

Sincronismo de eixos com o PLC Siemens S7-1500

Matias Rodrigues Rocha

Dissertação de Mestrado

Orientador na FEUP: Prof. Doutor António Pessoa de Magalhães

Orientador na Gislotica: Eng.º Rui Manuel Fazenda da Silva



Mestrado Integrado em Engenharia Mecânica

Ramo de Automação

Junho de 2018

Resumo

O controlo de movimento é um tópico em rápida evolução e em alta demanda no universo industrial atual. Nesse sentido, os principais fabricantes presentes no mercado da automação industrial dedicam forte atenção à engenharia de recursos capazes de dar resposta em tempo-real às exigências envolvidas nesta área de controlo.

Uma das mais recentes soluções apresentadas pela Siemens disponibiliza funcionalidades para controlo de movimento integradas nas CPUs dos PLCs das linhas S7-1200 e S7-1500. O objetivo final da dissertação passa pela aplicação das potencialidades destas ferramentas numa unidade de pultrusão, equipamento de fabrico de produtos compósitos de secção constante anteriormente desenvolvido pela empresa proponente da dissertação, e cujo *hardware* e *software* são considerados obsoletos.

Nesta unidade, umas das etapas mais preponderantes para o seu funcionamento envolve a sincronização de movimento entre os dois carros de arrasto que provocam o movimento do material, movimento esse que terá que ser mantido a uma velocidade constante. No entanto, as ferramentas a utilizar apresentam algumas limitações neste campo do controlo de movimento. Assim, um dos objetivos principais deste trabalho envolve também o estudo das limitações destas na área do sincronismo de movimento.

O documento que aqui inicia encontra-se dividido em três partes diferentes. Numa primeira fase, é realizada uma exposição das ferramentas a utilizar e uma análise dos seus pontos fortes, através de uma comparação com a solução da Siemens anteriormente utilizada pela empresa para a mesma gama aplicacional, na qual os algoritmos de controlo eram realizados exclusivamente no *drive*. O foco estará sobretudo na forma como as instruções de movimento são especificadas numa e noutra solução.

Numa segunda etapa, dedicada ao campo aplicacional do sincronismo de eixos, é realizada uma análise das razões que levaram a uma migração das abordagens mecânicas de sincronismo de movimento para soluções eletrónicas, assim como um enquadramento das necessidades de sincronismo de movimento no ambiente industrial, com a apresentação de algumas aplicações típicas. Posteriormente, serão apontadas as limitações das ferramentas em estudo para o campo aplicacional do sincronismo de eixos.

Por último, será apresentada a parte experimental relativa ao trabalho realizado, envolvendo desde as etapas de análise do equipamento em causa, passando pelas adaptações realizadas, e terminando com a exposição das funcionalidades implementadas. Ainda neste capítulo, e uma vez que o equipamento não se encontra nas instalações da empresa, mecanismos de simulação permitem validar a aplicabilidade das ferramentas na aplicação em causa.

Palavras-Chave: Controlo de movimento, Resposta em tempo-real, CPU, PLC, Automação industrial, Sincronismo de eixos

Abstract

Multi-Axis Synchronization with PLC Siemens S7-1500

Motion control is a rapidly evolving and in high demand topic in today's industrial universe. In that sense, leading manufacturers in the industrial automation dedicate close attention to the engineering of resources capable of responding in real-time to the requirements involved in this control area.

One of the latest solutions offered by Siemens provides motion control functionalities as an integral part of every CPU of the PLCs's lines S7-1200 and S7-1500. The main purpose of this dissertation is to apply the potential of these tools in a pultrusion unit, manufacturing equipment of composite products of constant section previously developed by the applicant company's of the dissertation, whose hardware and software are obsolete.

In this unit, one of the most important steps for its operation involves the movement synchronization between the two trailing cars that cause material movement, which will have to be kept at a constant speed. However, the tools to be used have some limitations in this area of motion control. Thus, one of the foremost aims of this work involves the study of their limitations in the area of motion synchronism.

This document is divided into three different parts. Initially, a display of the tools to be used and an evaluation of their strengths is performed through a comparison with Siemens solution previously used by the company for the same application range, in which the control algorithms were carried out exclusively in the drive. Focus will be on how motion instructions are specified in both solutions.

Secondly, dedicated to the axes synchronization field, an analysis of the reasons that led to a migration of the mechanical approaches of movement synchronism to the electronic solutions is made, as well as a framework of the needs of movement synchronism in the industrial environment, presenting some typical applications. Subsequently, limitations of the tools under study for the field of application of the axes synchronism will be pointed out.

Finally, the experimental part related to the work will be presented, from the analysis phases of the equipment, through the adjustments and the exposure of the implemented features. Still in this chapter, and given that the equipment is not on the premises of the company, simulation mechanisms allows to validate the applicability of the tools in the application in study.

Keywords: Motion control, Real-time response, CPU, PLC, Industrial automation, Multi-axis synchronization

Agradecimentos

Em primeiro lugar gostaria de agradecer à Gislótica, não só por toda a hospitalidade, como pela disponibilização de todos os meios essenciais ao trabalho desenvolvido. Em particular ao Engenheiro Humberto Ramos, por todo o seu acompanhamento e disponibilidade ao longo deste projeto.

Agradeço também ao Engenheiro Rui Manuel Fazenda da Silva e ao Professor Doutor Francisco Jorge Teixeira de Freitas por terem possibilitado o desenvolvimento deste trabalho na empresa Gislótica.

Gostaria ainda de agradecer ao Professor Doutor António Pessoa de Magalhães pela orientação e o aconselhamento dado na realização desta tese, sobretudo na fase de escrita da mesma.

Agradeço também a toda a malta que me tem acompanhado ao longo destes últimos anos, pelo apoio, amizade e companheirismo que sempre demonstraram.

Por fim, e de maior importância, gostaria de agradecer à minha família, em particular aos meus pais e irmão, que sempre me deram o maior apoio possível para concretizar os meus objetivos. Obrigado!

Índice de Conteúdos

1	Introdução	1
1.1	Empresa Gislótica – Projeto e Fabrico de Sistemas Mecânicos, Lda	2
1.2	Objetivos da dissertação	2
1.3	Estrutura da dissertação	3
2	Controlo de movimento	5
2.1	Ferramentas de <i>software</i> e recursos de <i>hardware</i>	5
2.1.1	<i>TIA Portal</i>	5
2.1.2	<i>SINAMICS S120</i>	7
2.2	Controlo de Movimento no Drive	8
2.3	Controlo de Movimento no PLC	11
2.3.1	<i>Visão Geral</i>	11
2.3.2	<i>Nível de execução</i>	14
2.3.3	<i>Instruções de movimento</i>	14
2.4	Comparação das potencialidades das duas soluções	15
2.5	Síntese.....	16
3	Sincronismo de eixos.....	17
3.1	Evolução	17
3.1.1	<i>Abordagem Mecânica</i>	17
3.1.2	<i>Abordagem Eletrónica</i>	18
3.2	Exemplos de aplicações	19
3.3	Soluções de Sincronismo com o PLC S7-1500	20
3.4	Síntese.....	22
4	Descrição Detalhada da Aplicação em Análise	23
4.1	Unidade de Pultrusão	23
4.1.1	<i>Etapas do processo de fabrico</i>	23
4.1.2	<i>Etapas-Alvo</i>	24
4.2	<i>Hardware Inicial</i>	25
4.2.1	<i>Controlo e acionamento</i>	25
4.2.2	<i>Sensores Magnéticos</i>	26
4.2.3	<i>Segurança</i>	26
4.2.4	<i>Aquecimento</i>	27
4.3	Alterações a efetuar a nível de <i>hardware</i> e respetiva justificação	27
4.3.1	<i>Acionamento</i>	27
4.3.2	<i>Componentes mecânicos</i>	28
4.4	Especificação justificada dos modos de funcionamento / operação	28
4.5	Síntese.....	29
5	Caso de estudo.....	31
5.1	Componentes Siemens usados no projeto	31
5.1.1	<i>PLC</i>	31
5.1.2	<i>Módulos I/O</i>	32
5.1.3	<i>Interface Homem-Máquina (HMI)</i>	33
5.2	Ciclos Automáticos de Funcionamento	34
5.2.1	<i>Definição dos objetos tecnológicos utilizados</i>	37
5.2.2	<i>Ciclo de Pulling</i>	37
5.2.3	<i>Ciclo de corte</i>	40
5.3	Interfaces e Funcionalidades de Operação.....	42
5.3.1	<i>Menu de Operação</i>	43
5.3.2	<i>Alarmes</i>	44
5.3.3	<i>Entradas e Saídas</i>	48
5.3.4	<i>Configurações</i>	48
5.3.5	<i>Movimento Manual</i>	49
5.3.6	<i>Objetos tecnológicos</i>	50

5.4	Testes	51
5.4.1	<i>PLCSim</i>	51
5.5	Síntese	53
6	Conclusões e perspectivas de trabalho futuro	55
6.1	Trabalhos futuros	56
7	Bibliografia	59
ANEXO A:	Diagrama de Estados PLCOpen [41].....	63

Abreviaturas, Acrónimos e Siglas

CNC	Computer Numerical Control
CPU	Central Processing Unit
DB	Bloco de dados (Data Block)
FC	Função (Function)
FB	Bloco de Função (Function Block)
FEUP	Faculdade de Engenharia da Universidade do Porto
GRAFCET	GRAphe Fonctionnel de Commande Etape Transition
HMI	Human-Machine Interface
OB	Bloco de Organização (Organization Block)
PC	Personal Computer
PLC	Programmable Logic Controller
POU	Program Organization Unit
SFC	Sequential Function Chart
TO	Objeto Tecnológico (Technology Object)

Índice de Figuras

Figura 2.1 – Interface entre o sistema operativo de controladores e os diferentes blocos possíveis no programa do utilizador no TIA Portal [2].....	6
Figura 2.2 – Exemplo de arquitetura do <i>drive</i> S120 [4]	7
Figura 2.3 – Anéis de controlo envolvidos, (a) na solução com controlo de posição no <i>drive</i> e (b) na solução com posicionamento no PLC [9]	8
Figura 2.4 – <i>Basic Positioner</i> com responsabilidades de geração de trajetória [10].....	9
Figura 2.5 – Configuração de <i>Traversing Blocks</i> no <i>software</i> Starter – no caso do S120 CU320-2 PN é possível configurar até 64 instruções de movimento.....	9
Figura 2.6 – Fluxo de informação do lado do TIA Portal envolvido na especificação, leitura e escrita de instruções de movimento [11]	10
Figura 2.7 – FBs para configuração de instruções de movimento, a) SINA_POS, b) SINA_SPEED, c) SINA_PARA_S, d) SINA_PARA.....	11
Figura 2.8 – Interfaces tecnológicas dos objetos tecnológicos [13]	12
Figura 2.9 - Exemplo de configuração das propriedades mecânicas do objeto final	13
Figura 2.10 – Princípio de operação do TO_PositioningAxis [13]	13
Figura 2.11 – Interface do FB para posicionamento absoluto de um eixo	15
Figura 3.1 – a) Sistema de came mecânica e b) Exemplo de uma relação de movimentos conseguida com este sistema [16]	18
Figura 3.2 – Configuração Gantry [17]	19
Figura 3.3 – Valor de referência para o anel de controlo proveniente de a) <i>encoder</i> associado ao movimento do eixo <i>master</i> e b) <i>setpoint</i> de posição gerado pelo gerador de trajetória do eixo <i>master</i> [17].....	19
Figura 3.4 - a) Operação de corte por medida denominada por <i>Flying Knife</i> e b) relação de movimentos inerentes a esta operação [17].....	20
Figura 3.5 – Instruções de movimentos: a)MC_GearIn, b)MC_GearInPos, c)MC_PhasingRelative, d)MC_PhasingAbsolute e e)MC_CamIn	21
Figura 4.1 – Representação esquemática de uma unidade de pultrusão [21]	24
Figura 4.2 – Desenho de conjunto presente no manual do equipamento	25
Figura 4.3 – Arquitetura do <i>drive</i> S120 para a solução a implementar	27
Figura 5.1 – Controlador lógico programável S7-1500 CPU 1517-3 PN/DP [28].....	31
Figura 5.2 – a) Módulos da estação ET200SP [24] e b) Pormenor das Base Units [25].....	32
Figura 5.3 – Emulação dos módulos I/O no TIA Portal	33
Figura 5.4 – a) SIMATIC HMI KTP900F [36] e b) Connection Box [37]	34
Figura 5.5 – Modelo comportamental do ciclo de <i>pulling</i>	38
Figura 5.6 – Algumas instruções de movimento utilizadas: a)MC_MoveAbsolute, b)MC_MoveJog, c)MC_MoveAbsolute, d) MC_GearIn, e)MC_MoveAbsolute e f) MC_GearIn.....	40
Figura 5.7 – Modelo comportamental do ciclo de corte.....	41

Figura 5.8 - Árvore de ecrãs desenvolvidos.....	42
Figura 5.9 – Menu de Operação.....	43
Figura 5.10 – Configuração de alarmes discretos.....	44
Figura 5.11 - a) Limitação imposta por estas funcionalidades e b) Configuração no TIA Portal.....	45
Figura 5.12 – Exemplo de um código de erro disponibilizado nas <i>words</i> do bloco de dados do objeto tecnológico.....	46
Figura 5.13 – Alarme proveniente da não calibração do <i>encoder</i> associado ao servomotor de acionamento do <i>charriot</i> 1: a) Armazenamento no bloco de dados e b) Representação na HMI.....	46
Figura 5.14 – Erro de supervisão originado pela especificação de uma velocidade demasiado alta para o processo.....	47
Figura 5.15 – Ecrã de entradas e saídas do sistema.....	48
Figura 5.16 – Ecrãs de configurações disponíveis.....	48
Figura 5.17 – Instrução MC_Home para calibração/referência dos <i>encoders</i>	49
Figura 5.18 – Ecrã de operação manual.....	49
Figura 5.19 – a) MC_MoveJog, b) MC_MoveVelocity e c) MC_Halt.....	50
Figura 5.20 – Ecrã com informações relevantes dos objetos tecnológicos.....	50
Figura 5.21 – a) Ferramenta para simulação <i>PLCSim</i> e b) Configuração a realizar nos objetos tecnológicos.....	51
Figura 5.22 - <i>Tracing</i> do período final de sincronismo entre os <i>charriots</i> 1 e 2.....	52
Figura 5.23 – Bancada de testes: a) <i>Drive</i> , b) PLC.....	52

Índice de Tabelas

Tabela 2.1 – Blocos de função do TIA Portal necessários para especificação de instruções de velocidade.....	10
Tabela 2.2 – Diferentes objetos tecnológicos presentes no S7-1500	12
Tabela 2.3 – Exemplos da aplicabilidade de alguns FBs presentes na biblioteca disponibilizada pela PLCOpen	15
Tabela 2.4 – Recursos da CPU associados à utilização de cada objeto tecnológico [12]	16
Tabela 5.1 – Referências dos módulos I/O a utilizar.....	33
Tabela 5.2 – Características funcionais da HMI KTP900F [36]	34
Tabela 5.3 – Variáveis de entrada	35
Tabela 5.4 – Variáveis de saída	35
Tabela 5.5 – Variáveis adicionais.....	36
Tabela 5.6 – Comentários adicionais às instruções de movimento	40
Tabela 5.7 – Diferença na contagem de bits nos SIMATIC S7 e no WinCC	44

1 Introdução

Os sistemas de acionamento eletromecânicos são hoje parte preponderante da automação industrial. Com uma gama de aplicabilidade vasta, têm como um dos seus principais atributos a versatilidade de possibilidades de controlo a eles afetas, sendo estas impulsionadas pela evolução constante da eletrónica de processamento de informação.

