u(t)) = p^T(t) \cdot f_u(t, x(t), u(t)).$$

Depois de obtido o novo controlo inicial, podemos integrar as equações diferenciais ordinárias (e.d.o.) da dinâmica do sistema ($\dot{x}(t) = f(t, x(t), u(t))$) em $[t_0, T]$ usando as condições iniciais x_0 , e obtendo, assim, a trajectória $x(t)$ em $[t_0, T]$.

O valor do instante (t_f) é, tal como na fase A, definido como o valor mais baixo que a função objectivo toma no intervalo $[t_0, T]$.

Para finalizar esta secção falta apenas referir que o método usado para integrar as equações diferenciais ordinárias referidas foi o método numérico de Runge-Kutta de quarta ordem.

Passamos de seguida a analisar pormenorizadamente cada uma das fases apresentadas.

3.3 Calculo da Estratégia Inicial

Um dos problemas postos a este algoritmo era o de definir uma estratégia inicial. Isto deve-se ao facto de ser incluída uma função de penalização na função objectivo, o que, implica que a estimativa inicial para o controlo a aplicar não poder ser aleatória, mas muito pelo contrario ela tem que garantir que tanto as restrições de controlo como as restrições sobre a forma de uma função de penalização são satisfeitas.

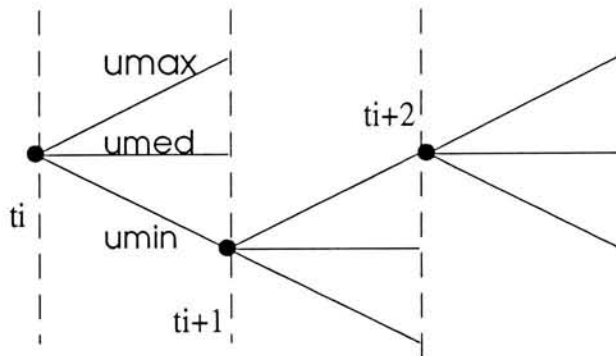
A abordagem que utilizamos foi a utilização de programação dinâmica para determinar esta primeira solução do problema de controlo óptimo. A razão pela qual foi feita esta escolha deve-se ao facto de a programação dinâmica ser um método bastante indicado para tratar processos de decisão multi-estágios pois garante uma solução perto da solução óptima.

Um processo de decisão multi-estágios é um processo que pode ser desdobrado segundo um certo número de etapas sequenciais, ou estágios, os quais podem ser completados de diversas maneiras. As opções para se completarem os estágios são chamadas de decisões. Uma política é um conjunto de decisões - uma decisão para cada estágio do processo.

A programação dinâmica baseia-se no **Princípio de optimalidade de Bellman**. Este princípio diz-nos que uma política ótima apresenta a propriedade segundo a qual, se ela é ótima num estágio então é ótima em todos os estágios anteriores. Estamos a afirmar que uma política é ótima no sentido dela levar à optimização do problema global.

No caso particular do nosso problema o tempo é discretizado e considera-se cada instante de tempo um estágio. Como decisões possíveis temos, saber qual o valor a atribuir ao controlo. Devido ao facto de ser uma variável contínua teríamos infinitas possibilidades, no entanto apenas se consideram no nosso algoritmo três valores possíveis para atribuir ao controlo. O valor máximo, o valor mínimo e um valor intermédio que é a media dos do valor máximo e mínimo. Para cada estágio o valor atribuído ao controlo é o que faz com que o custo resultante dessa escolha seja mínimo. Esta heurística não resolve o caso em que temos múltiplos controlos.

O desenrolar da acção pode ser depreendido da seguinte figura



Em cada instante guarda-se apenas o valor do controlo que minimiza a função custo e determinam-se os valores do estado nesse instante através da integração das equações da dinâmica.

3.4 Integração das e.d.o. Que Definem a Variável Adjunta

Devido ao facto de nós apenas conhecermos o valor de $p(t)$ em t_f ($= -g_x(x(t_f))$), o cálculo de $p(t)$ nos outros instantes $t \in [t_0, T] \setminus \{t_f\}$ é feito mediante a integração da e.d.o. $\dot{p}(t) = -p^T(t) \cdot f_x(t, x(t), u(t))$. Esta, devido ao facto do método geral de Runge-Kutta integrar apenas entre $[t_0, t_f]$ dado x_0 ou x_f , é feita em duas fases que passamos a apresentar:

- a primeira consiste em integrar a equação diferencial de traz para a frente (ou seja, de t_f ate t_0) dado $x(t_f)$.
- posteriormente integra-se desde t_f ate N dado também $x(t_f)$.

De notar que a ordem pela qual estas duas fases são executadas é perfeitamente irrelevante.

3.5 Teste de optimalidade

Para verificar se determinado controlo $u(t)$ é ou não óptimo, existem dois métodos possíveis:

3.5.A CASO COM RESTRIÇÕES NO CONTROLO

Este método vem directamente das condições necessárias de optimalidade e é baseado em considerações teóricas. Neste caso (t_f, x, u) são solução do problema sse verificarem simultaneamente as condições que se seguem:

$$H(t, x(t), u(t), p(t)) \geq H(t, x(t), w, p(t)) \quad \forall w \in \Omega$$

sendo $H(t, x(t), u(t), p(t))$ definido por

$$H(t, x(t), u(t), p(t)) = p^T(t) \cdot f(t, x(t), u(t))$$

Se denotarmos por $\bar{H}(u)$ a função $t \rightarrow H(t, x(t), u(t), p(t))$, então a maximização da condição pode ser expressa na forma

$$\nabla_u \bar{H}(u) \in C_{da}(u)^0$$

onde

$C_{da}(u)^0$ é o polar negativo do cone das direcções admissíveis em U , ou seja é o conjunto $\{d \in L_2([-1, 1], \mathbb{R}^m) : \exists \alpha > 0 \text{ tq } u + \alpha d \in U\}$

3.5.B CASO SEM RESTRIÇÕES NO CONTROLO

O presente método é mais fácil de implementar computacionalmente, no entanto há que ter em atenção que apenas é válido quando não existem restrições no controlo, ou seja quando $\Omega = \mathbb{R}^m$

Convém lembrar que

$$u_{i+1} = u_i + \delta_i \cdot H_u$$

Então, neste caso, o algoritmo pára se $\delta_i^* = 0$, com δ_i definido conforme foi visto na fase D. Isto acontece porque, sendo δ_i^* escolhido de forma a que dado $u_i(t)$ a função objectivo é minimizada. Como H_u é uma direcção de descida (ver próxima secção), $\delta_i^* = 0$ implica que já não posso diminuir mais a função custo o que leva à conclusão de que a estratégia em causa é óptima.

3.6 Actualização do controlo e contador de iterações

3.6.A MÉTODO DO GRADIENTE CONJUGADO

Tendo aplicado as condições necessárias de optimalidade à estimativa da solução obtida $(t_{f_i}, x_i(t), u_i(t))$ e verificando que estas não são satisfeitas, torna-se necessário encontrar um novo controlo $u_{i+1} \in \Omega$ de tal forma que o custo decorrente desta estratégia seja inferior ao da iteração anterior.

Sendo assim o que se pretende é encontrar $s_i > 0$ tal que

$$J(P_U(u_i + s_i \nabla_u \bar{H}(u_i))) < J(u_i)$$

Onde $J(u) = g(x(1))$ sendo x a resposta a u e $P_U(v)$ o ponto de U mais próximo de v , i.e., $\|P_U(v) - v\| \leq \|w - v\| \forall w \in U$

$$\text{Fazer } u_{i+1} = P_U(u_i + s_i \nabla_u \bar{H}(u_i))$$

3.6.B CASO SEM RESTRIÇÕES NO CONTROLO

Nesta secção assume-se $\Omega \equiv \mathcal{R}^m$, e utiliza-se o segundo método de paragem.

Tal como no caso anterior torna-se necessário encontrar um novo controlo $u_{i+1} \in \Omega$ de tal forma que o custo decorrente desta estratégia seja inferior ao da iteração anterior.

Assim, vamos começar por demonstrar que H_u é uma direcção de descida para em seguida apresentarmos uma heurística por nós pensada para a optimização de δ_i .

Teorema: H_u é uma direcção de máxima descida.

Nota: este teorema só é válido quando $\Omega \equiv \mathcal{R}^m$ que é o que estamos a supor.