O aparecimento destes recursos teve como elementos motrizes exigências de desempenho e de flexibilidade a nível de movimento mais elevadas do que as encontradas nas máquinas construídas com as soluções mecânicas e hidráulicas convencionais de até então. Associado ao seu desenvolvimento e às crescentes exigências a nível de controlo de movimento, tornou-se imprescindível o aparecimento e desenvolvimento de sistemas que suportassem “o controlo de eixo”.

O primeiro passo nesse sentido passou pelo desenvolvimento de módulos dedicados, com capacidades de microprocessamento especiais a ser conectados aos PLCs responsáveis pelo controlo de funcionalidades gerais das máquinas. Equipados com saídas analógicas para envio de *setpoints* para os *drives*, e capazes de receber e interpretar minimamente sinais de *feedback* de sensores de posição, esta foi durante muito tempo a solução típica (e “avançada”) de controlo de movimento no meio industrial.

No entanto, devido ao pioneirismo destas soluções e às suas limitações tecnológicas, a interação entre os ciclos de varrimento do PLC e os ciclos de varrimento destes módulos era ainda um pouco complicada, o que tornava difícil a implementação de rotinas de sincronização entre eles por parte do programador do PLC. Para além disso, em movimentações rápidas, a concretização do movimento poderia não ser registada pelo PLC, o que induziria incertezas ao programador. Para ultrapassar esta questão e garantir uma resposta eficiente em tempo-real, eram implementadas sub-rotinas que induziam um esforço computacional elevado e levavam ao limite a capacidade computacional dos PLCs da altura.

Com a evolução da tecnologia de processamento, as soluções para posicionamento de eixos acionados eletricamente progrediram, distribuíram-se e tornaram-se cada vez mais diversificadas, passando a orientar-se também para o controlo de velocidade e binário. Não só foi melhorada a dinâmica entre o ciclo de varrimento de controladores dedicados ao controlo de movimento e o ciclo do PLC de hierarquia superior, como houve a possibilidade de integração das funcionalidades de posicionamento nos equipamentos de controlo – tanto no *drive*, como no PLC.

A Siemens é um dos fabricantes de equipamentos de controlo que mais tem progredido e contribuído para o estado da arte das soluções de controlo de movimento. Uma das suas mais recentes soluções passa exatamente pelo último ponto especificado no parágrafo anterior: a integração de funcionalidades de controlo de movimento na CPU dos PLCs das famílias SIMATIC S7-1200 e S7-1500. Estes surgem como alternativas interessantes para resolver tarefas de controlo de movimento de múltiplos eixos independentes.

No entanto, a linha de controladores S7-1500 ainda vai mais longe, uma vez que possui capacidades de processamento e comunicação necessárias para garantir uma resposta em tempo-real à generalidade dos mais típicos problemas de sincronização de movimentos multi-eixo, recorrendo à técnica *master/slave* – também conhecida na literatura como técnica líder/seguidor. Esta técnica baseia-se em designar convenientemente um eixo como sendo o *master*, que realizará a trajetória para si requerida, enquanto que os eixos *slaves* seguem o movimento do primeiro.

É neste âmbito que surge esta dissertação. Assim, o que se pretende inicialmente é uma exploração das funcionalidades das ferramentas em questão para, numa fase posterior, se proceder a uma atualização das rotinas de funcionamento de uma máquina anteriormente construída pela empresa proponente da dissertação.

1.1 Empresa Gislotica - Projeto e Fabrico de Sistemas Mecânicos, Lda

Sediada em Matosinhos, a Gislotica é uma empresa especializada na construção de equipamentos para indústria de pneus, tendo como principais vertentes o projeto, fabrico e automação de sistemas mecânicos. Não obstante, possui o *know-how* necessário para concretizar o desenvolvimento de soluções em áreas distintas, como por exemplo na área dos materiais compósitos, para a qual desenvolveu equipamentos de pultrusão e enrolamento filamentar. Com uma base educacional elevada, inerente ao facto de ser constituída por um número significativo de professores e ex-professores, esta empresa aposta na inclusão de estudantes para resolver problemas pontuais, testar novas soluções ou realizar atualizações mecânicas ou de automação a soluções previamente construídas.

Uma questão relativa à empresa que é conveniente salientar é a sua aposta atual em acionamentos eletromecânicos concretizados por servomotores, quando estão envolvidas necessidades de posicionamento e controlo com elevada exatidão de velocidade. Assim, todas as considerações a realizar daqui para a frente nesta dissertação terão como base os servomotores e correspondente servo controlo, a menos que indicado o contrário.

1.2 Objetivos da dissertação

O objetivo final da dissertação passa pela atualização das rotinas de funcionamento de um equipamento anteriormente desenvolvido pela empresa, tendo em conta as possibilidades introduzidas pelas recentes ferramentas da Siemens para controlo de movimento no PLC, e que ficarão disponíveis para uma possível implementação futura.

O equipamento em estudo é uma unidade de pultrusão, no qual está patente uma necessidade de sincronismo de movimento entre dois eixos. Assim, um dos pontos principais desta dissertação será conhecer e analisar as limitações inerentes a estas ferramentas nessa área de controlo de movimento. A sua interpretação será essencial para o planeamento das rotinas a construir.

Após o desenvolvimento das rotinas de funcionamento, um passo seguinte será a conceção de uma interface homem-máquina que permita impor as diferentes funcionalidades requeridas para o equipamento. Uma vez que o mesmo não se encontra dentro das instalações da empresa, a validação tanto das rotinas como da HMI será realizada por simulação no *software* de programação destas, o TIA Portal Professional V14 SP1.

Por fim, a organização e o relato do trabalho desenvolvido é também um objetivo desta dissertação.

1.3 Estrutura da dissertação

O documento encontra-se dividido em seis capítulos, servindo o presente para a apresentação, contextualização e motivação do trabalho realizado.

O capítulo 2 apresenta e expõe as ferramentas a utilizar. Para tal, procede-se a uma comparação entre estas e a solução percursora da Siemens utilizada pela Gislotica para a mesma gama aplicacional, que embora não sejam aquelas que estão empregues no equipamento, permitem justificar as grandes vantagens das ferramentas em causa para controlo de movimento no PLC.

O capítulo 3 centra-se na questão do sincronismo de movimento entre eixos e tem como objetivo principal a discriminação das principais limitações das ferramentas apresentadas no capítulo anterior para suprimir necessidades de sincronismo de movimento entre eixos em ambientes industriais.

O capítulo 4 descreve ao pormenor a aplicação sobre a qual a implementação prática recaiu. Apresenta as características mecânicas mais importantes do sistema, assim como especifica as funcionalidades pretendidas. Por fim, aponta também as adaptações a realizar tendo em vista as ferramentas a utilizar.

O capítulo 5 é dedicado à implementação realizada, com o relato dos problemas e soluções adotadas. As ferramentas de simulação para a realização da validação das rotinas são também apresentadas.

O capítulo 6 encerra a dissertação. Nele são especificadas as principais conclusões a retirar do trabalho realizado, assim como são apresentadas sugestões de trabalhos futuros.

2 Controlo de movimento

O presente capítulo expõe as ferramentas para controlo de movimento nos PLCs das linhas S7-1200 e S7-1500 da Siemens. O seu enquadramento é realizado confrontando esta alternativa com a solução de controlo de movimento no *drive* S120, que era, até à introdução desta, a solução da Siemens praticada pela Gislotica para posicionamento de eixos nas suas máquinas. No entanto, dadas as especificidades das duas soluções, torna-se obrigatória a apresentação prévia de alguns conceitos-chaves a nível de *hardware* e *software* necessários para a posterior explicação das mesmas.

Assim, numa primeira fase serão explorados os conceitos em causa, para, numa fase posterior, serem apresentadas e exploradas as duas soluções de controlo de movimento em questão, com principal foco no modo como são especificadas as instruções de movimento. De seguida, será feito um balanço que permita ostentar as vantagens desta nova alternativa.

2.1 Ferramentas de *software* e recursos de *hardware*

Os conceitos-chave a introduzir neste capítulo estão relacionados com o *software* de programação TIA Portal e com o *drive* da família SINAMICS S120, equipamento presente nas duas soluções a explorar.

O TIA Portal apresenta-se como uma das plataformas mais completas para o projeto, desenvolvimento e implementação de soluções de automação industrial. É precisamente devido à sua elevada panóplia de funcionalidades que, para efeitos de estruturação do relatório, os pontos fortes do *software* relevantes no âmbito da dissertação serão apresentados de uma forma sucessiva e oportuna, e não de uma forma centralizada, permitindo uma melhor compreensão dos mesmos. Tendo em vista o objetivo deste capítulo, o foco estará relacionado com o seu modelo organizacional e respetiva interação com o objeto final de controlo.

No que diz respeito à família de *drives* SINAMICS S120, dar-se-á destaque à sua modularidade e à forma como é realizada a comunicação interna – isto é, entre os seus módulos – assim como a troca de informações com dispositivos de controlo de hierarquia superior. Outras questões relativas, por exemplo, à alimentação dos módulos e respetivos motores serão discutidas mais tarde no relato referente à parte prática desta dissertação.

2.1.1 TIA Portal

De modo a potenciar ao máximo as funcionalidades que um *software* disponibiliza, é deveras importante que o utilizador conheça e interprete adequadamente o seu modelo organizacional, sobretudo quando procura uma programação devidamente estruturada. A estruturação de um programa facilita a deteção, análise e respetiva solução de erros de programação, simplificando igualmente as fases de manutenção por parte do cliente final.

O modelo organizacional de um ambiente de programação encontra-se refletido na forma como as diferentes entidades unitárias de programação interagem umas com as outras. No caso

do TIA Portal, é concedido ao utilizador três diferentes tipos de unidades de programação [1, 2, 3]:

- **OB (Bloco de Organização)** – os blocos do tipo OB podem ser entendidos como a interface entre o sistema operativo do controlador e o programa do utilizador (Figura 2.1). São chamados pelo sistema operativo de acordo com a sua prioridade e, por sua vez, chamam de forma sequencial as sub-rotinas desenvolvidas em Blocos de Função (FB) ou em Funções (FC). Ao contrário do POU (*Program Organization Unit*) do tipo Programa, definido na norma IEC 61131-3 [1], os OBs já têm propriedades de *run-time* implícitas, que pré-definem precisamente a prioridade com que são invocados pelo sistema operativo. A título de exemplo, para além da OB *Main* – que é processada de ciclo em ciclo pelo controlador – é possível encontrar também OBs com propriedades de interrupções periódicas – OB *Cyclic Interrupt* – ou ainda OBs cujo *trigger* é dado por um evento físico – OB *Hardware Interrupt*.

- **FB (Bloco de Função)** – Ao contrário do anterior, este tipo de bloco já possui uma filosofia em tudo semelhante à homónima definida na norma IEC 61131-3. Tem um carácter sequencial, uma vez que os valores das suas variáveis internas e de saída são guardadas de ciclo para ciclo. Este armazenamento de memória permite que os dados provenientes de FBs possam ser disponibilizados a sistemas de supervisão e interfaces homem-máquina (HMI), por exemplo. O encapsulamento de outros FBs ou de FCs é também uma possibilidade inerente a um FB.

- **FC (Função)** – Por contraste à unidade organizacional precedente, os blocos deste tipo têm um carácter combinacional – ou seja, em diferentes ciclos, com os mesmos valores nas entradas, fornecem sempre os mesmos valores nas suas saídas – dado que não têm acesso aos valores obtidos em ciclos anteriores, uma vez que estes não são guardados em bancos de dados. Sempre que necessário, poder-se-á disponibilizar os valores das suas variáveis a blocos de dados.

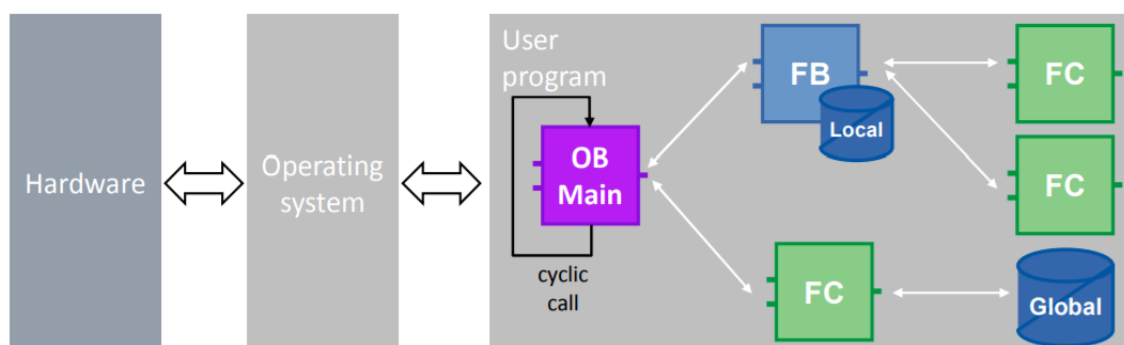


Figura 2.1 – Interface entre o sistema operativo de controladores e os diferentes blocos possíveis no programa do utilizador no TIA Portal [2].

Para além das unidades destinadas à programação de rotinas, o TIA Portal disponibiliza ainda blocos representativos de áreas dedicadas ao armazenamento de dados – denominados por DBs (Blocos de dados). Estes podem ainda ser subdivididos em DBs locais ou DBs globais: os primeiros são comumente utilizados para instanciamento exclusivo de FBs, não trocando dados com nenhum outro bloco; já os segundos têm os seus valores lidos/escritos por diferentes blocos ao longo de um ciclo de execução.

2.1.2 SINAMICS S120

Com uma arquitetura modular particularmente interessante para aplicações multi-eixo (Figura 2.2), o *drive* S120 tem como principais constituintes os seguintes [4, 5, 6]:

- Control Unit – unidade central encarregue do processamento de todas as funcionalidades tecnológicas do *drive*, desde a execução dos anéis de controlo dos motores a si associados, à comunicação em tempo-real com controladores de hierarquia superior, ou ainda a comunicação com o sistema de configuração e parametrização do *drive*.
- Módulo de Linha – gera tensão DC para os restantes módulos do *drive* a partir da alimentação trifásica da rede.
- Módulos Motores – convertem a tensão do barramento DC gerada pelo módulo de linha em alimentação AC para os motores.

A comunicação entre os componentes constituintes desta família de *drives* SINAMICS, é realizada por *Drive-Cliq*, rede baseada em Ethernet, montada sobre portas físicas RJ45 [7]. De entre as vantagens que esta comunicação possui, destacam-se a possibilidade de diagnósticos por parte da *Control Unit* de todos os módulos do *drive*, assim como o reconhecimento das propriedades desses nós. Para além disso, os dados do motor e *encoder* são guardados como placas eletrónicas num motor equipado com uma conexão *Drive-Cliq*, sendo essa informação transferida para a *Control Unit* durante a configuração do *drive* no *software* de parametrização do mesmo.

Quanto à comunicação cíclica com controladores de hierarquia superior, esta pode ser realizada por rede Profinet IO ou Profibus DP, à base de telegramas [8]. Estes telegramas não são mais do que sequências de *words* e *double words* estruturadas tendo em vista as informações a trocar.

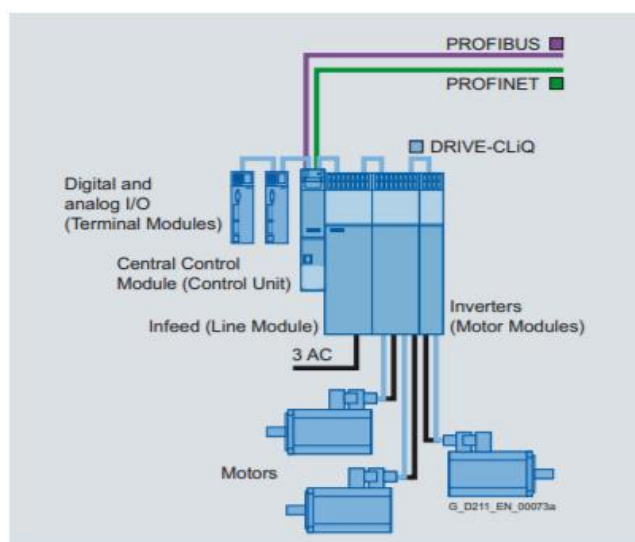


Figura 2.2 – Exemplo de arquitetura do *drive* S120 [4].

O controlo de movimento a realizar é, naturalmente, imposto pelas necessidades das aplicações envolvidas. Um dado binário motor implica necessariamente uma dada corrente nos enrolamentos do motor, cuja constância é mantida à custa do anel de controlo de corrente. O anel de controlo de velocidade fornece os *setpoints* de binário a serem tratados pelo anel de controlo de corrente para que o motor atinja a velocidade requerida. Por sua vez, o anel de

controle de posição irá gerar os *setpoints* de velocidade necessários para que o motor atinja, ou mantenha, a posição requerida.

Um dado perfil de movimento de um eixo pode ser refletido numa função posição *versus* tempo, o que no fundo traduz que o controle de movimento (no tempo) acaba por envolver implicitamente toda esta “cascata” de anéis de controle (Figura 2.3).

Antes de avançar para a apresentação das duas soluções para controle de movimento em estudo, importa ter a noção de que o controle em velocidade de um eixo – segmento da máquina que se move juntamente com o motor e o dispositivo de *feedback* associado (ao motor ou no lado da carga) – é uma funcionalidade básica presente em todos os *drives* da família SINAMICS. Assim, o que vai diferenciar entre as duas soluções será o local onde é realizado o controle de posição do eixo e os cálculos necessários para gerar a trajetória de movimento baseada nas instruções do utilizador, o que envolve, por exemplo, o condicionamento do valor de referência de posição que assegure um perfil de movimento compatível com a dinâmica do motor. Se estes são realizados no *drive*, fala-se portanto da arquitetura refletida na Figura 2.3a. Já a Figura 2.3b apresenta a arquitetura que envolve o controle de posicionamento a partir do PLC [9].

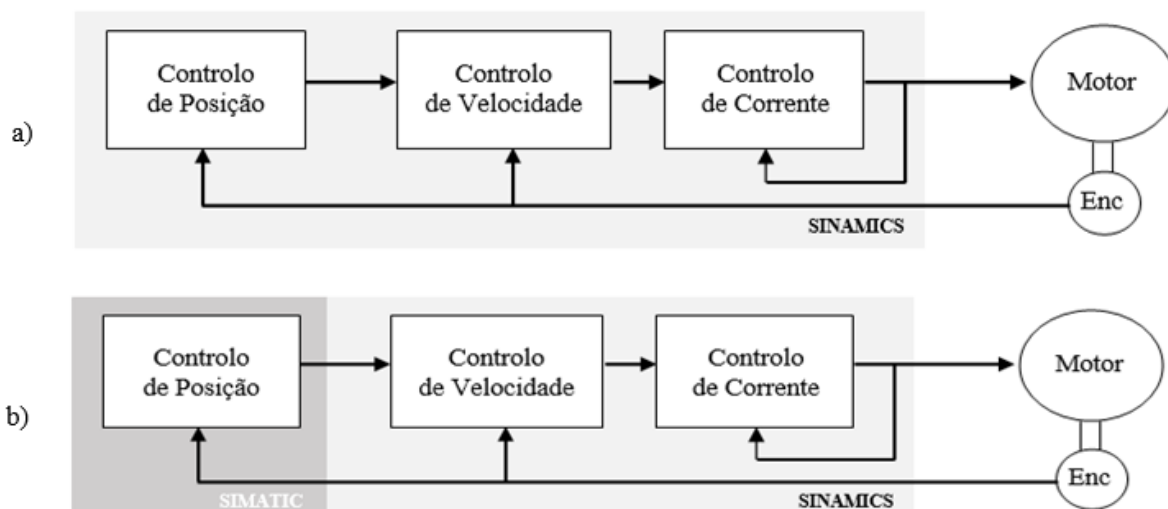


Figura 2.3 – Anéis de controle envolvidos, (a) na solução com controle de posição no *drive* e (b) na solução com posicionamento no PLC [9].

2.2 Controle de Movimento no Drive

O processamento de tarefas de posicionamento de eixos com o S120 é possibilitado devido à presença na sua *Control Unit* de um módulo funcional denominado por Basic Positioner (EPos – Figura 2.4). Com este módulo, os *drives* da Siemens ficam habilitados para resolver tarefas de posicionamento de eixos, dispensando um controlador de posição de hierarquia superior.

Com esta opção de controle de movimento no *drive*, é possibilitado ao utilizador a seleção de cinco modos de operação diferentes, dos quais se destacam o *MDI: Directly Input Setpoint* e os *Traversing Blocks* [10]. Através do primeiro, o utilizador define um *setpoint* único, enquanto que o segundo caso é utilizado para definir uma sequência de comandos de

movimentos e respetivos parâmetros cinemáticos (Figura 2.5). Esta sequência de movimento também poderia ser obtida através de alterações *on-the-fly* dos diversos *setpoints*.

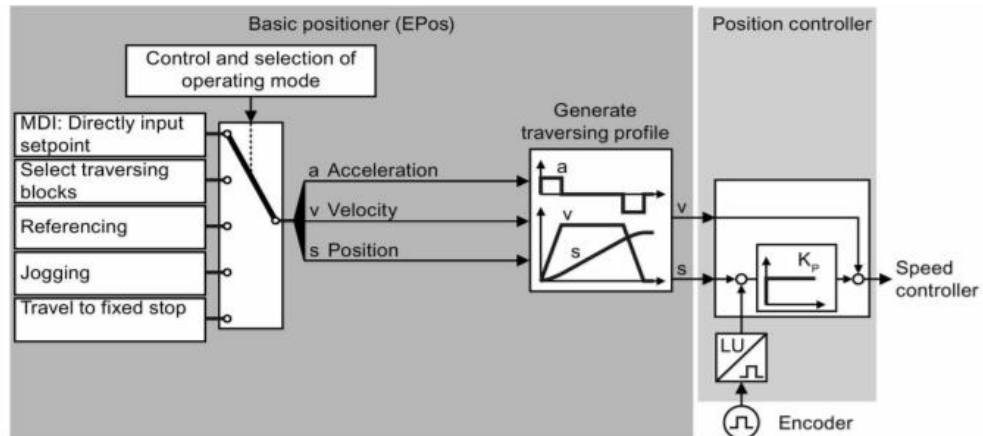


Figura 2.4 – *Basic Positioner* com responsabilidades de geração de trajetória [10].

index	Job	Parameter	Mode	Position	Velocity	Acceleration	Deceleration	Advance	Hide
1	-1	GOTO	0	ABSOLUTE	600	100	100	END (0)	<input checked="" type="checkbox"/>
2	-1	POSITIONING	0	ABSOLUTE	100	100	100	END (0)	<input checked="" type="checkbox"/>
3	-1	POSITIONING	0	ABSOLUTE	600	100	100	END (0)	<input checked="" type="checkbox"/>
4	-1	GOTO	0	ABSOLUTE	200	400	100		<input checked="" type="checkbox"/>
5	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
6	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
7	-1	POSITIONING	0	ABSOLUTE	200	200	100	CONTINUE_FLYING (2)	<input checked="" type="checkbox"/>
8	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
9	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
10	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
11	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
12	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
13	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
14	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
15	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
16	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
17	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
18	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
19	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
20	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
21	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
22	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
23	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
24	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
25	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
26	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
27	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
28	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
29	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
30	-1	POSITIONING	0	ABSOLUTE	0	600	100	END (0)	<input checked="" type="checkbox"/>
31	-1	POSITIONING	0	ABSOLUTE	600	100	100	END (0)	<input checked="" type="checkbox"/>

Figura 2.5 – Configuração de *Traversing Blocks* no *software Starter* – no caso do S120 CU320-2 PN é possível configurar até 64 instruções de movimento.

Embora se possam especificar as instruções de movimento no *software* de configuração e parametrização do drive – *Starter* –, o mais comum na maioria das aplicações industriais é a necessidade de especificar as instruções de controlo de movimento a partir de um controlador de hierarquia superior – até porque é este que permite a interface com os dispositivos HMI. Dependendo da informação a trocar, a comunicação pode ser cíclica – isto é, em cada ciclo de comunicação PROFINET, ou pode ser realizada quando há um pedido explícito para tal – ou seja, de forma acíclica. Neste sentido, o *software* de programação do PLC – o TIA Portal – está equipado com blocos de função para a configuração da informação a trocar e blocos de função responsáveis por acionar pedidos para a leitura e escrita de parâmetros do *drive*, que são imediatamente criados aquando da conceção dos primeiros. Na Tabela 2.1 são apresentados os blocos acima referidos, sendo ainda descritos exemplos da sua utilização. Por sua vez, na Figura 2.6 está ilustrado o modo como estes blocos interagem entre si.

Tabela 2.1 – Blocos de função do TIA Portal necessários para especificação de instruções de velocidade

FB – Configuração	FB – Escrita/Leitura	Comunicação	Aplicação	Exemplos
SINA_POS (FB284)	DPRD_DAT/DPWR_DAT	Cíclica	Eixo controlado em posição	Envio de <i>setpoints</i> para o <i>drive</i> ; seleção de instruções de movimento configuradas nos <i>Traversing Blocks</i> (apenas SINA_POS); leitura de valores atuais de posição / velocidade e outros parâmetros de interesse
SINA_SPEED (FB285)			Eixo controlado em velocidade	
SINA_PARA (FB286)	RDREC/WRREC	Acíclica	Leitura e escrita de até 16 parâmetros do <i>drive</i> .	Leitura e escrita de instruções de movimento presentes nos <i>Traversing Blocks</i>
SINA_PARA_S (FB287)			Leitura e escrita de qualquer parâmetro do <i>drive</i> .	

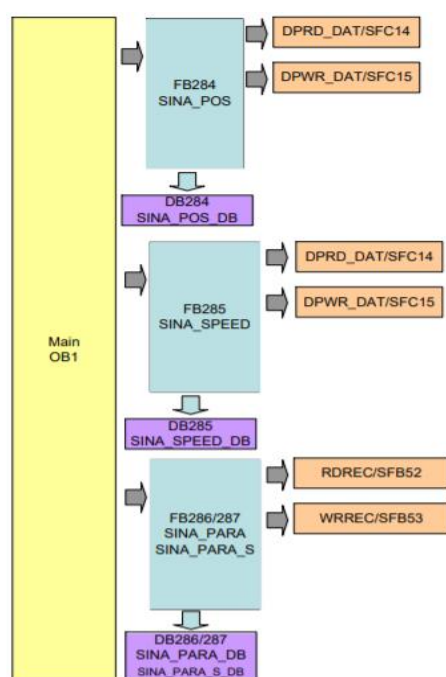


Figura 2.6 – Fluxo de informação do lado do TIA Portal envolvido na especificação, leitura e escrita de instruções de movimento [11].

Por sua vez, as interfaces para configuração das instruções de movimento encontram-se ilustradas na Figura 2.7.

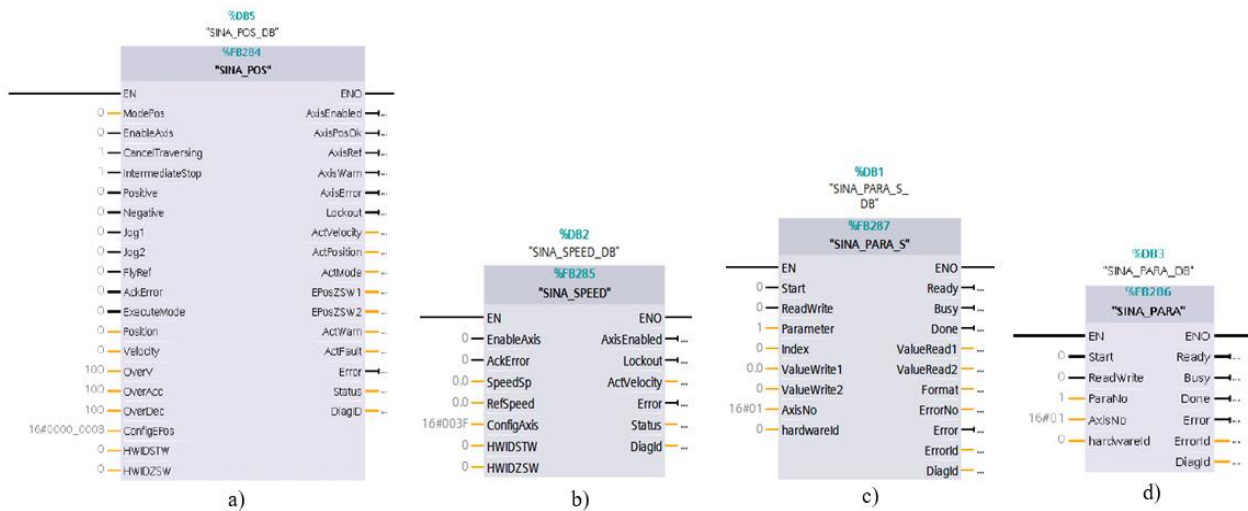


Figura 2.7 – FBs para configuração de instruções de movimento, a) SINA_POS, b) SINA_SPEED, c) SINA_PARA_S, d) SINA_PARA.

2.3 Controlo de Movimento no PLC

Tendo em vista o mesmo leque aplicacional da solução relatada na secção anterior, uma segunda alternativa a considerar passa pela utilização das recentes funcionalidades de controlo de movimento integradas na CPU dos controladores das famílias dos S7-1200 e S7-1500, libertando o *drive* da realização de algoritmos de posicionamento do eixo. Os recursos tecnológicos introduzidos são manipulados por parte do utilizador a partir de objetos tecnológicos (*Technology Objects*) presentes no TIA Portal – o ambiente de programação do PLC.

Uma vez que estas ferramentas serão utilizadas mais tarde, uma atenção mais aprofundada será dada às mesmas.

2.3.1 Visão Geral

Uma definição explícita e concisa para estes elementos tecnológicos é algo difícil de encontrar na literatura. Contudo, uma reunião das suas propriedades permite compreender um pouco das suas funcionalidades [11, 12]:

- Representam objetos, reais ou virtuais, no controlador.
- Representam os componentes mecânicos associados ao eixo.
- Encapsulam as funcionalidades tecnológicas presentes no *firmware* da CPU do PLC.
- Permitem uma configuração relativamente simples e intuitiva, incluindo a atribuição de limites dinâmicos aos eixos em questão.
- Realizam a monitorização do movimento dos eixos.