Demonstração:

Sejam

$$u_{i+1}(t) = u_i(t) + \delta \cdot w$$

e $\bar{x}$ uma perturbação de x .

Se $\dot{\bar{x}} = f(t, \bar{x}, \bar{u})$ e $y = \bar{x} - x$ então

$$\dot{\bar{y}} = f(t, \bar{x}, \bar{u}) - f(t, x, u) \approx f_x(t, x, u) \cdot y + f_u(t, x, u) \cdot \Delta u$$

e $y(0) = \bar{x}(0) - x(0) = y_0$

Sendo assim, podemos escrever $y(t_f)$ como

$$y(t_f) = \phi(t_f, t_0) \cdot y_0 + \int_{t_0}^{t_f} \phi(t_f, t) \cdot f_u(t, x(t), u(t)) \cdot \Delta u \cdot dt$$

Se $\Delta J = g_x(x(1)) \cdot y(1)$ (nota: $g_x(x(1)) = -p(1)$) então ΔJ vem dado por

$$\Delta J = -p^T(t_0) \cdot y(t_0) - \int_{t_0}^{t_f} p^T(t) \cdot f_u(t, x(t), u(t)) \cdot \Delta u \cdot dt \quad (1)$$

sendo

$$p(1) \cdot \phi(t_f, t) = p(t)$$

Se $-p^T(t_0) \cdot y(t_0)$ e Δu forem constantes, se fizermos $w = p^T(t) \cdot f_u(t, x(t), u(t))$, então, quanto maior for w , menor é o valor de ΔJ , ou seja, mais baixa o custo. Sendo assim, é fácil ver que $w = H_u$ é uma direcção de descida e que é também a de máxima descida pois (e segundo (1)) é a direcção para a qual J diminui mais depressa.

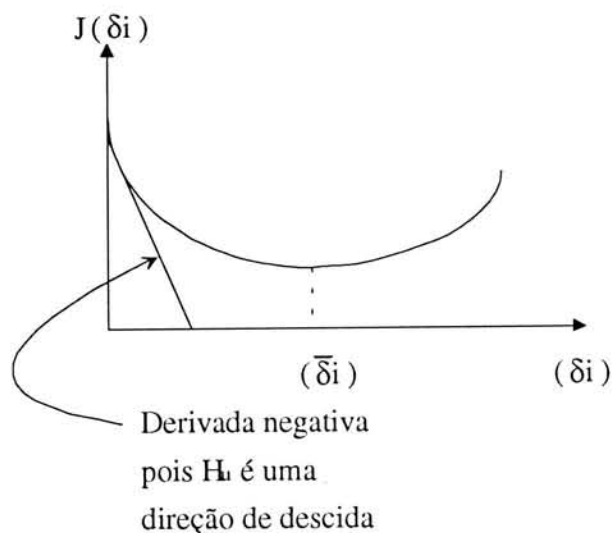
Assim, sabendo que H_u é uma direcção de máxima descida temos

$u_{i+1} = u_i + \delta_i \cdot H_u$, $\delta_i > 0$ (se $\delta_i = 0$ então u_i é óptimo tal como foi explicado na secção anterior)

com

$$H_{u_i}(t, x_i(t), p_i(t), u_i(t)) = p_i^T(t) f_u(t, x_i(t), u_i(t))$$

O problema que se pretende resolver é determinar o valor de $\bar{\delta}_i$ que minimiza o custo da função $J(\delta_i)$. Sendo $J(\delta_i)$ o valor da função objectivo para uma determinada solução $(t_{f_i}, x_i(t), u_i(t), \delta_i)$ temos que o seu gráfico tem a seguinte forma numa vizinhança da origem.



Para resolver este problema de minimização da função $J(\delta_i)$ foi criada uma heurística com os objectivos de chegar a um valor de δ_i muito próximo de $\bar{\delta}_i$ e para além disso não obter um peso computacional muito elevado (a heurística realiza em média 5 iterações). Não é preocupante encontrar o valor exacto de $\bar{\delta}_i$ visto que este valor, apesar de minimizar $J(\delta_i)$ pode levar a que o problema entre em "Zig-Zag" e convirja, deste modo, muito lentamente.

Descrição da heurística

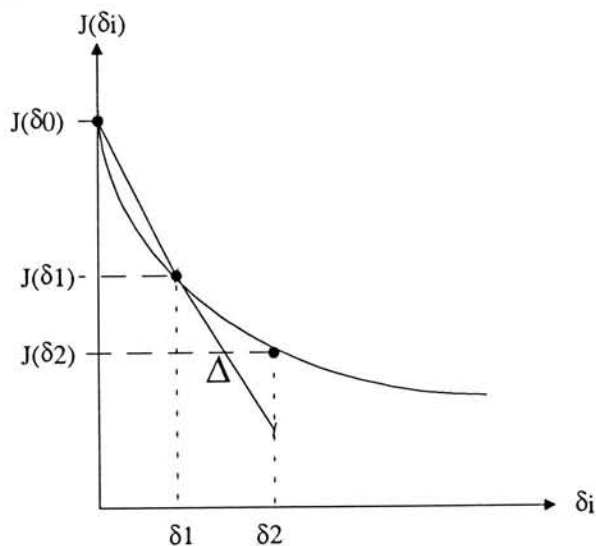
À partida conhecemos o valor de $J(\delta)|_{\delta=0} = J(0)$ que é o custo referente à iteração anterior.

1º passo: Atribuir um valor qualquer para δ_1 .

2º passo: Calcular o custo associado a δ_1 ou seja $J(\delta_1)$. De notar que cada vez que pretendamos calcular $J(\delta)$ é necessário calcular $u = (u_{ant} + \delta H_u)$ em $[t_0, T]$ e integrar as equações da dinâmica para obter $x(t)$ e t_f .

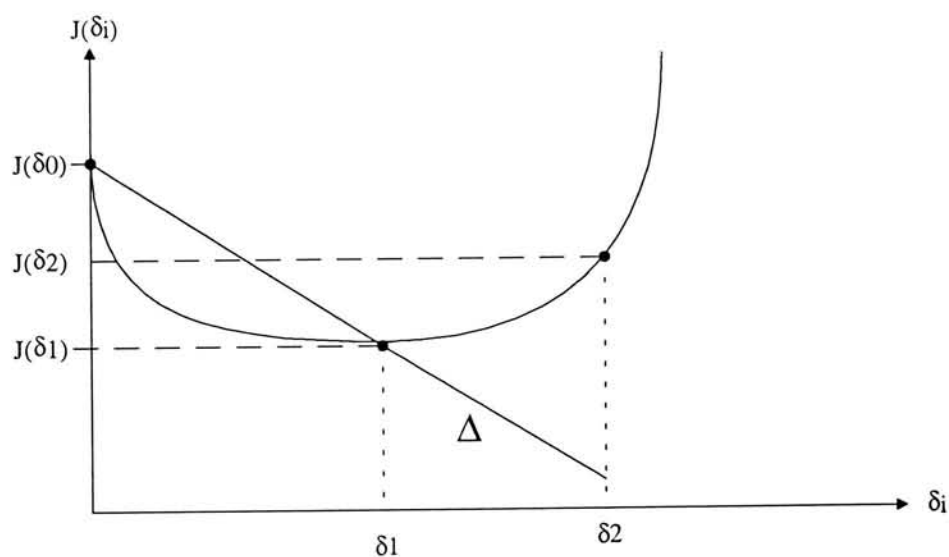
3º passo: Conhecendo os valores de $J(0)$, δ e $J(\delta)$ e supondo $J(\delta_1) < J(0)$, define-se assim uma recta que passa por $(0, J(0))$ e $(\delta_1, J(\delta_1))$ e que se prolonga de Δ para além deste último ponto.

De notar que Δ é definido como na figura seguinte e tem um valor pequeno de modo a encontrar exactamente o primeiro mínimo e representa a distância entre os dois pontos $(\delta_1, J(\delta_1))$ e $(\delta_2, J(\delta_2))$, permitindo encontrar desta maneira o novo passo δ_2 . Este processo pode ser mais facilmente compreendido através da observação da seguinte figura.