O esquema mostrado na Figura 2.8 procura representar a interação entre os objetos tecnológicos e as fronteiras tecnológicas dos mesmos. Para a correta aplicação dos comandos de controlo, o utilizador é responsável pela configuração destes elementos tecnológicos no

ambiente de programação TIA Portal, tendo ainda acesso à evolução do estado de variáveis de interesse, como por exemplo, a posição e velocidade atual de um eixo. Todos estes parâmetros - a nível da configuração, *setpoints*, valores atuais e informações de estado de eixos (como deteção de erros) - são escritos/lidos em blocos de dados dos respetivos objetos tecnológicos, representativos dos locais de memória da CPU destinados a armazenamento dos mesmos. O controlo e o comando de ações de movimento por parte dos eixos é realizada, na interface do utilizador, através de instruções de movimento normalizadas pela PLCOpen Motion Control, cabendo ao *firmware* da CPU o processamento das tarefas de movimento iniciadas pelo utilizador.

Tabela 2.2- Diferentes objetos tecnológicos presentes no S7-1500

Simbologia	Designação
	TO_SpeedAxis
	TO_PositioningAxis
	TO_SynchronousAxis
	TO_ExternalEncoder
	TO_MeasuringInput
	TO_OutputCam
	TO_CamTrack

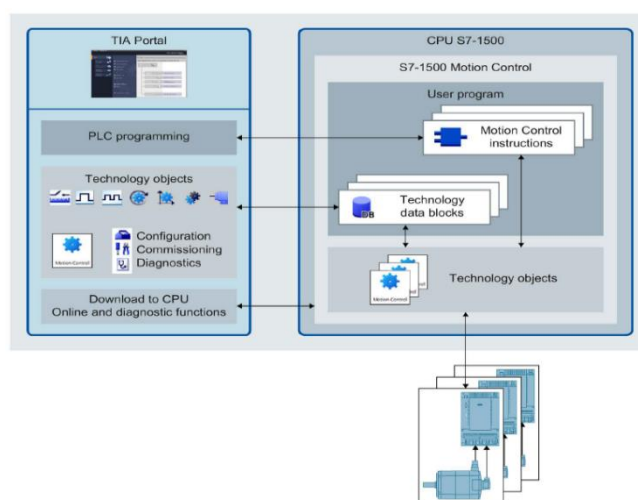


Figura 2.8 – Interfaces tecnológicas dos objetos tecnológicos [13].

O tipo de objeto tecnológico a utilizar depende da funcionalidade de controlo de movimento requerida. Caso o interesse da aplicação seja o controlo em velocidade de um eixo, o objeto tecnológico a utilizar é o *TO_SpeedAxis*, que permitirá calcular o *setpoint* de velocidade a enviar para o *drive* tendo em conta as propriedades mecânicas do objeto final, especificadas pelo utilizador na configuração do objeto tecnológico (Figura 2.9). Em sentido contrário, disponibiliza ao utilizador informações de estado relevantes do eixo, como a sua velocidade atual – caso esta esteja disponível – que é comunicado ao PLC pelo *drive*.

Se, por outro lado, a aplicação envolve necessidades de posicionamento do eixo, o objeto tecnológico a utilizar é o *TO_PositioningAxis*, responsável pelo processamento do *setpoint* de velocidade a enviar para o *drive*, tendo em conta a diferença de valores entre a posição atual do eixo e o *setpoint* de posição calculado durante o movimento (Figura 2.10).

Para aplicações que envolvem necessidades de sincronismo de movimento, o objeto tecnológico a aplicar ao eixo *slave* – ou seguidor – é o *TO_SynchronousAxis*. Importa também apontar que as funcionalidades dos três objetos tecnológicos acima referidos são cumulativas, no sentido em que o *TO_SynchronousAxis* possui capacidade para se comportar como um *TO_PositioningAxis*, que por sua vez consegue realizar as mesmas funções que um *TO_SpeedAxis*.

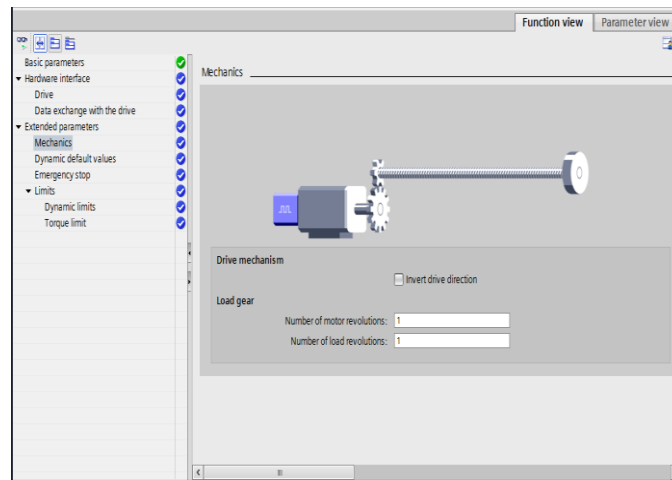


Figura 2.9 - Exemplo de configuração das propriedades mecânicas do objeto final.

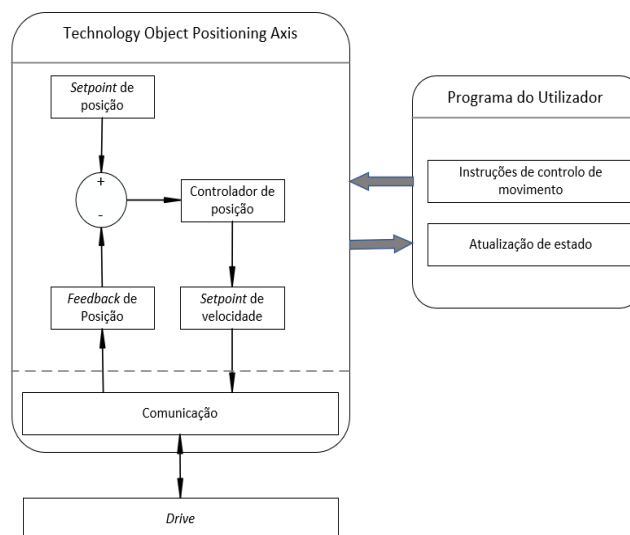


Figura 2.10 – Princípio de operação do TO_PositioningAxis [13].

Para além dos objetos tecnológicos acima referidos, são ainda disponibilizados os seguintes:

- TO_ExternalEncoder – permite a leitura dos valores de posição, velocidade ou aceleração de movimentos que não são diretamente influenciados por um eixo.
- TO_OutputCam – disponibiliza ao utilizador sinais booleanos consoante a posição de um dado eixo.
- TO_CamTrack – permite a especificação de até 32 *Output Cams*.
- TO_MeasuringInput – aquisição de valores de posição de um eixo ou *encoder* após um dado *trigger*.

O que se poderá equacionar nesta altura é de que forma estes três últimos objetos tecnológicos diferem de algo que se pode implementar, através de código, no ciclo de varrimento principal do PLC. Esta questão permite introduzir o próximo tópico de discussão – os níveis de execução associados ao objetos tecnológicos.

2.3.2 Nível de execução

Para realizar o processamento dos objetos tecnológicos de controlo de movimento, o PLC cria um nível de execução diferente, que será chamado periodicamente de acordo com o ciclo de aplicação de controlo de movimento. Normalmente, a prioridade e o intervalo entre ciclos de processamento destas unidades são suplantados apenas por funcionalidades de segurança.

Os blocos de organização responsáveis pelo tratamento destas unidades são denominados por MC-Servo [OB91] e MC-Interpolator [OB92]. O primeiro executa os algoritmos de controlo de posição para todas as unidades tecnológicas, sendo as questões relacionadas com a evolução das instruções de controlo de movimento, a monitorização e a geração de *setpoints* alicerçadas na execução do OB92. O intervalo entre processamentos destas unidades pode ser configurado pelo utilizador, resultando de um compromisso entre qualidade de controlo e carga para o PLC. Contudo, este valor temporal tem de ser superior ao requerido para o processamento de todas as unidades tecnológicas num mesmo ciclo – caso contrário é gerado uma informação de *overflow* pelo sistema, levando à paragem do processamento do PLC (mudança para o modo Stop) [12].

Apesar de serem criados blocos de organização especialmente designados para processar os objetos tecnológicos, isto não implica que as instruções para movimento sejam especificadas dentro dos mesmos – aliás, a escrita de código nestes OBs é impossível, sendo que o código a eles inerentes não é sequer visível – *know-how protection*. Estas instruções poderão ser especificadas em qualquer OB ou FB presente no *software*.

2.3.3 Instruções de movimento

A especificação de instruções de movimento no *software* de programação do PLC é realizado a partir de uma biblioteca de blocos de função normalizada criada pela organização PLCOpen. Uma das tarefas desta organização é, por exemplo, a atribuição de selos de conformidade com a norma IEC 61131-3 a produtos de diversos fabricantes [13].

No entanto, outra das práticas desta organização é resolver adicionais questões de normalização a nível industrial. Foi nesse âmbito que, com a ajuda das principais marcas de fabricantes a nível de automação, criaram ferramentas normalizadas para controlo de movimento, materializadas em bibliotecas de blocos de função.

Estes blocos de função permitem especificar diversas tarefas de controlo, desde simples calibrações e referências de *encoders*, passando por comandos de posicionamento absoluto, até às mais complexas envolvendo a sincronização de eixos. O diagrama de estados associado a estas instruções de movimento encontra-se no anexo A. Como se pode compreender, apresentar todas as instruções individualmente seria algo moroso. No entanto, alguns exemplos ilustrativos de alguns desses FBs e respetivas funcionalidades estão presentes na Tabela 2.3. Por sua vez, um exemplo de uma interface de um desses blocos no TIA Portal pode ser visto na Figura 2.11.

Tabela 2.3 - Exemplos da aplicabilidade de alguns FBs presentes na biblioteca disponibilizada pela PLCOpen

Blocos de Função	Função
MC_Home	Referenciamento ou calibração do objeto tecnológico
MC_Halt	Cessação do movimento do motor
MC_MoveAbsolute	Posicionamento absoluto do motor
MC_MoveRelative	Posicionamento relativo do motor
MC_MoveVelocity	Movimento em velocidade do motor
MC_MoveJog	<i>Jogging</i> do motor

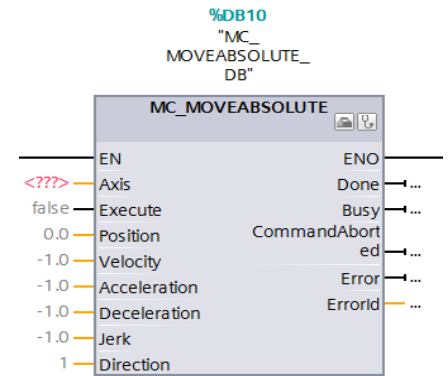


Figura 2.11 – Interface do FB para posicionamento absoluto de um eixo.

2.4 Comparação das potencialidades das duas soluções

De facto, a grande vantagem desta nova alternativa em relação à anterior – com controlo de posicionamento pelo *drive* – reside neste último ponto apresentado: a possibilidade de se especificar comandos para movimentos através de instruções normalizadas, num *software* já por si composto por linguagens de programação normalizadas, o que tem impactos significativos no que pode ser considerado como o ciclo de vida de um *software*, que engloba desde a fase da sua conceção, até à fase de manutenção por parte do cliente final.

Esta questão, aliada ao facto de o *software* de configuração do *drive* – Starter – disponibilizar uma panóplia de funcionalidades extensa, o que torna o processo de seleção de funcionalidades um pouco mais complicado, assim como a deteção de erros neste, justifica a migração para esta nova solução. É nesse sentido que a Siemens tem incluindo pouco-a-pouco as funcionalidades deste para o TIA Portal, possibilitando que toda a programação seja realizada num só ambiente de programação.

A necessidade do PLC realizar complexos algoritmos de controlo terá, naturalmente, algum impacto no desempenho do mesmo. Assim, é expectável que cada CPU tenha as suas possibilidades e limitações a nível dos recursos para controlo de movimento que podem ser utilizados. Por exemplo, no caso da família dos S7-1200, de entre os objetos tecnológicos apresentados na Tabela 2.2, apenas se encontra disponível o objeto tecnológico para posicionamento do eixo. Já dentro de controladores da mesma família, o mesmo tipo de objetos tecnológicos estão disponíveis, sendo a grande diferença de controlador para controlador o número de recursos que cada um disponibiliza. Com isto, associado a cada CPU, a SIEMENS informa sobre o número de recursos que esta disponibiliza, que, por sua vez, limita o número de objetos tecnológicos passíveis de ser usados com esse controlador (Tabela 2.4). Por exemplo, se se tratar de um S7-1511 PN, encontram-se disponíveis 800 recursos, enquanto que um S7-1517-3 PN/DP disponibiliza 10240 recursos.

Tabela 2.4 – Recursos da CPU associados à utilização de cada objeto tecnológico [12]

Objeto Tecnológico	Recursos utilizados
TO_SpeedAxis	40
TO_PositioningAxis	80
TO_SynchronousAxis	160
TO_ExternalEncoder	80
TO_MeasuringInput	40
TO_OutputCam	20
TO_CamTrack	160

Esta questão da limitação das capacidades inerentes às CPUs do PLC é importante, e suscita para uma realidade que é a limitação que esta alternativa tem em desempenhar alguns problemas mais complexos, como aqueles encontrados em aplicações que envolvem sincronismo de eixos. Nesse sentido, no capítulo 3 pressupõe-se analisar e apresentar as possibilidades e limitações desta solução na área do sincronismo de movimento.

2.5 Síntese

Este capítulo descreveu e comparou as duas soluções para posicionamento de eixos elétricos concretizadas por *hardware* Siemens normalmente utilizadas pela Gislotica, tendo como principal interesse o modo como as instruções de movimentos são especificadas num e noutro caso.

Constatou-se a pertinência de soluções normalizadas no impacto do ciclo de vida de um *software*, principalmente na parte referente à manutenção do mesmo por parte do cliente final, diminuindo tempos de aprendizagem. Esta questão é importante, sobretudo num mercado tão competitivo como o da automação industrial.

Por fim, foram ainda apresentadas algumas considerações sobre as limitações a nível tecnológico destas ferramentas, e que serão estendidas à discussão do próximo capítulo.

3 Sincronismo de eixos

A necessidade de sincronização multi-eixo surge sempre que diversos eixos se devem mover simultânea ou sequencialmente, e a relação entre os respetivos movimentos é importante. Esta questão está patente desde aplicações que envolvem a movimentação de uma ferramenta num deslocamento resultante da soma vetorial de deslocamentos sobre eixos ortogonais – máquinas CNC – até situações onde um eixo se deve reger pelo movimento do outro, com um certo fator de escala. Enquanto que a primeira aplicação não está ao alcance das linhas de controladores programáveis em estudo, a segunda situação já se encontra dentro da gama aplicacional destas, através da técnica *master/slave*, para o qual o estudo será orientado.

Deste modo, o presente capítulo aponta as possibilidades e limitações das ferramentas para controlo de movimento no PLC em estudo na área do sincronismo de eixos. De forma a realizar o seu enquadramento nas necessidades industriais, é realizada numa fase inicial uma retrospectiva das causas que levaram à migração da abordagem mecânica inicial para as soluções eletrónicas atuais, seguindo-se a apresentação de alguns exemplos de mecanismos que utilizam a técnica *master/slave* no seu movimento. Por fim, serão apontadas as limitações das ferramentas em estudo nesta área.