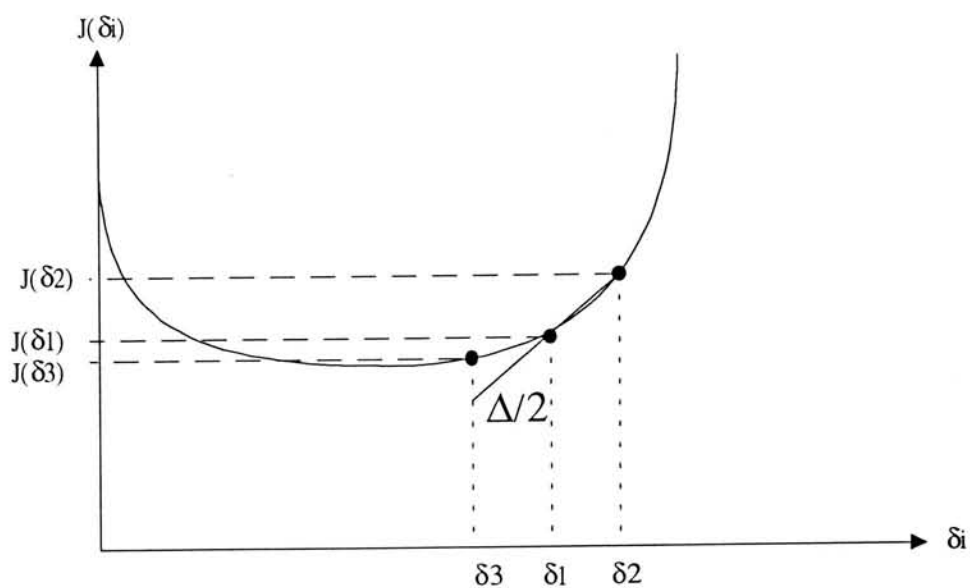


4º. Se $J(\delta_1) > J(\delta_2)$ o algoritmo volta a executar novamente o passo 3, agora com os pontos $(\delta_1, J(\delta_1))$ e $(\delta_2, J(\delta_2))$.

Se $J(\delta_2) < J(\delta_1)$ estamos perante a seguinte situação



Neste caso o que se fazemos $\Delta = \Delta / 2$ e o novo δ_i é determinado prolongando agora no sentido inverso a recta que une os pontos anteriores, ou seja



5º passo: O algoritmo continua até que $\delta_{i+1} - \delta_i \approx 0$ saltando para o passo 2.

4. MANIPULADORES ROBÓTICOS

4.1 Introdução

Nesta secção, vamos analisar um caso particular de grande aplicação deste tipo de algoritmos: minimizar o tempo de que leva um determinado manipulador robótico a mover-se de qualquer posição e velocidade iniciais ate uma outra posição e velocidade desejadas.

A optimização destas trajectórias justifica-se, por exemplo, na manufactura, devido ao facto de os manipuladores robóticos desempenharem tarefas repetitivas e, ao minimizar o tempo despendido pelo manipulador numa determinada operação, estou a minimizar também o tempo global de produção aumentando, assim, a produtividade.

Nesta secção vamos, então, começar por formular matematicamente o problema para, em seguida, o colocarmos na forma canónica de controlo óptimo. Numa terceira parte vamos formular o princípio do máximo para este problema em particular. A seguir, apresentamos o algoritmo e resolução focando as principais diferenças deste algoritmo em relação ao apresentado na secção 4.3. Para finalizar, daremos o exemplo de um manipulador robótico com dois graus de liberdade.

4.2 Formulação matemática do problema

A dinâmica de um manipulador pode ser obtida a partir da Lagrangeana do sistema:

$$L = K - P$$

onde K é a energia cinética e P a energia potencial.

Através do clássico método de Lagrange-Euler tem-se que o binário T pode ser dado por

$$T = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q}$$

onde, para n graus de liberdade, q representa as n posições no espaço das articulações do manipulador. Para obter a dinâmica podemos escrever as equações na forma

$$T = M(q, \dot{q}) \cdot \ddot{q} + N(q, \dot{q})$$

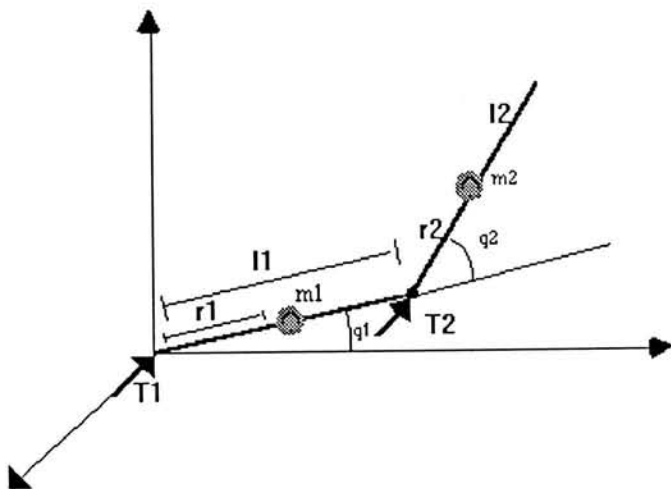
onde

$$N(q, \dot{q}) = V_m(q, \dot{q}) \cdot \dot{q} + G(q) + F_d \cdot \dot{q} + F_s(\dot{q})$$

sendo

- T um vector com n componentes que descreve o binário a aplicar pelos motores do robot ao manipulador
- $M(q, \dot{q})$ uma matriz de $n \times n$ (finita, simétrica e positiva definida) que contém os termos inerciais do manipulador
- $V_m(q, \dot{q})$ uma matriz de $n \times n$ contendo os termos centrípetos e de Coriolis
- $G(q)$ um vector com n elementos contendo os termos gravitacionais

- e F_d e $F_s(\dot{q})$ são termos que vão representar o atrito dinâmico e estático respectivamente (F_d é matriz $n \times n$ e $F_s(\dot{q})$ é um vector de dimensão n).
 Algumas das variáveis descritas atrás, podem ser visualizadas na seguinte figura



As equações descritas, em conjunto com restrições do espaço físico ao vector q (obstáculos, alcance físico do manipulador, etc.), irão determinar um conjunto de trajectórias possíveis que irão servir de base ao nosso trabalho.

Para cada operação que pretendamos fazer com o manipulador, existe um elevado grau de liberdade na trajectória a escolher. Geralmente dado um ponto de partida e sua velocidade (estado inicial), pretende-se atingir um determinado estado final (posição e velocidade desejada) acrescentando algumas restrições estáticas tais como: espaço livre.

A metodologia aqui apresentada pretende encontrar a melhor trajectória, obedecendo às restrições impostas. O critério de optimalidade consiste em minimizar a duração do trajecto do manipulador entre dois pontos dados.

4.3 Forma canónica do problema de controlo óptimo

A partir das equação da dinâmica se fizermos a seguinte mudança de variável

$$\begin{aligned} x &= q \\ y &= \dot{q} \end{aligned}$$

obtemos o sistema de equações diferenciáveis de primeira ordem

$$\begin{cases} \dot{x} = y \\ \dot{y} = M^{-1}(x) \cdot [T - N(x, y)] \end{cases}$$

Seja

$$z(t) = \text{col}(x(t), y(t))$$

o estado do sistema em cada instante podemos rescrever as equações da dinâmica como:

$$\begin{aligned} \dot{z} &= f(z, T) \\ \text{onde } f &= \text{col}(f_1, f_2) \\ f_1(x, y, T) &= y \\ \text{e } f_2(x, y, T) &= M^{-1}(x) \cdot [T - N(x, y)] \end{aligned}$$

Sendo o nosso objectivo minimizar o tempo da trajectória (t_f) entre dois estados (z_0 e z_f), podemos escrever o problema na forma

$$(P) \quad \text{Minimizar } t_f + K \cdot \|z(t_f) - z_f\|^2$$

Sujeito a:

$$\begin{cases} \dot{z} = f(z, T) \\ z(0) = z_0 \\ T(t) \in \Omega(t) \end{cases}$$

onde K é uma constante positiva escolhida suficientemente grande de modo a penalizar o erro com que se atinge o estado final pretendido e $W(t)$ o conjunto dos valores possíveis dos binários a aplicar em cada instante.