3.1 Evolução

Uma das práticas mais patentes na área do sincronismo multi-eixo é a técnica *master/slave* – ou líder/seguidor – a partir da qual é possível obter um sincronismo de movimentos entre um eixo convenientemente especificado como *master*, e os restantes eixos especificados como *slaves*, cujos *setpoints* para movimentação são derivados do movimento do primeiro [14]. Assim, enquanto que o eixo *master* executa um qualquer desejado perfil de movimento descrito em função do tempo, os eixos *slaves* terão o seu perfil de movimento especificado em função da posição/velocidade do eixo *master*.

Os progressos verificados nos sistemas tecnológicos, e associadamente nos sistemas de controlo em malha aberta e em malha fechada, aliados à crescente exigência industrial, levou a que o sincronismo, que inicialmente era concretizado através de uma abordagem exclusivamente mecânica, seja nos dias de hoje fundamentalmente obtido através de variantes eletrónicas. Posto isto, urge neste capítulo apresentar essa abordagem inicial, apontar as devidas falhas e demonstrar como essas questões foram resolvidas com a eletrónica de movimento.

3.1.1 Abordagem Mecânica

Por definição, o sincronismo de eixos assenta numa dada relação entre o movimento do eixo *master* e o movimento dos eixos *slave*. Antes da difusão dos sistemas eletrónicos na área do sincronismo, esta relação era garantida por uma abordagem fundamentalmente mecânica: através de engrenagens era possível obter uma razão constante entre a rotação do veio de um motor e a rotação imposta aos eixos de transmissão secundários, que levariam o movimento ao local pretendido; por sua vez, uma relação não linear de movimentos era originada através de

comes mecânicas, cujo contorno físico induziria um determinado padrão de movimento ao elemento seguidor (Figura 3.1) [15].

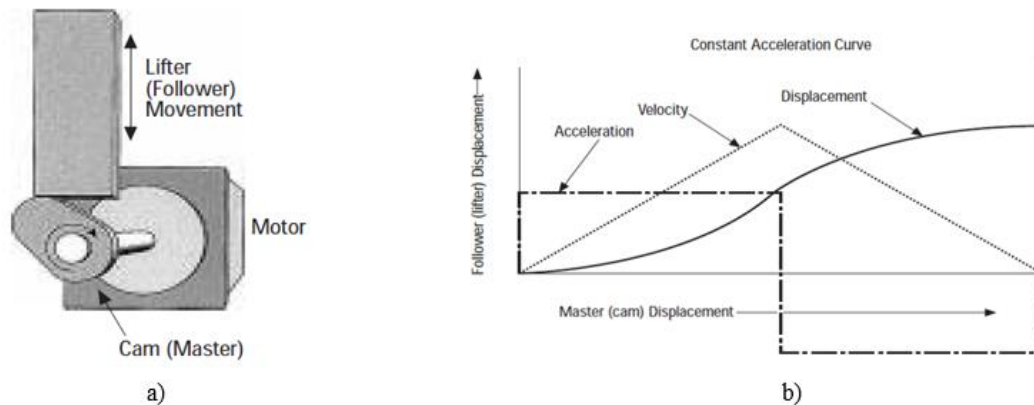


Figura 3.1 – a) Sistema de came mecânica e b) Exemplo de uma relação de movimentos conseguida com este sistema [16].

Apesar da simplicidade destes sistemas mecânicos, é possível identificar imediatamente algumas limitações no desempenho dos mesmos, sobretudo a nível de:

- Flexibilidade de rácios de movimento – a alteração da razão entre movimentos resultante, por exemplo, do processamento de um diferente produto, originaria tempos de inatividade da máquina para substituição das engrenagens/comes.
- Padrão de movimentos complexos – uma relação de movimentos complexa tornaria o *design* do excêntrico da came difícil de obter, o que aumentaria os tempos de projeto, assim como os tempos e custos de produção.
- Desgaste mecânico – o desgaste das comes, assim como os problemas de *backlash* mecânico das engrenagens, enfraqueceria a precisão e a repetibilidade do movimento do eixo seguidor.
- As etapas de arranque e paragem dos eixos seguidores não são controláveis, o que pode originar alguma incerteza nas restantes etapas de movimento.

3.1.2 Abordagem Eletrónica

Um sistema eletrónico de controlo de movimento pode ser, entre outras, constituído por uma das arquiteturas apresentadas no capítulo anterior: um controlador que executa – ou não – algoritmos de controlo de movimento e envia os respetivos *setpoints* para o *drive*; um *drive* que controla a corrente a fornecer aos motores, tendo em conta a posição final requerida; e os motores, que materializam o movimento propriamente dito.

Com as possibilidades introduzidas por estes sistemas, os problemas associados às soluções mecânicas foram, em grande parte, ultrapassadas [14]: as relações de posição e velocidade entre eixos podem ser continuamente ajustadas através do programa do utilizador; relações de movimento complexas são programadas e armazenadas no controlador, eliminando custos de projeto e produção das comes mecânicas; os motores entregam individualmente o movimento no local apropriado, eliminando a necessidade dos eixos de transmissão; redução dos problemas associados ao desgaste mecânico, aumentando a precisão e repetibilidades do movimento dos eixos seguidores; a aceleração e desaceleração são programadas, resultando

num maior controlo por parte do utilizador de todo o movimento associado ao processo. Perante isto, o desempenho da máquina estará menos influenciado por variáveis de perturbação mecânicas, e estará sobretudo dependente da programação, *timing* e capacidade tecnológica do elemento controlador de movimento.

3.2 Exemplos de aplicações

A técnica *master/slave* cobre desde sistemas que englobam rácios de movimento de 1:1, até aplicações que necessitam de relações não lineares entre os mesmos. Os casos mais simples são de facto aqueles cuja relação de movimentos é linear, isto é, de 1:n, comumente referidos como *eletronic gearing*, no quais se podem encontrar, essencialmente, três variantes.

Duas delas dizem respeito à sincronização de movimentos entre eixos diretamente controlados pelo mesmo dispositivo de controlo (por um PLC, por exemplo), sendo que a diferença entre as duas situações está no valor que é utilizado como referência para o(s) eixo(s) *slave*. Numa primeira situação, este valor é dado por um *encoder* – ou outro dispositivo de medição semelhante – associado ao motor *master*, método usualmente utilizado nos equipamentos baseados em configurações Gantry – também conhecidos por tipo-H (Figura 3.2). Nestes, dois motores (1 e 2) movem linearmente a base da máquina, devendo estar sincronizados para prevenir inclinações na estrutura montada transversalmente. Atribuído a um dos motores o papel de *master*, que irá realizar a trajetória de movimento desejada, o motor *slave* terá como *setpoints* os valores de posição lidos pelo *encoder* do primeiro, evitando assim inclinações da estrutura transversal resultantes de perturbações no eixo *master* (Figura 3.3a).

No entanto, é possível apontar algumas limitações a estes sistemas. O primeiro tem a ver com o facto das perturbações ocorridas no eixo *slave* não terem impacto no movimento do eixo *master*. O segundo é o facto dos desvios da trajetória no eixo *master* estarem a ser somados aos desvios já presentes no *loop* de controlo do eixo *slave*, distanciando-se assim ainda mais daquilo que se pretendia como sendo a trajetória para este eixo. Para além disso, erros de quantização na leitura dos valores dos *encoders* resulta em informação perdida e distorcida [16].

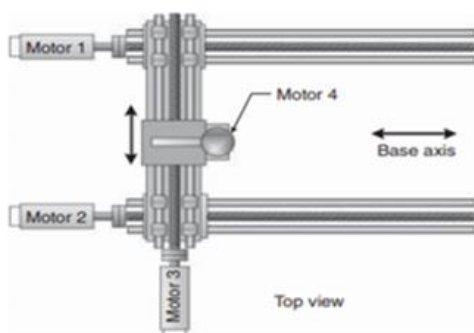


Figura 3.2 – Configuração Gantry [17].

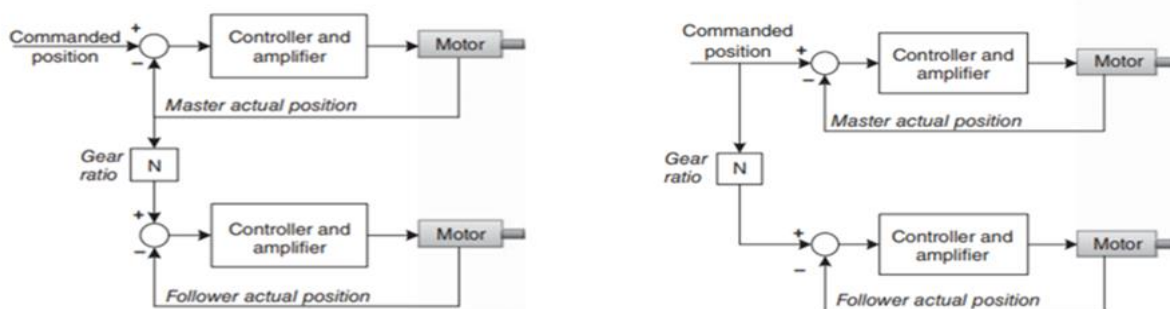


Figura 3.3 – Valor de referência para o anel de controlo proveniente de a) *encoder* associado ao movimento do eixo *master* e b) *setpoint* de posição gerado pelo gerador de trajetória do eixo *master* [17].

Caso o interesse da aplicação esteja mais direcionado para o cumprimento das trajetórias definidas, a segunda solução passa por utilizar como valor de referência para o eixo *slave* o *setpoint* de posição a aplicar no loop de controlo do *master* (Figura 3.3b). No entanto, as perturbações inseridas em cada motor não terão impacto no movimento do motor adjacente.

Uma terceira variante diz respeito a uma situação em que o movimento com o qual se pretende sincronizar não é diretamente influenciado pelo elemento de controlo. Como tal, o valor de referência poderá ser obtido através da leitura fornecida por um *encoder* associado a uma roda de medição que mensura esse movimento.

Já no que diz respeito às relações de movimento não linear, o termo normalmente utilizado é de *eletronic camming*. Exemplos desses sistemas são os equipamentos de corte por medida (Figura 3.4a). Nestes, o eixo *slave* que transporta o equipamento de corte deverá igualar a velocidade do material durante a fase de corte, sendo necessário, no entanto, um período de aceleração até atingir essa velocidade, e um período de desaceleração subsequente. Por fim, o elemento de corte deverá ser recuado até à posição inicial, normalmente a uma velocidade alta, de modo a estar disponível para iniciar um novo ciclo de corte. A relação de movimentos necessária pode ser vista na Figura 3.4b.

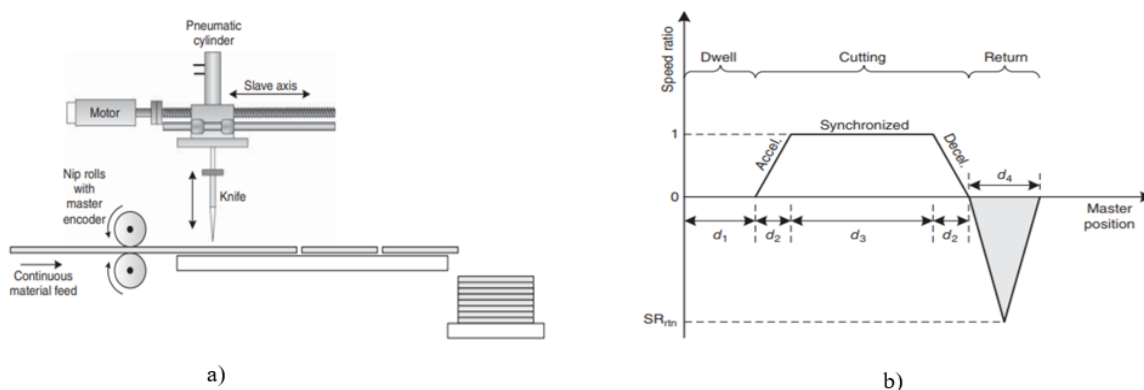


Figura 3.4 - a) Operação de corte por medida denominada por *Flying Knife* e b) relação de movimentos inerentes a esta operação [17].

3.3 Soluções de Sincronismo com o PLC S7-1500

Anteriormente, já foi referida a incapacidade do S7-1200 em trabalhar com o objeto tecnológico de sincronismo de movimento, uma vez que não possui funcionalidades para este tipo de controlo de movimento na sua CPU. Assim, este capítulo irá centrar-se no sincronismo de movimento concretizado com a família de controladores S7-1500.

Para a realização de operações de sincronismo, a PLCOpen propõe as seguintes instruções de movimento:

- **MC_GearIn** – dá início a uma operação de sincronização entre um eixo *master* e um eixo *slave* (Figura 3.5a). Após a ordem de execução deste bloco, o eixo *slave* acelera (ou desacelera) com os valores indicados na interface do bloco, tendo em vista a atingir a relação de movimentos requerida.

- MC_GearInPos – esta difere da anterior no sentido em que se pretende que os eixos estejam já sincronizados nas posições especificadas na interface deste bloco (Figura 3.5b). Para tal, a sua ordem de execução deverá ser iniciada atempadamente, de modo a que o controlador possa calcular o perfil de movimento a aplicar ao eixo *slave* necessário para que este atinja a posição especificada para si ao mesmo tempo que o eixo *master* atinge a sua posição.
- MC_PhasingAbsolute – com este bloco de função é alterada a fase de movimentos entre o eixo *slave* e o eixo *master* em operação síncrona. O valor a introduzir pode ser absoluto (Figura 3.5c), ou então poderá ser adicionado à fase já existente – MC_PhasingRelative (Figura 3.5d).
- MC_CamIn – inicia uma operação de sincronização com relação não linear entre os eixos (Figura 3.5e), cuja configuração é realizada num objeto tecnológico para o efeito – TO_Cam.



Figura 3.5 – Instruções de movimentos: a)MC_GearIn, b)MC_GearInPos, c)MC_PhasingRelative, d)MC_PhasingAbsolute e e)MC_CamIn.

No entanto, nem todas estas instruções estão disponíveis nos S7-1500 – apenas as CPUs das séries tecnológicas das Siemens (T-CPU) permitem uma utilização completa destas. Estes controladores são um pouco sobredimensionados, e necessariamente de custo excessivo, para a gama das aplicações em estudo.

De entre as instruções especificadas, os controladores da família S7-1500 apenas disponibilizam o MC_GearIn. Mas as limitações não ficam por aqui - se tivermos em conta o

valor de referência para o *setpoint* a aplicar ao eixo *slave*, somente se poderá utilizar o *setpoint* de posição aplicado no eixo *master* – impossibilitando assim o uso de valores atuais de posição do eixo *master*, ou ainda a referência dada por *encoders* externos.

Contudo, após pesquisa aprofundada em tópicos de discussão e outros similares, foi possível descobrir um *workaround* interessante, que permite sincronizar o eixo *slave* com os valores lidos por um *encoder* externo [17], ultrapassando-se assim a limitação imposta pela Siemens no TIA Portal na fase de configuração do objeto tecnológico.

Terminado o levantamento das possibilidades e limitações deste controlador na área do sincronismo de movimento, poder-se-á avançar para a implementação a concretizar. Contudo, será necessário realizar previamente uma descrição detalhada da aplicação em análise, algo que será realizado no capítulo que se segue.

3.4 Síntese

O capítulo que aqui finda concentrou-se nas possibilidades a nível de sincronismo de movimento com o S7-1500, e que serão importantes para delinear a estratégia de implementação a realizar nos próximos capítulos.

Numa primeira fase, e de modo a enquadrar as necessidades de sincronismo de movimento no ambiente industrial, foram apresentados os motivos que levaram à difusão de soluções elétricas nesta área. Com estas, a performance da máquina fica maioritariamente dependente do programa e capacidade do *hardware* em desempenhar funções de movimentos complexos, minimizando-se a dependência com outras variáveis muito pouco controláveis.

De seguida levantaram-se diversas questões inerentes à técnica *master/slave*, com a apresentação paralela de algumas das aplicações que utilizam esta técnica para atingir o sincronismo de movimento entre os seus eixos. Estas serviram para abrir o livro sobre as diversas variantes de controlo existentes, que no entanto foram imediatamente limitadas no capítulo seguinte pelas possibilidades do controlador em análise.

4 Descrição Detalhada da Aplicação em Análise

Este capítulo descreve de forma exaustiva os principais requisitos de funcionamento da unidade de pultrusão. Inicialmente é apresentada uma visão geral do processo de fabrico em causa, tanto a nível de operacionalidade, como a nível de aplicabilidade, seguindo-se uma apresentação das funcionalidades objeto de análise. Posteriormente são expostas as particularidades do projeto, concluindo-se com a aceção das adaptações a realizar.

4.1 Unidade de Pultrusão

O processo de pultrusão é um processo de fabrico de materiais compósitos, destacado pela sua automatização na produção em contínuo de perfis de secção transversal constante [18, 19]. A nível de funcionamento, este pode ser dividido em três estágios diferentes, tendo cabido à Gislotica a intervenção direta em dois destes. Sendo assim, o que se propõe neste capítulo é, numa primeira fase, apresentar os principais aspetos aliados a todas as etapas do processo, para, numa segunda fase, serem aprofundados as particularidades essenciais das etapas alvo de estudo por parte da Gislotica, com a consequente documentação das funcionalidades a implementar por parte do autor.

Uma vez que esta unidade não se encontra atualmente dentro das instalações da empresa, a interpretação do seu funcionamento foi realizada recorrendo ao manual do equipamento.

4.1.1 Etapas do processo de fabrico

A pultrusão tem como especificidade de produção a capacidade de converter matrizes de resina e reforços de fibra em perfis para fins estruturais e decorativos. Assim, numa primeira etapa, as fibras – tipicamente de vidro – dispostas em bobines/*rovings*, são conduzidas a um banho de resina, previamente formulado de acordo com as necessidades do produto final (Figura 4.1) [20].

Após a fase de impregnação da resina nas fibras de reforço, as fibras impregnadas são guiadas até uma fieira ou molde através de umas placas que lhes vão pré-definindo uma determinada forma desejada, ao mesmo tempo que possibilitam a retirada de excedentes de resina. Já no molde, as fibras sofrem o processo de polimerização, através de um sistema de aquecimento por resistências.

O avanço do material pultrudido é assegurado por um sistema de *pulling* – que dá o nome ao processo (*pull* e extrusão) – e que garante uma velocidade constante ao material. Esta velocidade deverá, entre outros aspetos, garantir um tempo de permanência dentro da fieira – ou molde – suficiente para que o processo de cura se dê por completo, de modo a que o material adquira a resistência necessária para resistir às forças exercidas durante a fase de tração.

Por último, um módulo de corte é responsável pelo corte do produto final segundo o comprimento especificado. Como imposição, este sistema deverá adquirir a velocidade do

material antes do início da realização do corte, para que este se realize corretamente e sem paragens no processo de tração.

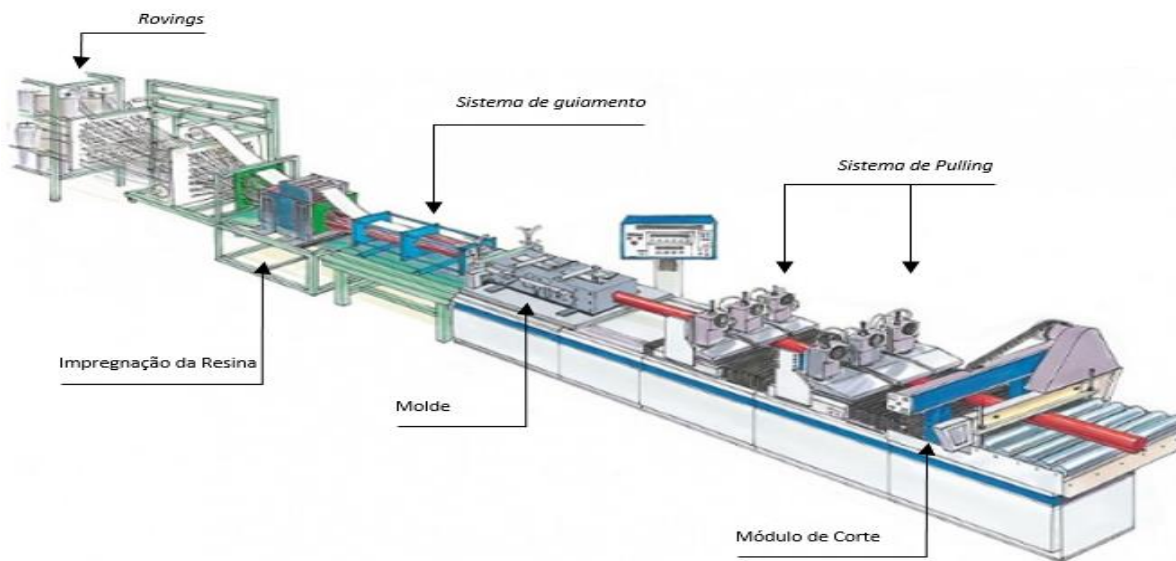


Figura 4.1 – Representação esquemática de uma unidade de pultrusão [21].

4.1.2 Etapas-Alvo

A nível da Gislotica, a produção incidiu essencialmente sobre os mecanismos de *pulling* e de corte (Figura 4.2), tendo sido o mecanismo de impregnação da resina nas fibras de reforço realizado fora das instalações da empresa.

O sistema de *pulling* é constituído por dois carros de arrasto – também designados por *charriots* – responsáveis por impor a velocidade de produção do perfil que, por especificação do processo, deverá manter-se constante. Esta velocidade é obtida experimentalmente, e ronda tipicamente a gama das unidades de metro por minuto, o que é considerado como uma baixa velocidade de trabalho, mas que é necessária para garantir as condições de cura apropriadas.

A chave para concretizar a constância de velocidade necessária está, obviamente, no modo como estão interligados os movimentos dos carros de arrasto. De um modo geral, estes deverão alternar como elemento movimentador do perfil, sendo imprescindível existir um período de sincronismo de movimento, num fator de 1:1, durante a transição, para prevenir qualquer imprevisto que possa induzir uma alteração de velocidade do material, o que colocaria em causa a integridade estrutural do material já curado e do material em processo de cura.

A tração do material por parte do sistema de *pulling* é garantido por maxilas, que acionadas pneumáticamente garantem o atraque ao material. Estas não deverão entrar em contacto com o material caso o respetivo carro não possua a velocidade do pultrudido, sob pena de induzir marcas permanentes na peça final. Importa também referir que, na mudança de perfil, caso se verifique que este tem uma altura diferente da do perfil anterior, as maxilas são ajustadas no sentido de garantir sempre o mesmo curso de trabalho.

No que diz respeito ao módulo de corte, este é constituído por dois carros, um que acompanha o movimento do material e desloca todo o conjunto de corte – *charriot 3*, e um outro, o carro porta-serra, que transporta e desloca a serra numa direção transversal ao

movimento do produto. Para além disso, a estabilidade do corte é garantido por maxilas, que apertam o material e tracionam-no.

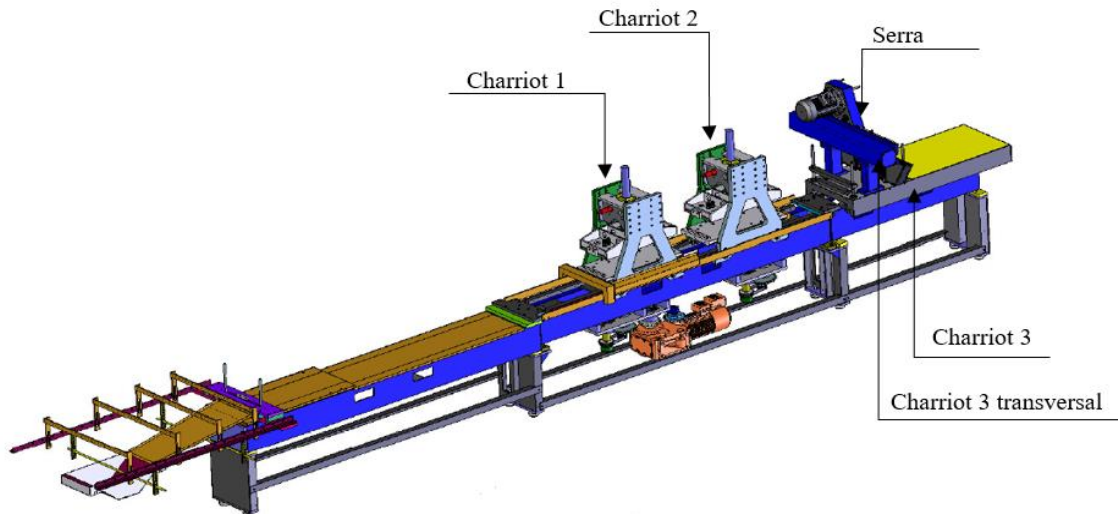


Figura 4.2 – Desenho de conjunto presente no manual do equipamento.

A implementação realizada incidiu essencialmente sobre os ciclos de funcionamento associados ao sistema de *pulling* e ao módulo de corte. No entanto, é necessário realizar um levantamento do *hardware* da solução inicial para se concluir sobre as principais adaptações a realizar. Nesse âmbito, a próxima secção aponta algumas dessas características.

4.2 Hardware Inicial

4.2.1 Controlo e acionamento

O controlador programável presente no equipamento é um controlador da família CX de PCs embebidos da Beckhoff [21].

A nível de acionamento é possível verificar que o movimento dos carros do sistema de *pulling* são acionados por servomotores assíncronos [22], com *encoders* incrementais integrados, controlados por *drives* da série Multiaxis [23]. Em termos de acionamentos eletromecânicos, ainda é possível de destacar que o movimento do carro transversal e da serra é provocado por um motor de indução trifásico [24], com variador de frequência da série Movitrac.

No que diz respeito ao acionamento do carro que transporta os módulos de corte, este é realizado pneumáticamente, sendo calibrado para uma velocidade superior à velocidade requerida para o material. Como tem baixo binário, o seu contacto com o material não irá provocar-lhe danos.

4.2.2 Sensores Magnéticos

A referência entre os valores de posição fornecidos pelos *encoders* incrementais e o sistema físico em si é efetivada devido à presença de sensores magnéticos no curso dos carros do sistema de *pulling*.

Por seu lado, sensores magnéticos presentes no curso do carro transversal têm como função impor o seu curso de funcionamento.

Ainda existem dois sensores magnéticos que indicam o avanço e recuo do cilindro de acionamento do *charriot* 3.

4.2.3 Segurança

Como em qualquer outro equipamento industrial, a integridade, quer física ao nível do operador, quer estrutural ao nível da máquina, é prioridade bem assente. Essa importância tem uma magnitude tal, que se reflete na obrigação do cumprimento de um determinado conjunto de normas bem definidas, cujo incumprimento impede o lançamento do equipamento no mercado por parte do fabricante. Fala-se, claro, das normas estipuladas na Diretiva Máquina (“Diretiva 2006/42/CE”).

De entre os dispositivos de segurança presentes na unidade de pultrusão, destacam-se os fins-de-curso mecânicos, que determinam as posições limites a partir das quais os carros poderão provocar danos à estrutura, e as botoneiras de emergência, que serão atuadas pelo operador em situações de risco. Os primeiros são colocados a uma distância que permita a paragem do motor antes de qualquer contacto crítico com a estrutura da máquina, e os segundos têm os seus requisitos funcionais e princípios de conceção especificados na norma EN ISO 13850.

O desencadeamento das reações de emergência devido à atuação de qualquer um dos elementos anteriormente referidos é iniciado através de um relé de segurança [25], que atua diretamente sobre a válvula de corte geral do ar presente na unidade de tratamento de ar, sobre os contactores presentes tanto na linha do módulo de alimentação dos *drives* dos carros de *pulling*, como na alimentação do variador de frequência, e sobre a alimentação dos módulos de saídas do PLC, o que desarmará o contactor da serra. É de realçar que, como em qualquer comum relé de segurança, após o seu desarme, devido à atuação de um dos dispositivos de segurança, e caso estes voltem à sua condição de desatuados, é necessário proceder ao rearme do relé de segurança.

Ainda no que concerne a questões de segurança, encontra-se implementado que enquanto o PLC não arrancar, a linha de segurança, correspondente ao relé de segurança, não será habilitada, o que impede o arranque da máquina.

Para efeitos de tratamento de informação por parte do PLC, de modo a influenciar a evolução dos ciclos de forma apropriada, é sinalizado a este a atuação das botoneiras de emergência, a atuação dos fins-de-curso referidos anteriormente e o estado do relé de segurança – armado ou desarmado. É também indicado ao PLC o estado dos disjuntores de alimentação dos motores, assim como dos disjuntores presentes na linha de alimentação das resistências de aquecimento. Falhas na alimentação do ar, sinalizadas por um pressóstato, e erros nos *drives* dos motores são, também, alvo de análise por parte do controlador programável.

Ainda no que diz respeito à proteção de equipamentos, mas neste caso respeitante a componentes elétricos, é interessante verificar a presença de uma resistência encarregue de dissipar a energia da linha de potência, caso esta perca capacidade de regeneração de energia para a rede elétrica, em períodos de travagem ou paragem dos motores. Dito isto, importa

sinalizar ao PLC o estado do disjuntor da linha da resistência, de modo a que se impeça o funcionamento da máquina caso ocorra o seu desarme.

4.2.4 Aquecimento

A lógica aplicada ao aquecimento do molde foi também incorporada nas funcionalidades do PLC. Assim, de modo a conceber uma solução que permita que a curva de temperaturas dentro do molde se situe num intervalo o mais pequeno possível em torno do valor de *setpoint* de temperatura requerido, é necessário ir conhecendo a temperatura atual das zonas de aquecimento. Desta forma, quatro termopares, respeitantes a quatro zonas de aquecimento, permitem a leitura de temperatura necessária, e irão permitir ligar e desligar a alimentação das resistências de aquecimento.

4.3 Alterações a efetuar a nível de *hardware* e respetiva justificação

Uma vez apresentadas as diversas particularidades a ter em conta na elaboração dos ciclos de funcionamento, proceder-se-á de seguida à apresentação de algumas das mudanças a realizar a nível de *hardware*.

4.3.1 Acionamento

A nível de acionamento, foi proposto que os *drives* dos motores dos *charriots* 1 e 2 seriam operados em servo controlo, concretizado por servomotores da Siemens, equipados com *encoders* absolutos e interface *Drive-Cliq*, assim como o motor do *charriot* 3, abandonando-se a solução pneumática inicial. No que diz respeito ao movimento do carro transversal e à rotação da serra, um controlo em V/f é aplicado.