O vector adjunto para este problema satisfaz

$$\begin{cases} -\dot{p}^T(t) = p^T(t) \cdot \nabla_z f(z, T) \\ -p(t_f) = 2K(z(t_f) - z_f) \end{cases}$$

o que em termos de velocidade e posição se traduz nas equações

$$\begin{cases} -\dot{p}_1^T(t) = p_2^T(t) \cdot \nabla_x f_2(x, y, t) \\ -\dot{p}_2^T(t) = p_1^T(t) + p_2^T(t) \cdot \nabla_y f_2(x, y, t) \\ -p_1(t_f) = 2K(x(t_f) - x_f) \\ -p_2(t_f) = 2K(y(t_f) - y_f) \end{cases}$$

4.4 Princípio do máximo

Dada a complexidade do problema, torna-se impossível o cálculo da sua solução de forma fechada. O algoritmo iterativo apresentado abaixo permite a melhoria progressiva de uma estimativa da solução sendo a sua convergência detectada pelas condições necessárias de optimalidade na forma do Princípio do Máximo de Pontryagin. Este resultado pode ser descrito do seguinte modo:

Princípio do Máximo:

Seja (T, t_f) a solução de (P)

Então, existe uma função p , absolutamente contínua, satisfazendo

$$\begin{cases} \dot{-p}^T(t) = p^T(t) \cdot \nabla_{x,y} h(x(t), y(t), T(t)) \\ \dot{-p}^T(t_f) = 2K((x(t_f), y(t_f)) - (x_1, y_1)) \end{cases}$$

onde $T(t)$ maximiza, em $\Omega(t)$, a função do pseudo-Hamiltoniano

$$\tau \rightarrow H(x(t), y(t), p(t), \tau) \quad (1)$$

definida por $H(x(t), y(t), p(t), \tau) = p^T(t) \cdot h(x(t), y(t), \tau)$ no intervalo $[0, t_f]$ e satisfazendo

$$H(x(t_f), y(t_f), p(t_f), T(t_f)) = 1 \quad (2)$$

e

$$h(x, y, T) = \begin{bmatrix} y \\ f(x, y, T) \end{bmatrix}$$

Uma vez que $\nabla_{x,y} h(x, y, T)$ é dado por

$$\nabla_{x,y} h(x, y, T) = \begin{bmatrix} 0 & 1 \\ \frac{\partial}{\partial x} [-M^{-1}(x) \cdot [N(x, y) - T]] & \frac{\partial}{\partial y} [-M^{-1}(x) \cdot [N(x, y) - T]] \end{bmatrix}$$

tem-se que a variável adjunta $p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$ satisfaz

$$\begin{bmatrix} \dot{-p}_1^T & \dot{-p}_2^T \end{bmatrix} = \begin{bmatrix} p_2^T \cdot \nabla_x f(x, y, T) & p_1^T + p_2^T \cdot \nabla_y f(x, y, T) \end{bmatrix}$$

o que permite escrever a maximização do pseudo-Hamiltoniano na forma

$$p_2(t) f(x(t), y(t), T(t)) \geq p_2(t) f(x(t), y(t), \tau)$$

para todo o τ em $\Omega(t)$ e para quase todo o $t \in [0, t_f]$ onde t_f satisfaz (2).

4.5 Algoritmo de resolução

O algoritmo iterativo utilizado para calcular a solução de (P) é constituído pelos seguintes passos:

A. Inicialização

Escolhem-se os valores iniciais para estratégia de controlo T_0 , tal que $T_0(t) \in \Omega(t)$, fixa-se o contador e iterações, i , em zero e considera-se N suficientemente grande.

B. Cálculo de trajectória e variável adjunta

Para obter a trajectória e a variável adjunta integram-se as respectivas equações diferenciais satisfazendo as condições de fronteira e estipula-se t_f como sendo o primeiro instante, entre 0 e N , em que $\|z(t) - z_i\|$ toma o menor valor possível.

C. Teste de optimalidade

Verificar se (1) e (2) são simultaneamente válidos.

Caso positivo o algoritmo termina.

Doutro modo vai para D.

D. Preparação da iteração seguinte

Neste passo ir-se-á proceder a uma optimização unidimensional para a especificação completa da nova estimativa.

$$T_{i+1}(t) = T_i(t) + \delta_i \cdot \nabla_T H(x(t), y(t), p(t), q(t), T_i(t))$$

onde $\delta_i > 0$ é tal que o custo de (P) é mínimo e $T_{i+1}(t) \in \Omega$.

Faz-se $i = i + 1$ e vai-se para 2.

Qualquer processo de controlo admissível que não satisfaz o princípio do máximo não pode ser óptimo. Mais, o gradiente do Hamiltoneano em relação ao controlo dá uma direcção de descida do custo.

Podemos assim concluir que: o princípio do máximo não só diz quando é que uma estimativa da solução não é aceitável como também, sugere estratégia de actualização desta, conveniente de forma a baixar o custo da função objectivo.

4.6 Exemplo. Manipulador com dois graus de liberdade

O exemplo que se segue é um manipulador robótico planar com dois graus de liberdade.

A sua dinâmica foi obtida através do método explicado no ponto 9.2 e 9.3.

A lagrangeana que define o manipulador é a seguinte:

$$L=K-P$$

K , a energia cinética, vem dada por (ver notação mais abaixo)

$$K=1/2m_1\dot{r}_1^2+1/2m_2[\dot{l}_1^2\dot{q}_1^2+r_2^2\dot{q}_{12}^2+2l_1r_2\dot{q}_1\dot{q}_{12}C_2)]$$

e P, a energia potencial,

$$P=m_1gr_1S_1+m_2g[l_1S_1+r_2S_{12}]$$

A notação que vai ser utilizada é a seguinte

q	[q ₁ ,q ₂]
x	[x ₁ ,x ₂]
y	[y ₁ ,y ₂]
T	[T ₁ ,T ₂]
q ₁₂	q ₁ +q ₂
C ₁	cos(q ₁)
C ₂	cos(q ₂)
C ₁₂	cos(q ₁₂)
S ₁	sin(q ₁)
S ₂	sin(q ₂)
S ₁₂	sin(q ₁₂)

Desenvolvimento da equação da dinâmica:

$$\frac{dL}{d\dot{q}_1} = m_1\dot{r}_1^2\dot{q}_1+1/2m_2[2\dot{l}_1^2\dot{q}_1+2r_2^2\dot{q}_{12}+2l_1r_2\cos(q_2)\dot{q}_1+2l_1r_2\cos(q_2)\dot{q}_{12}]$$

$$\frac{d}{dt}\frac{dL}{d\dot{q}_1} = m_1\dot{r}_1^2\ddot{q}_1+1/2m_2[2\dot{l}_1^2\ddot{q}_1+2r_2^2\ddot{q}_{12}+2l_1r_2(\ddot{q}_1\cos(q_2)-\dot{q}_1S_2\dot{q}_2)+2l_1r_2(\ddot{q}_{12}\cos(q_2)-\dot{q}_{12}S_2\dot{q}_2)]$$

$$\frac{dL}{d\dot{q}_2} = 1/2m_2[2r_2^2\dot{q}_{12}+2l_1r_2\dot{q}_1\cos(q_2)]$$

$$\frac{d}{dt}\frac{dL}{d\dot{q}_2} = 1/2m_2[2r_2^2\ddot{q}_{12}+2l_1r_2(\ddot{q}_1\cos(q_2)-\dot{q}_1S_2\ddot{q}_2)]$$

$$dL/dq_1 = -m_1gr_1\cos(q_1)-m_2g(l_1\cos(q_1)+r_2\cos(q_2+q_2))$$

$$dL/dq_2 = -m_2l_1r_2\sin(q_2)[\dot{q}_1\dot{q}_{12}+\dot{q}_1^2]-m_2gr_2\cos(q_1+q_2)$$

$$T_1=d(dL/d\dot{q}_1)/dt-dL/dq_1 =$$

$$\ddot{q}_1[r_1^2 m_1 + l_1^2 m_2 + r_2^2 m_2 + 2l_1 r_2 m_2 \cos(q_2)] + \ddot{q}_2[r_2^2 m_2 + l_1 r_2 m_2 \cos(q_2)] - 2\dot{q}_1 \dot{q}_2 l_1 r_2 m_2 \sin(q_2) - m_2 l_1 r_2 \sin(q_2) \dot{q}_2^2 + g[m_1 r_1 \cos(q_1) + m_2(r_2 \cos(q_1 + q_2) + l_1 \cos(q_1))]$$

$$T_2 = d(dL/d\dot{q}_2)/dt - dL/dq_2 = \ddot{q}_1 m_2 [r_2^2 + l_1 r_2 \cos(q_2)] + \ddot{q}_2 m_2 r_2^2 + m_2 l_1 r_2 \sin(q_2) \dot{q}_1^2 + m_2 g r_2 \cos(q_1 + q_2)$$

Obtemos, finalmente, a equação da dinâmica dada por

$$T = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q}$$

Sendo T o vector dos binários a aplicar (o controlo)

$$T = M(q, \dot{q}) \cdot \ddot{q} + N(q, \dot{q})$$

desprezando os atritos dinâmicos e estáticos.

Matricialmente, M , o momento de inércia do sistema, é dado por

$$M(q) = \begin{bmatrix} r_1^2 m_1 + m_2 (l_1^2 + r_2^2 + 2l_1 r_2 C_2) & r_2^2 m_2 + l_1 r_2 m_2 C_2 \\ m_2 (r_2^2 + l_1 r_2 C_2) & m_2 r_2^2 \end{bmatrix}$$

$V_m(q, \dot{q})$ os termos de binário devido às forças centríptas e de Coriolis por

$$V_m(q, \dot{q}) = \begin{bmatrix} -2\dot{q}_1 \dot{q}_2 l_1 r_2 m_2 S_2 - l_1 r_2 m_2 S_2 \dot{q}_2^2 \\ l_1 r_2 m_2 S_2 \dot{q}_1^2 \end{bmatrix}$$

$G(q)$ os termos de binário devido à força gravitacional por

$$G(q) = \begin{bmatrix} g(r_1 m_1 C_1 + m_2 (r_2 C_{12} + l_1 C_1)) \\ g m_2 r_2 C_{12} \end{bmatrix}$$

A partir da equação da dinâmica e fazendo as mudanças de variáveis

$$\begin{aligned} x_3 &= q_1 \\ y_4 &= \dot{q}_2 \\ y_1 &= \dot{q}_1 \\ y_2 &= q_2 \end{aligned}$$

com

$$\dot{y}_1 = \frac{T_1 \cdot e - T_2 \cdot b + b \cdot f - c \cdot e}{a \cdot e - b \cdot d}$$

$$\dot{y}_2 = \frac{T_2 \cdot a - T_1 \cdot d - a \cdot f + c \cdot d}{a \cdot e - b \cdot d}$$

$$\dot{y}_3 = y_1$$

$$\dot{y}_4 = y_2$$

sendo,

$$a = m_1 r_1 r_1 + m_2 (l_1 l_1 + r_2 r_2 + 2 l_1 r_2 \cos(y_4));$$

$$b = m_2 (r_2 r_2 + l_1 r_2 \cos(y_3));$$

$$c = -2 m_2 l_1 r_2 \sin(y_4) y_1 y_2 - m_2 l_1 r_2 \sin(y_4) y_2 y_2 + g (m_1 r_1 \cos(y_3) + m_2 (r_2 \cos(y_3 + y_4) + l_1 \cos(y_3)));$$

$$d = m_2 r_2 r_2 + m_1 l_1 r_2 \cos(y_4);$$

$$e = m_2 r_2 r_2;$$

$$f = m_2 l_1 r_2 \sin(y_4) y_1 y_1 + m_2 g r_2 \cos(y_3 + y_4);$$

4.6.A INICIALIZAÇÃO E ESTRATÉGIA INICIAL

A estratégia inicial consiste na definição de um controlo T_0 tal que $T_0(t) \in \Omega$. No nosso algoritmo a solução adoptada para encontrar um conjunto de binários (controlos) admissíveis, ou seja aqueles que levam o manipulador da sua posição inicial até ao estado final desejado num intervalo de tempo finito, foi a implementação de um controlador PD.

A escolha deste controlador baseou-se nos seguintes pressupostos:

- a possibilidade de poder escrever o binário em função de $x(t)$, $y(t)$ e $\dot{y}(t)$;
- o uso do controlador P (proporcional à posição) permite que o manipulador atinja o estado final em tempo finito e com apenas um pequeno erro;
- o uso do controlador D (proporcional à velocidade) permite que esse estado final seja atingido mais rapidamente.

Descrição do controlador

Sabendo que a dinâmica do manipulador é

$$\begin{cases} \dot{x} = y \\ \dot{y} = -M^{-1}(x) \cdot [N(x, y) - T] \end{cases} \quad (3)$$

em que $N(x, y) = V_m(x, y) \cdot \dot{y} + G(x)$,

Se o binário $\bar{T} \in \Omega$ num determinado instante t for dado por

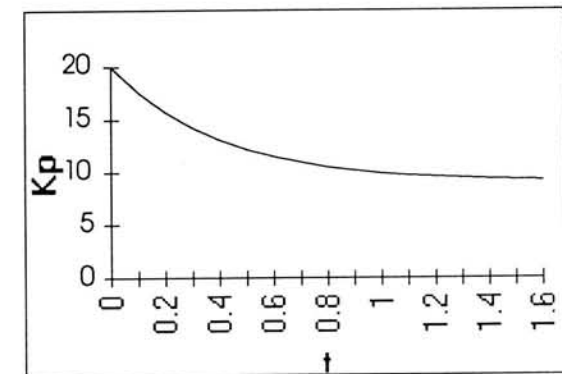
$$\bar{T} = V_m(x, y) \cdot \dot{y} + G(x) + M(x) \cdot [K_p \cdot I \cdot (x_f - x) - K_D \cdot I \cdot (y - y_f)] \quad (4)$$

onde I é a matriz identidade de dimensão n , x e y representam, respectivamente, o valor da posição e da velocidade dos braços do manipulador nesse instante e x_f e y_f os valores da posição e velocidade finais pretendidas, então a aceleração ($\ddot{y}$) é dada (depois de substituir (4) em (3)) por

$$\ddot{y} = K_p \cdot I \cdot (x_f - x) - K_D \cdot I \cdot (y - y_f)$$

Pode-se ver que, em cada instante, a aceleração é proporcional aos erros de posição e velocidade.

Para melhorar o desempenho, projectamos o controlador com os parâmetros K_p e K_D variáveis no tempo. Assim, K_p e K_D dependem do instante de tempo e são descritas pelas seguintes funções:



e

$$K_p = 0.55 \cdot \bar{K}_p \cdot e^{(t/\tau)} + \bar{K}_p \cdot 0.45$$

$$\text{com } \bar{K}_p = 20 \\ \tau = 0.4$$

if ($t < 3\tau$)

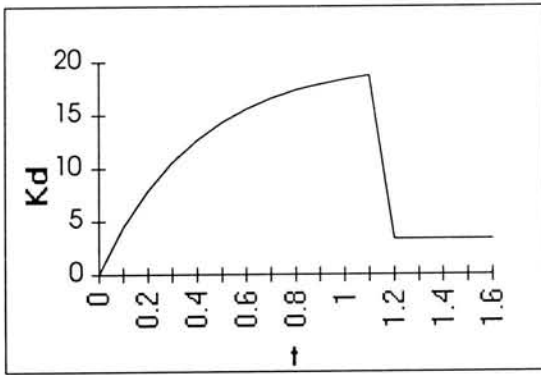
$$K_D = \bar{K}_D - \bar{K}_D \cdot e^{(-t/\tau)};$$