Assim, resulta que a constituição do *drive* S120 ficaria:

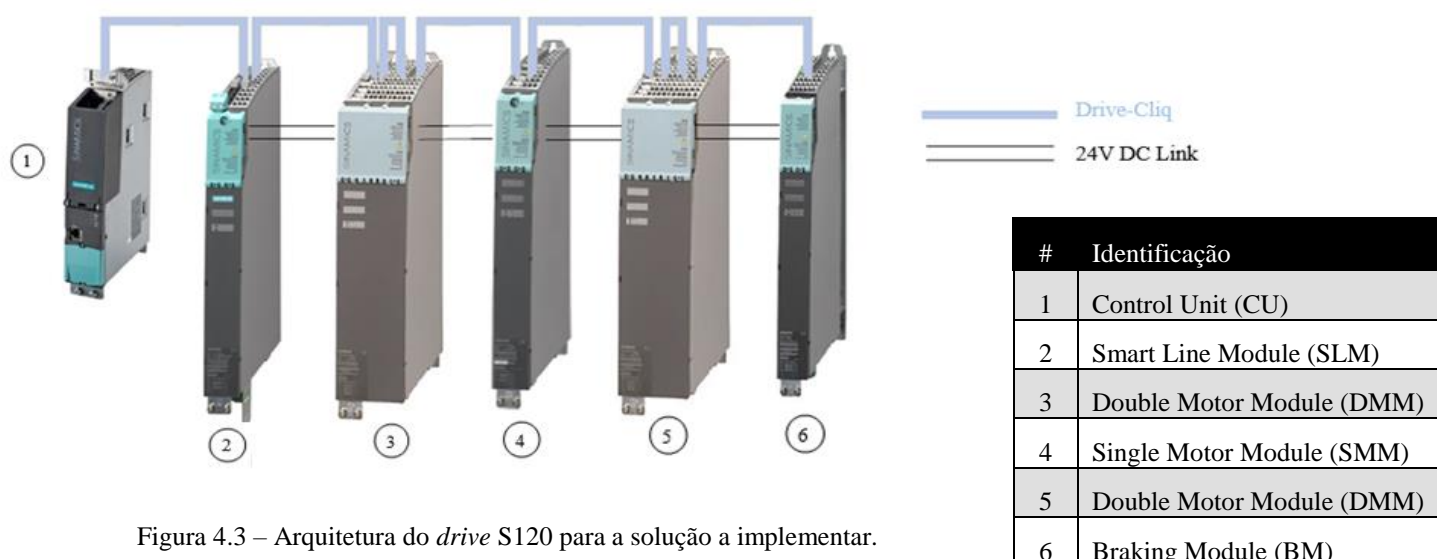


Figura 4.3 – Arquitetura do *drive* S120 para a solução a implementar.

Para além do DMM associado aos motores dos *charriots* 1 e 2, do SMM correspondente ao *charriot* 3, do DMM relacionado com os motores da serra e do carro transversal, e, claro, da *Control Unit*, é possível distinguir ainda os seguintes módulos:

- *SLM* – componente elétrico responsável por gerar e disponibilizar tensão DC para toda a linha, a partir da alimentação trifásica. Para além disso, é capaz de recuperar a energia regenerativa para a rede elétrica, criada durante a paragem ou travagem dos motores [26].
- *Braking Module* – com uma resistência de dissipação associada, tem como função garantir a proteção dos equipamentos elétricos da linha caso o SLM perca capacidade em lidar com a energia regenerativa, ou quando esta é superior à capacidade de regeneração do SLM. Ademais, garantem uma travagem controlada do motor em caso de falha elétrica na linha. Ainda em relação à energia regenerativa, uma vez que partilham o mesmo *DC Link*, os módulos podem trocar energia uns com os outros – assim, numa situação em que, por exemplo, um motor esteja a funcionar como gerador, a energia gerada pode ser usada por outro motor que não se encontre a operar em modo gerador. Uma possível falha deste módulo, assim como de qualquer outro módulo desta família, é comunicada à *Control Unit* através do *Drive-Cliq* [26], que por sua vez a sinaliza ao PLC.

4.3.2 Componentes mecânicos

Para além das mudanças a nível de acionamento, também se optou por adicionar fins-de-curso mecânicos com o objetivo de informar acerca do recuo das maxilas associados aos *charriots* do sistema de *pulling*. Isto permitirá realizar o recuo de cada um dos *charriots* em segurança, uma vez que será garantido que as maxilas já não estarão em contacto com o material.

Para realizar o corte com uma precisão mais elevada, uma roda de medição estará em contacto com o material. Associada a essa roda, um *encoder* de 5 V dará os valores necessários a ter em conta na operação do corte.

4.4 Especificação justificada dos modos de funcionamento / operação

Como referido anteriormente, a pultrusão é um processo de fabrico de materiais em contínuo. Nesse sentido, durante o seu funcionamento, a movimentação das partes móveis da máquina será restritamente automática, isto é, sem qualquer intervenção por parte do operador. No entanto, é necessário contemplar a situação em que, por um qualquer motivo, o material se tenha deformado e o operador pretenda cortar imediatamente o segmento de material deformado, de modo a não prejudicar a restante produção e minimizar desperdícios. Nesse sentido, dever-se-á disponibilizar na HMI do processo a possibilidade de realizar um corte imediato ao operador.

Outra questão a contemplar tem a ver com as fases de manutenção do equipamento. Assim, deverá ser concebida a possibilidade ao operador de mover de forma independente as várias partes móveis da unidade de pultrusão. Para além disso, este modo manual perfaz uma mais-valia para as fases de teste.

A construção de todo o programa tem que ter em conta as funcionalidades que vai ser possível dar ao operador. Assim, durante a parte operacional do processo – isto é, sem erros – o operador deverá ter como possíveis interações no processo as seguintes:

- ✓ Seleção do modo de funcionamento da máquina: automático ou manual. Esta será realizada através de um seletor, e não na HMI, como acontecia na solução inicial.
- ✓ No modo automático

- Ordem de início (Start) e paragem (Stop) do ciclo de *pulling*
 - Ordens de início e paragem do ciclo de corte (Start Cut e Stop Cut)
 - Ordem de corte imediato (Do Cut)
 - Reinício da contagem do número de peças cortadas de forma automática (Reset)
- ✓ No modo manual
- *Jogging* dos componentes móveis lineares – *charriot 1*, *charriot 2*, *charriot 3* e *charriot 3T*
 - Rotação da Serra
- ✓ Realizar a calibração dos encoders absolutos do *charriot 1*, *charriot 2* e *charriot 3*.
- ✓ Definição do valor de *setpoint* de velocidade, comprimento do material a cortar, velocidades de rotação da serra, velocidade de translação do carro transversal.
- ✓ Visualização do número de peças cortadas de forma automática.

4.5 Síntese

O capítulo que aqui termina expôs as principais particularidades sobre o projeto em estudo. Inicialmente, foi apresentada uma visão geral do processo de fabrico de pultrusão, que permitiu enquadrar o mesmo no âmbito industrial, seguindo-se uma análise dos principais dinamismos da aplicação.

De seguida, foi realizado um estado da arte do *hardware* implementado na solução inicial, e que permitiu posteriormente apontar as adaptações a realizar.

Por fim, foram especificados os modos de funcionamento e operação a aplicar no próximo capítulo.

5 Caso de estudo

O presente capítulo relata a parte experimental desenvolvida durante a dissertação. O grande objetivo passou pela aplicação das ferramentas da Siemens para controlo de movimento no PLC na programação das rotinas de funcionamento da unidade de pultrusão descrita no capítulo anterior.

Numa primeira fase são expostas as opções tomadas a nível do *hardware* considerado, seguindo-se a apresentação e exploração dos ciclos-máquina desenvolvidos para os mecanismos de *pulling* e de corte. Posteriormente serão apresentados os ecrãs desenvolvidos para a interface homem-máquina (HMI), sendo realizado paralelamente (quando considerado conveniente) a explicação de segmentos do programa realizado. Esta opção de estruturação tem como principal objetivo evitar a repetição de conceitos ao longo da descrição da implementação.

Por fim, serão apontados os testes realizados para validação das rotinas programadas.

5.1 Componentes Siemens usados no projeto

Neste ponto são apresentadas as razões que levaram à escolha do *hardware* considerado. Tendo já algumas dessas opções sido justificadas no capítulo anterior, os objetos alvo de análise serão o controlador lógico programável, os módulos de I/O selecionados, e a HMI escolhida.

5.1.1 PLC

Tendo em conta as conclusões retiradas dos capítulos anteriores, não é surpresa que o elemento de controlo selecionado pertença à família dos S7-1500 da Siemens. Concretamente, para a emulação dos ciclos de funcionamento, a opção no TIA Portal recaiu sobre o S7-1500 CPU 1517-3 PN/DP (Figura 5.1) [27]. A nível de tempos de processamento, consegue realizar operações com bits em 2 ns e com words em 3 ns [28], conseguindo ainda, por exemplo, realizar o processamento dos algoritmos de seis objetos tecnológicos de posicionamento em cerca de 1 ms [29].



Figura 5.1 – Controlador lógico programável S7-1500 CPU 1517-3 PN/DP [28].

5.1.2 Módulos I/O

Os módulos de I/O realizam a interface entre o processo e o respetivo controlador do processo. Nesse sentido, o SIMATIC ET200SP (Figura 5.2a) apresenta-se como uma solução altamente modular para realizar a ligação entre os dispositivos de campo e o controlador S7-1500, permitindo a qualquer momento a integração de uma nova carta de I/O.

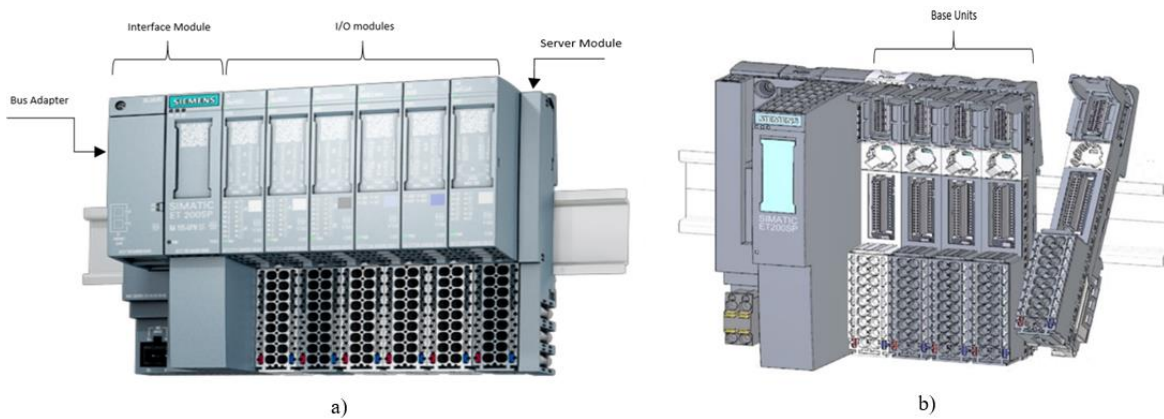


Figura 5.2 – a) Módulos da estação ET200SP [24] e b) Pormenor das Base Units [25].

Uma estação ET200SP é constituída normalmente por uma *Interface Module* (IM), módulos I/O e um *Server Module* [30]:

- ✓ A *Interface Module*, como o próprio nome indica, trata de realizar a interface entre o controlador do processo e a estação. Para tal, destaca-se o *Bus Adapter* (BAs), responsável pela comunicação PROFINET com o controlador.
- ✓ Os módulos de entradas e saídas permitem a troca de dados entre a estação e os dispositivos de campo.
- ✓ O *Server Module* delimita física e eletricamente a estação – normalmente também conhecido por “cabeceira”.

Para a montagem dos módulos da estação ao *rail* são utilizadas *Base Units* (Figura 5.2b). Estas permitem não só a conexão mecânica, como elétrica entre módulos [31].

Um dos pormenores que este sistema de I/O distribuídos apresenta e que se afigura como uma vantagem em relação às soluções semelhantes – ET200S e ET200M – é a incorporação da funcionalidade de alimentação dos restantes módulos da estação no sistema, o que reduz a necessidade de ter um módulo extra responsável por esta função – o *Power Module*. Assim, esta solução é apresentada pela SIEMENS como cerca de 50 % mais compacta do que as soluções semelhantes [32].

Na Tabela 5.1 especificam-se as referências dos módulos selecionados. De entre estes, destaca-se a carta responsável pela leitura dos valores de posição da roda de medição, fornecidos por um *encoder* incremental de 5 V, e a carta responsável pela leitura dos valores de temperatura dados pelos termopares [33, 34]. Por sua vez, na Figura 5.3 encontra-se a emulação dos mesmos no *software* TIA Portal.



Figura 5.4 – a) SIMATIC HMI KTP900F [36] e b) Connection Box [37].

Tabela 5.2 – Características funcionais da HMI KTP900F [36]

Ecrã	TFT widescreen display 9 polegadas 16 777 216 cores Resolução 800x480 Pixels
Elementos de controlo	10 teclas funcionais configuráveis Teclados numérico e alfanumérico no ecrã
Tipo de processador	ARM
Alimentação	24 VDC
Memória	12 Mbyte

5.2 Ciclos Automáticos de Funcionamento

Para além das variáveis analógicas associadas à leitura dos termopares existentes nas quatro zonas de aquecimento, assim como dos valores provenientes da leitura de posição da roda de medição, é possível ainda distinguir as diferentes variáveis de entrada e saída associados aos módulos de I/O especificados anteriormente, e que se encontram disponíveis nas tabelas 5.3 e 5.4.

Para além destas, variáveis adicionais necessárias para a compreensão dos ciclos de corte e *pulling* que se seguem encontram-se disponibilizadas na Tabela 5.5.