else

$$K_D = \bar{K}_D / 6;$$

$$\bar{K}_D = 20$$

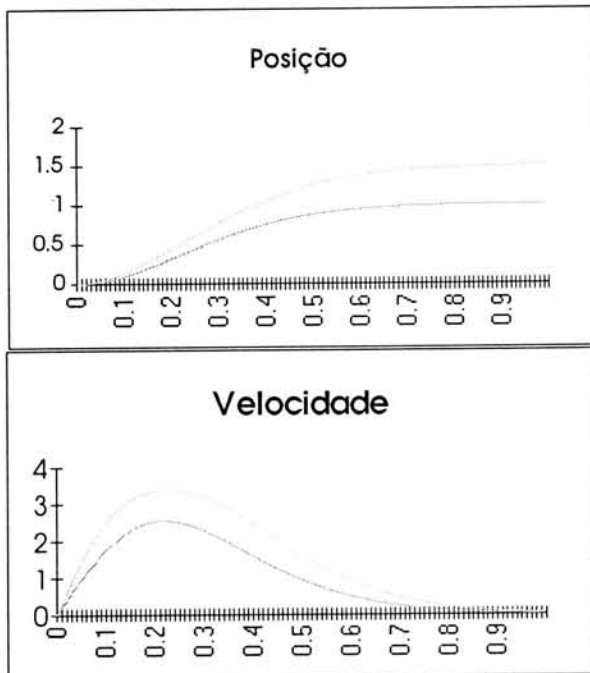
$$\tau = 0.4$$

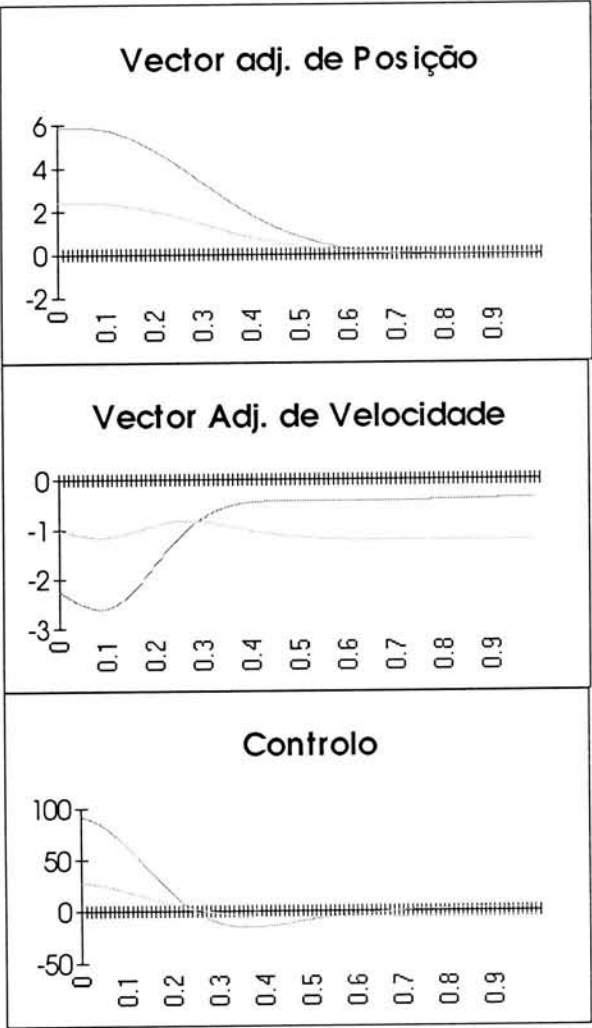


Como é visível, no início, K_P tem um valor superior a K_D de modo a imprimir ao manipulador uma aceleração que depende maioritariamente do erro de posição. Quando este se começa a aproximar do estado final, interessa-nos que o controlador seja mais proporcional ao erro de velocidade do que ao erro de posição (começar a travar antes de se atingir o estado final). Quando nos encontramos já muito perto do estado final (isto é, com uma velocidade pequena), diminuimos K_D de modo a que K_P seja maior, para que o manipulador chegue finalmente ao o estado final.

Tendo mostrado como se obtiveram as equações da dinâmica e a estratégia inicial vamos passar a analisar os resultados.

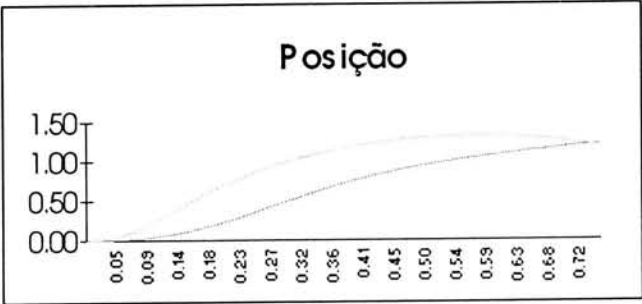
Estratégia inicial

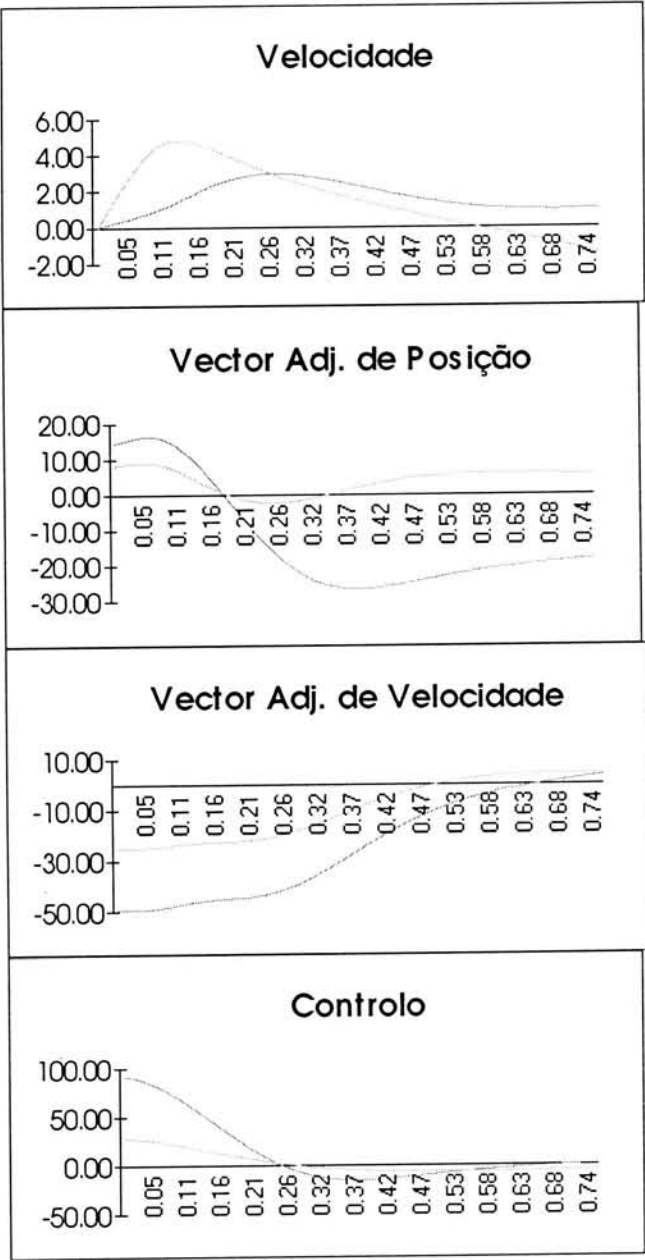




Esta iteração tem um custo associado de 33 e um t_f de 0.9 s.

Iteração final





O resultado do algoritmo como se pode verificar é uma solução bastante melhorada da anterior, apresentando um custo final de 8 com um t_{final} de 0.54 s.

5. TRANSFORMAÇÃO DO PROBLEMA DE TEMPO LIVRE NUM PROBLEMA DE TEMPO FIXO

5.1 Introdução

Neste capítulo, através de uma mudança de variável, vamos transformar o problema inicial de tempo livre num outro equivalente mas de tempo fixo. A vantagem desta transformação de variável é a de que para este tipo de problemas (tempo fixo) existem uma maior variedade de algoritmos desenvolvidos e de melhor conhecimento destes. Vamos, portanto, tentar encontrar a mesma solução que obteríamos no caso de termos o problema original.

Assim, na primeira secção, vamos mostrar como se faz a mudança de variável e que problema é que obtemos para de seguida lhe aplicarmos o princípio do máximo. Na terceira secção descrevemos o algoritmo que permite resolver computacionalmente o problema, focando principalmente as principais diferenças entre os algoritmos até agora apresentados. Para finalizar este capítulo daremos um exemplo para mostrar o seu funcionamento. Para mostrar que se obtém a mesma solução, o exemplo é o mesmo da secção 6.1.

5.2 Mudança de variável

Para recordar, o problema de tempo livre pode ser escrito da seguinte forma

$$(P) \quad \begin{cases} \text{Min } t_f + g(x(t_f)) \\ \dot{x}(t) = f(t, x(t), u(t)) \\ x(0) = x_0 \\ u(t) \in \Omega(t) \end{cases} \quad \text{q.s. em } [t_0, t_f]$$

Para termos um problema equivalente mas que a variável independente varie entre um intervalo sempre fixo (por exemplo, entre zero e um) temos de considerar uma nova variável τ . Fazendo, então, τ a nova variável de tempo o estado e o controlo o ficam

$$\tilde{x}(\tau) = (x \circ t)(\tau)$$

$$\tilde{u}(\tau) = (u \circ t)(\tau)$$

e o problema de tempo fixo (Pfixo) pode ser escrito da seguinte forma

$$(P_{\text{fixo}}) \quad \text{Min } t(1) + g(\tilde{x}(1))$$

$$\begin{cases} \dot{t}(\tau) = v(\tau) \\ \dot{\tilde{x}}(\tau) = f(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)) \\ \tilde{x}(0) = x_0 \\ t(0) = 0 \\ v(\tau) > 0 \\ \tilde{u}(\tau) \in \Omega \end{cases}$$

Então:

$(\tilde{u}(\tau), v(\tau), \tilde{x}(\tau), t(\tau))$ é Sol de (Pfixo) $\Leftrightarrow (x(t), u(t), t_{final})$ é Sol de (P)

$$\text{com } \begin{cases} \tilde{x}(\tau) = (x \circ t)(\tau) = x(t(\tau)) \\ \tilde{u}(\tau) = (u \circ t)(\tau) \\ t(1) = t_f, \quad t(0) = 0 \\ \dot{t}(\tau) = v(\tau) \end{cases}$$

5.3 Princípio do Máximo

Princípio do máximo para (Pfixo)

Seja $(\tilde{u}(\tau), v(\tau), \tilde{x}(\tau), t(\tau))$ sol. de (Pfixo)

Então $\exists \tilde{p}, \tilde{q}: [0, 1] \rightarrow \mathfrak{R}^n \times \mathfrak{R}$ tal que

$$\text{com } \begin{cases} -\dot{\tilde{p}}^T(\tau) = \tilde{p}^T(\tau) \cdot f_x(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)) \\ -\dot{\tilde{q}}(\tau) = \tilde{p}^T(\tau) \cdot f_t(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)) \\ -\tilde{p}(1) = g_x(\tilde{x}(1)) \\ -\tilde{q}(1) = 1 \end{cases}$$

e $(\tilde{u}(\tau), v(\tau))$ maximiza $(\tilde{u}, v) \rightarrow H(t(\tau), \tilde{x}(\tau), \tilde{p}(\tau), \tilde{q}(\tau), \tilde{u}, v)$ em Ω .

Seja

$$H(t(\tau), \tilde{x}(\tau), \tilde{p}(\tau), \tilde{q}(\tau), \tilde{u}(\tau), v(\tau)) = \tilde{p}^T(\tau) \cdot f(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)) \cdot v(\tau) + \tilde{q}(\tau) \cdot v(\tau)$$

Então $v(t) \in \mathfrak{R} \Rightarrow v(t)$ maximiza

$$0 = \frac{d}{dv} H(t(\tau), \tilde{x}(\tau), \tilde{p}(\tau), \tilde{q}(\tau), \tilde{u}(\tau), v(\tau)) = \tilde{p}^T(\tau) f(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)) + \tilde{q}(\tau)$$

$$\tilde{u}(\tau) \text{ maximiza } \tilde{u} \rightarrow [\tilde{p}^T(\tau) \cdot f(t(\tau), \tilde{x}(\tau), \tilde{u}) + \tilde{q}(\tau)] v^*(\tau)$$

$\Downarrow$

$$\tilde{u}(\tau) \text{ maximiza } \tilde{u} \rightarrow \tilde{p}^T(\tau) \cdot f(t(\tau), \tilde{x}(\tau), \tilde{u}(\tau)) \quad \text{em } \Omega \text{ em } [0, 1]$$

Departamento de Engenharia Electrotécnica e Computadores
Faculdade de Engenharia da Universidade do Porto
1993

Comunicações Apresentadas

MÉTODOS DE CONTROLO ÓPTIMO APLICADOS A MANIPULADORES ROBÓTICOS

Jorge Manuel Mendes Silva Gonçalves

ALGORITMOS DE TEMPO MÍNIMO PARA PROBLEMAS DE CONTROLO EM ROBÓTICA

Fernando Lobo Pereira
Rui Jorge Tunes
Jorge Manuel Gonçalves

I.S.R., F.E.U.P. - DEEC
Rua dos Bragas
4099 Porto Codex

Sumário

Neste artigo, vai ser dado um método de optimização em tempo mínimo para manipuladores de precisão em operações de fabrico repetidas. Dado que manipuladores sofisticados estão a ser cada vez mais usados, na prática, torna-se necessário encontrar maneiras de melhorar o seu desempenho, nomeadamente nos casos em que a trajectória é conhecida estamos particularmente preocupados com o tempo gasto em cada operação.

Introdução

Neste artigo propomos uma metodologia para encontrar estratégias de controlo óptimo para um manipulador robótico a desempenhar tarefas repetitivas numa linha de produção de um sistema de manufactura.

Neste contexto, uma vantagem importante da utilização de manipuladores consiste na flexibilidade necessária quando os locais para onde os componentes ou materiais se deslocam dependerem do resultado de operações anteriores (de inspecção por exemplo) e, embora conhecidos à priori, poderão variar por períodos com curta frequência.

Tipicamente, a minimização do tempo de ciclo da produção leva também a minimizar o tempo de manuseamento pelo manipulador. Por outro lado, o alto desempenho destas operações, aliado à grande precisão requerida, levam à consideração de modelos o mais precisos possíveis dos sistemas envolvidos.

Deste modo, a estratégia de controlo do manipulador deve ser obtida como solução de um problema de tempo mínimo sujeito não só às restrições dinâmicas, bem como a restrições estáticas associadas ao espaço livre para movimentação e limites máximos de velocidade.

Nesta abordagem as estratégias de controlo optimizantes são calculadas através da aplicação de um algoritmo de controlo óptimo. Este consiste num processo iterativo em que sucessivas estimativas da solução são melhoradas até encontrar um processo de controlo que satisfaz as condições necessárias de optimalidade. A trajectória assim obtida constituirá a referência para o controlador robótico estável e robusto (a perturbações e incertezas).

Descrição do problema

No espaço operacional a dinâmica de um manipulador robótico com n graus de liberdade poderá ser descrita por

$$T = M(q) \cdot \ddot{q} + N(q, \dot{q})$$

onde

$$N(q, \dot{q}) = V_m(q, \dot{q}) \cdot \dot{q} + G(q) + F_d \cdot \dot{q} + F_s(\dot{q})$$

Sendo q o vector posição do extremo do manipulador, T um vector com n componentes que descrevem o binário a aplicar pelos motores do robot ao manipulador, $M(q)$ uma matriz de $n \times n$ (finita, simétrica e positiva definida) $V_m(q, \dot{q})$ uma matriz de $n \times n$ contendo os termos centrípetos e de coriolis, $G(q)$ um vector com n elementos contendo os termos gravitacionais e F_d e $F_s(\dot{q})$ termos que vão representar o atrito dinâmico e estático respectivamente (F_d matriz $n \times n$ e $F_s(\dot{q})$ vector de dimensão n).

Estas equações em conjunto com restrições do espaço físico ao vector q (obstáculos, alcance físico do manipulador, etc.) irão de determinar um conjunto de trajectórias possíveis que irão servir de base ao nosso trabalho.

Para cada operação, que pretendamos fazer com o manipulador, existe um elevado grau de liberdade da trajectória a escolher. Geralmente dado um ponto de partida e sua velocidade, estado inicial, pretende-se atingir um determinado estado final (posição e velocidade desejada) acrescentando algumas restrições estáticas do tipo: posições por onde se deva passar por entre obstáculos, etc.

Será objectivo deste nosso trabalho encontrar a melhor trajectória, obedecendo às restrições impostas e, em particular, analisaremos o caso em que se pretende otimizar o tempo de execução de uma determinada operação.

Habitualmente, este tipo de planeamento de trajectórias é feito mais ou menos on-line por controladores, sem que haja a preocupação em otimizar, mas, apenas escolher uma trajectória dentro de um conjunto de possíveis trajectórias, obedecendo a critérios de robustez e estabilidade [5].

Não é pretensão deste trabalho calcular controlos óptimos on-line mas apenas calcula-los off-line, para posteriormente serem usados como referência (input) de controladores clássicos de manipuladores que elaborem operações em processos repetitivos como é o caso mais frequente em vários processos industriais.

Reformulação do problema:

A equação que nos dá o binário é a seguinte:

$$T = M(q) \cdot \ddot{q} + N(q, \dot{q})$$

$$\text{sendo } N(q, \dot{q}) = V_m(q, \dot{q}) \cdot \dot{q} + G(q) + F_d \cdot \dot{q} + F_s(\dot{q})$$

Fazendo uma mudança de variável

$$x = q$$

$$\text{e } y = \dot{q}$$

obtemos,

$$\dot{y} = -M^{-1}(x) \cdot (N(x, y) - T)$$

$$\dot{x} = y$$

Lembrar que o objectivo para cada percurso é fase-lo o mais rápido possível minimizando deste modo o tempo total de operação.

Escrevendo matematicamente o que se acabou de dizer, obtemos:

$$(P) \quad \text{Minimizar} \quad t_f + K \cdot \|x(t_f) - x_1\|^2$$

Sujeito a:

$$\dot{x} = y$$

$$(1) \quad \dot{y} = f(x, y, T)$$

$$x(0) = x_0$$

$$T(t) \in \Omega$$

sendo K uma constante positiva suficientemente grande

$$\text{e } f(x, y, T) = -M^{-1}(x) \cdot [N(x, y) - T]$$

Método de Solução

Dada a complexidade do problema, torna-se impossível o cálculo da sua solução de forma fechada. O algoritmo interactivo apresentado abaixo (ver [1]) permite a melhoria progressiva de uma estimativa da solução sendo a sua convergência detectada pelas condições necessárias de optimalidade na forma de um Princípio do Máximo de Pontrygin. Este resultado pode ser descrito do seguinte modo:

Seja (x, y, T) solução de (P)

Então, $\exists (p, q)$ satisfazendo:

$$(2) \quad -\dot{p}^T(t) = q^T(t) \cdot \nabla_x f(x, y)$$

$$-\dot{q}^T(t) = p^T(t) + q^T(t) \cdot \nabla_y f(x, y)$$

$$(3) \quad -p(t_f) = 2K(x(t_f) - x_1)$$

$$-q(t_f) = 0$$

$$(4) \quad H(x(t_f), y(t_f), p(t_f), q(t_f), T(t_f)) = 1$$

$$\text{e } T(t) \text{ maximixe em } \Omega$$

$$\tau \rightarrow q^T(t) \cdot f(x(t), y(t), \tau(t)) \text{ no intervalo } [0, t_f]$$

$$\text{onde } H(x, y, p, q, T) = p^T(t) \cdot y(t) + q^T(t) \cdot f(x, y, T)$$

Algoritmo de resolução

Conforme já foi referido, o algoritmo iterativo tem os seguintes passos:

1. Inicialização

Fixa-se:

- Contador de iterações $i=0$
- Estratégia de controlo T_0 tal que $T_0(t) \in \Omega$

e considera-se N suficientemente grande.

2. Cálculo de trajectória e variável adjunta

Integrando (1) em $[0, N]$ e utilizando as condições de optimalidade adequadas obtém-se X_0, Y_0 .

De (2) e (3) obtém-se p e q para o intervalo $[0, N]$.

Define T_f^0 como sendo o primeiro instante em que $\|X(t) - X_i\|$ tomam o menor valor possível em $[0, N]$

3. Teste de optimalidade

Verificar se (3) e (4) são simultaneamente válidos.

Caso positivo o algoritmo termina.

Doutro modo vai para 4.

4. Actualização do controlo e contador de iterações

$T_{i+1} = T_i + \delta_i \nabla_T H(X, Y, p, q, T)$ onde $\delta_i > 0$ é tal que o custo de (P) desce e $T_{i+1}(t) \in \Omega$
Faz $i = i + 1$ e vai para 2.

Os aspectos delicados deste algoritmo consistem na escolha do passo δ_i bem como na escolha da estratégia de controlo inicial.

Enquanto que a escolha do passo em cada iteração é efectuada memorizando aproximadamente o custo associado com o controlo T_{i+1} , a definição da estratégia de controlo inicial pode ser obtida através de heurísticas que produzem uma solução de T a partir de uma versão discreta que aproxima as funções envolvidas no problema.

Conclusões

Neste artigo foi dado um algoritmo para controlo óptimo de tempo mínimo. Este é caracterizado por ser iterativo e é baseado no princípio do máximo. O principal resultado deste artigo é mostrar que num problema em que o manipulador robótico realiza tarefas repetitivas é possível obter para todos os casos as trajectórias óptimas que vão servir de referência ao controlador e que permitem realizar essas mesmas tarefas em tempo mínimo, o que como vimos traz grandes vantagens. A implementação do algoritmo descrito para a resolução deste tipo de problema permite obter bons resultados, apesar do tempo de computação obrigar a um cálculo off-line.

Bibliografia e Referências

- [1] F. Lobo Pereira e J. Borges de Sousa, An Algorithm for Optimal Control Problems, ICIAM, 1991
- [2] D. M. Dawson, Z. Qu e S. Lim, Re-thinking the Robust Control of Robot Manipulators, CDC, 1991
- [3] Jack Macki e Aaron Strauss, Introduction to Optimal Control Theory, Springer-Verlag, 1982
- [4] Arthur E. Bryson, Jr. e Yu-Chi Ho, Applied Optimal Control, Hemisphere Publishing Corporation, 1975

MINISTÉRIO DA EDUCAÇÃO
DIREÇÃO-GERAL DO ENSINO SUPERIOR
SUBPROGRAMA 4 - PRODEP
MEDIDA 4.3

Bibliografia

- [Aubin 89] Aubin J. and Cellina A. (1984). "Differential Inclusions: Set-Valued Maps and Viability Theory", Springer-Verlag, NY.
- [Bertsekas 76] Bertsekas D. (1976). "On the Goldstein-Levitin-Polyak Gradient Projection Method", IEEE Transactions on Automatic Control, vol. AC-21, No 2, pp. 174-184.
- [Bonnans 86] Bonnans J. (1986). "On an algorithm for optimal control using Pontryagin's maximum principle", SIAM Journal of Control and Optimization, vol 24, pp. 579-588.
- [Clarke 90] Clarke, F. (1990) "Optimization and Nonsmooth Analysis", SIAM.
- [Dontchev 92] Dontchev A. and Lempio F. (1992). "Difference Methods for Differential Inclusions: A Survey", SIAM Review, vol. 34, N° 2, pp. 263-294.
- [Frankowska 89] Frankowska, H. (1989) "Contingent Cones to Reachable Sets of Control Systems", SIAM Journal of Control and Optimization, pp. 170-198.
- [Gurman 84] Gurman (1984), "Control of Mixed Discrete-Continuous Systems", Systems & Control Letters, 3, pp. 23-35.
- [Halkin 66] Halkin, H. (1966). "A Maximum Principle of the Pontryagin Type for Systems Described by Nonlinear Difference Equations", SIAM Journal of Control, Vol. 4, No. 1, pp 91-111.
- [Hauser 86] Hauser J. (1986). "Proximity Algorithms: Theory and Implementation", Department of Electrical Eng. and Computer Science, University of California Berkeley, CA, Memo. UCB/ERL M86/35, 1986.
- [Kaskosz 85] Kaskosz B. and Lojasiewicz S. (1985). "A Maximum Principle for Generalized Control Systems", Nonlinear Analysis, 9, pp. 109-130.
- [Loewen 87] Loewen P. and Vinter R. (1987), "Pontryagin-type necessary conditions for differential inclusion problems", Systems & Control Letters, 9, pp. 263-265.
- [Mayne 87] Mayne D. and Polak E. (1987). "An Exact Penalty function Algorithm for Control Problems with State and Control Constraints", IEEE Trans. on Autom. Control, vol. AC-32, No 5, pp. 380-387.
- [Mayne 90] Mayne, D. Q. and Michalska, H. (1990). "Robust Receding Horizon Control", Proceedings of the 29th IEEE Conference on Decision and Control, pp. 575-581.
- [Polak 77] Polak E. and Mayne D., "A Feasible Directions Algorithm for Optimal Control Problems with Control and Terminal Inequality Constraints", IEEE Transactions on Automatic Control, vol. AC-22, No 5, 1977, pp. 741-751.
- [Polak 90] Polak E., Yang T. and Mayne D. (1990). "A method of centers based on barrier functions for solving optimal control problems with continuum state and control constraints", Proceedings of 29th Conf. on Decision and Control, Hawaii, pp. 2327-2332.
- [Sakawa 80] Sakawa Y. and Shindo Y. (1980). "On global convergence of an algorithm for optimal control", IEEE Transactions on Automatic Control, AC-25, pp. 1149-1153.
- [Vinter --] Vinter, R. B. and Wolenski, P. R. (to appear) "Adjoint variables and the value function in optimal control theory: the measurable case", Journal of Mathematical Analysis and Applications.
- [Vinter 91] Vinter, R. B. and Michalska, H. (1991). "Receding Horizon Control for Nonlinear Time-Varying Systems", Proceedings of the 30th IEEE Conference on Decision and Control, pp. 75-76.



FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000101629