Tabela 5.3 – Variáveis de entrada

Entrada	Variável	Significado quando TRUE
I0.0	Relé de segurança	Relé de segurança desarmado
I0.1	Potência	Não existe falha na potência
I0.2	Potência de aquecimento	Não existe falha na potência de aquecimento
I0.3	Falha no ar	Não existe falha na alimentação do ar
I0.4	Botoneira de emergência principal	Botoneira de emergência principal atuada
I0.5	Botoneira de emergência HMI	Botoneira de emergência HMI atuada
I0.6	Botoneira de emergência corte	Botoneira de emergência na zona de corte atuada
I1.0	Seletor manual	Seletor em posição “manual”
I1.1	Seletor auto	Seletor em posição “automático”
I1.2	Fim-de-curso maxilas Ch1	Maxilas Ch1 recuadas
I1.3	Fim-de-curso maxilas Ch2	Maxilas Ch2 recuadas
I1.4	Fim-de-curso Ch1 atrás	Ch1 atinge posição limite a partir da qual provoca danos no equipamento
I1.5	Fim-de-curso Ch1 frente	Ch1 atinge posição limite a partir da qual provoca danos no equipamento
I1.6	Fim-de-curso Ch2 atrás	Ch2 atinge posição limite a partir da qual provoca danos no equipamento
I1.7	Fim-de-curso Ch2 frente	Ch2 atinge posição limite a partir da qual provoca danos no equipamento
I2.1	Fim-de-curso Ch3 atrás	Ch3 atinge posição limite a partir da qual provoca danos no equipamento
I2.2	Fim-de-curso Ch3 frente	Ch3 atinge posição limite a partir da qual provoca danos no equipamento
I2.3	Fim-de-curso Ch3T atrás	Ch3T atinge posição limite a partir da qual provoca danos no equipamento
I2.4	Fim-curso Ch3T frente	Ch3T atinge posição limite a partir da qual provoca danos no equipamento
I2.5	Sensor Ch3T atrás	Ch3T na posição inicial de curso
I2.6	Sensor Ch3T frente	Ch3T na posição final de curso

Tabela 5.4 – Variáveis de saída

Saída	Variável	Significado quando TRUE
Q1.0	Maxilas Ch1	Atuação das maxilas associadas ao <i>Charriot 1</i>
Q1.1	Maxilas Ch2	Atuação das maxilas associadas ao <i>Charriot 2</i>
Q1.2	Maxilas Ch3	Atuação das maxilas associadas ao <i>Charriot 3</i>
Q1.3	Resistências zona 1	Alimentação das resistências de aquecimento associada à zona 1
Q1.4	Resistências zona 2	Alimentação das resistências de aquecimento associada à zona 2
Q1.5	Resistências zona 3	Alimentação das resistências de aquecimento associada à zona 3
Q1.6	Resistências zona 4	Alimentação das resistências de aquecimento associada à zona 4

Tabela 5.5 – Variáveis adicionais

Variável	Significado quando TRUE
Start	Atuação do botão Start da HMI
Stop	Atuação do botão Stop da HMI
Start Corte	Atuação do botão Start Cut da HMI
Stop Corte	Atuação do botão Stop Cut da HMI
Do Cut	Ordem de corte imediato dada pelo operador na HMI
Reset	Reset do contador de peças automáticas
<i>Acknowledge</i> do erro	<i>Acknowledge</i> do erro na HMI por parte do operador
Ciclo de corte	Ativa ciclo de corte – é possível realizar o corte
Mover Ch1 para Posição Inicial	Ordem de execução para posicionamento absoluto do <i>Charriot 1</i> na sua posição inicial de ciclo automático
Mover Ch2 para Posição Inicial	Ordem de execução para posicionamento absoluto do <i>Charriot 2</i> na sua posição inicial de ciclo automático
Mover Ch3 para Posição Inicial	Ordem de execução para posicionamento absoluto do <i>Charriot 3</i> na sua posição inicial de ciclo automático
Mover Ch3T para Posição Inicial	Ordem de execução para movimento do <i>Charriot 3T</i> até atingir a sua posição inicial de ciclo automático
Avanço do Ch1 à velocidade de Setpoint	Ordem de execução para avanço do <i>Charriot 1</i> , para a posição target e com a velocidade de <i>setpoint</i> requerida
Avanço do Ch2 à velocidade de Setpoint	Ordem de execução para avanço do <i>Charriot 2</i> , para a posição target e com a velocidade de <i>setpoint</i> requerida
Sincronizar M1S2	Ordem de execução para “engrenar” os movimentos do <i>Charriot 1</i> e do <i>Charriot 2</i> , com o <i>Charriot 1</i> sendo o eixo <i>master</i>
Sincronizar M2S1	Ordem de execução para “engrenar” os movimentos do <i>Charriot 1</i> e do <i>Charriot 2</i> , com o <i>Charriot 2</i> sendo o eixo <i>master</i>
Erro de supervisão_1 (e_2)	Erro referente a uma especificação de velocidade demasiado elevada
M	Corte não automático
Mover Ch3 para Posição Inicial	Ordem de execução para posicionamento absoluto do <i>Charriot 3</i> na sua posição inicial de ciclo automático
Mover Ch3 para Posição Inicial	Ordem de execução para posicionamento absoluto do <i>Charriot 3</i> na sua posição inicial de ciclo automático
Avanço do Ch3T	Ordem de execução para avanço do <i>Charriot 3T</i>
Serra	Ordem de execução para rotação da serra
Sincronizar Ch3 com o material	Ordem de execução para “engrenar” o movimento do <i>Charriot 3</i> com o movimento do material
Mover Ch3T para Posição Inicial	Ordem de execução para movimento do <i>Charriot 3T</i> até atingir a sua posição inicial de ciclo automático
Erro de supervisão_3 (e_4)	Erro referente ao contacto defeituoso da roda de medição com o material

Para a programação da sequência de movimentos associado aos ciclos de *pulling* e de corte, recorreu-se à ferramenta de estruturação SIMATIC S7-GRAPH da Siemens. Sendo um método de programação com um *background* gráfico bastante intuitivo, permite não só facilitar a fase de manutenção por parte do cliente final, como permite um rastreio mais rápido de erros durante a fase de teste do programa.

O seu funcionamento assemelha-se ao da linguagem SFC (*Sequential Function Chart*) especificada na norma IEC 61131-3 [39]: sequências de movimento são materializadas em evoluções por etapas, após a validação das recetividades presentes nas respetivas transições. No entanto, o S7-GRAPH apresenta algumas peculiaridades dignas de registo. Uma delas prende-se com a possibilidade de associar a cada etapa uma instrução de *Interlock*, responsável por condicionar a execução de determinadas ações especificadas dentro da respetiva etapa; de modo semelhante, para a monitorização de variáveis durante o estado ativo de uma etapa é possibilitada a criação de uma condição de *Supervision*, que caso se verifique congela a evolução da sequência, assinalando um erro ao utilizador.

Antes da apresentação de qualquer programação realizada, importa também especificar os objetos tecnológicos atribuídos. Posteriormente, serão apresentadas e descritas as sequências de movimentos desenvolvidas para os ciclos de *pulling* e corte. Contudo, uma vez que a transcrição dos modelos de implementação e respetiva lógica dificultaria a interpretação por parte do leitor, serão apresentados os modelos comportamentais (definidos de acordo com a 2ª edição do Grafset [40]) que serviram como base para a implementação em si. Não obstante, algumas instruções associadas ao controlo de movimento serão também mencionadas.

5.2.1 Definição dos objetos tecnológicos utilizados

Os *charriots* do sistema de *pulling* (1 e 2) vão ser materializados em objetos tecnológicos de sincronismo, assim como o *charriot* 3, uma vez que irá entrar em sincronismo com o material no período de corte.

Por sua vez, o *charriot* transversal e a serra serão materializados em objetos tecnológicos de velocidade.

Por último, os valores do encoder associado à roda de medição serão obtidos através do objeto tecnológico de *encoder* externo (TO_ExternalEncoder).

5.2.2 Ciclo de *Pulling*

A leitura desta secção deverá ser acompanhada da visualização do modelo comportamental representado na Figura 5.5. Por sua vez, na Figura 5.6 estão representadas algumas das instruções de movimento utilizadas, e que serão referidas ao longo do texto explicativo dos principais pontos do modelo comportamental. Adicionalmente são realizados alguns comentários referentes aos blocos de função dessas instruções de movimento na Tabela 5.6.

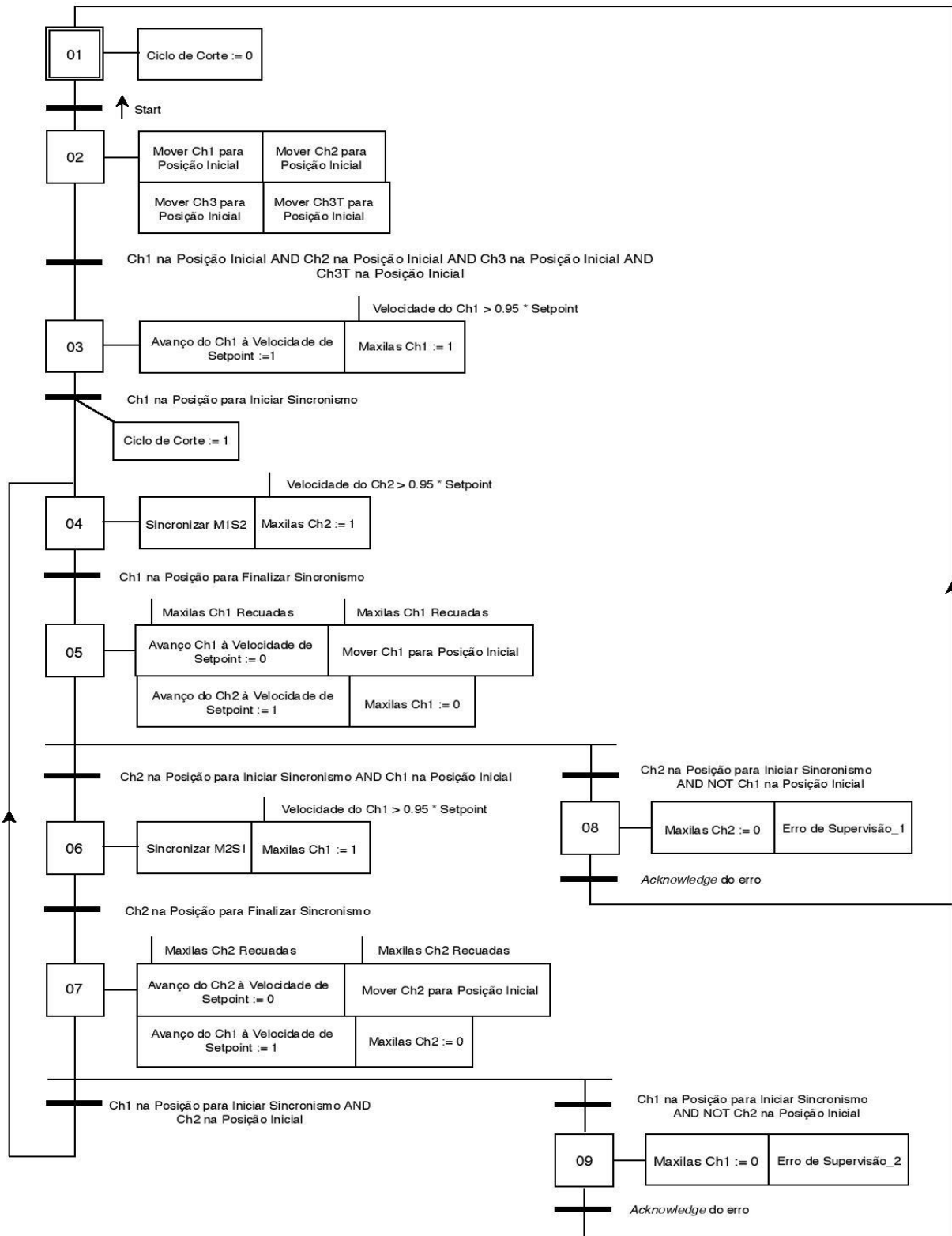


Figura 5.5 – Modelo comportamental do ciclo de *pulling*.

Dada a sequência de movimentos necessária à execução do processo automático da unidade de pultrusão, e uma vez que o ciclo de *pulling* figura necessariamente como o primeiro ciclo de trabalho automático, optou-se então por incluir nesta sequência o movimento das partes móveis lineares da máquina para a sua posição inicial – posição de partida para a sequência de movimentos. A ordem para o movimento dos *charriots* 1, 2 e 3, é dada pela instrução para posicionamento absoluto na posição zero (Figura 5.6a) – o que obriga a que os objetos tecnológicos correspondentes estejam em condição de “*homed*”, isto é, que os seus *encoders* tenham sido calibrados previamente. Já no que diz respeito ao *charriot* 3T é necessário levar o objeto tecnológico até ao sensor de posição inicial, usando para isso a instrução MC_MoveJog (Figura 5.6b), a uma velocidade de deslocação baixa, de modo a permitir uma paragem o mais próxima possível dessa posição.

Finalizado este movimento, dar-se-á início ao ciclo de *pulling* com o avanço do primeiro carro. Este movimento será realizado em controlo de posição, à velocidade de *setpoint* requerida para o material, tendo como *target* a posição que se pretende como sendo a (aproximadamente) limite do seu curso (Figura 5.6c).

De seguida ocorre o sincronismo de movimento entre os carros de arrasto, no qual o *charriot* 2 terá que acompanhar e seguir o movimento do *charriot* 1. Este período de sincronismo poderia ser definido à base de *timers*; contudo, uma vez considerados os servomotores como elementos de acionamento, ter-se-á toda a vantagem em utilizar os valores de posição fornecidos pelo *encoders* destes para a aceção da fase de sincronismo, uma vez que permitem assim uma maior certeza e definição de programação. Como tal, especificam-se as posições de início e fim de sincronismo, que porventura terão de ser atribuídas conforme as necessidades do projeto, sendo normalmente deduzidas durante a operação de teste. A instrução para sincronização de movimentos em questão encontra-se na Figura 5.6d. Já a instrução para o sincronismo de movimento em que o *charriot* 1 segue o movimento do *charriot* 2 encontra-se na Figura 5.6f.

Aqui, uma questão importante a ter em conta é que o arrasto do material pelos *charriots* apenas deverá iniciar quando a sua velocidade já for igual à velocidade do material, caso contrário provocar-lhe-á danos devido à diferença de velocidades. Contudo, dado que entre a ordem de atuação das maxilas e o ataque efetivo no material ocorre um determinado intervalo de tempo, e devido às baixas velocidades inerentes ao processo, essa ordem pode ser dada um pouco antes de o respetivo *charriot* atingir a velocidade do material. Apesar desse intervalo de tempo ser desconhecido, a Gislótica indicou que, por experiência, a ordem pode ser dada quando a velocidade for igual a 95 por cento da velocidade do material.

Outra questão essencial dá-se na fase final do sincronismo de movimentos. A partir deste momento, enquanto que se pretende que o *charriot master* deixe de ter contacto com o material e recue à velocidade máxima para a sua posição inicial, é imprescindível que o *charriot slave* mantenha a mesma velocidade, de modo a assegurar a integridade do material. Embora outros fabricantes disponibilizem uma instrução de movimento para o efeito – MC_GearOut –, no caso da Siemens tal não se encontra disponível. A solução passou por utilizar a instrução presente na Figura 5.6e.

Outro ponto a ter em conta é o erro gerado por um *setpoint* de velocidade demasiado elevado, e que não permite o recuo atempado dos *charriots* antes do início da fase de sincronismo. Este erro, de supervisão, congela a evolução do ciclo e necessita de ser reconhecido pelo utilizador antes de se transitar para a etapa inicial.

- A primeira seria realizar o *Gearing* do *charriot* 3 com os *setpoints* do *charriot* que estivesse a realizar a movimentação do material.
- A segunda passava por utilizar o *workaround* referido no capítulo 3.

Dado que a segunda solução necessitaria obrigatoriamente do uso de um objeto tecnológico adicional e necessitaria de mais adaptações a nível de projeto de simulação que a primeira, optou-se pela utilização da primeira solução. Essa escolha justifica também a circunstância do período de ativação do ciclo de corte.

Uma das preocupações a garantir durante o ciclo de corte é a de que o *encoder* está a fornecer os valores certos, dado que pode acontecer que este perca algum contacto com o material, resultando daí valores que afetam a certeza dos comprimentos de corte. Assim, antes do início de uma operação de corte, esta questão é sempre verificada.

Para além de se manter a decisão de acionar as maxilas quando a velocidade do *charriot* atinge uma velocidade igual a 95 por cento da velocidade do material, tem que se garantir que o corte só inicia quando as maxilas já estão em contacto com o material. Para tal define-se uma dada posição, que mais uma vez será depreendida de testes realizados. Esta posição deve ser tal que quando o *charriot* 3 a atingir, as maxilas já deverão estar em contacto com o material.

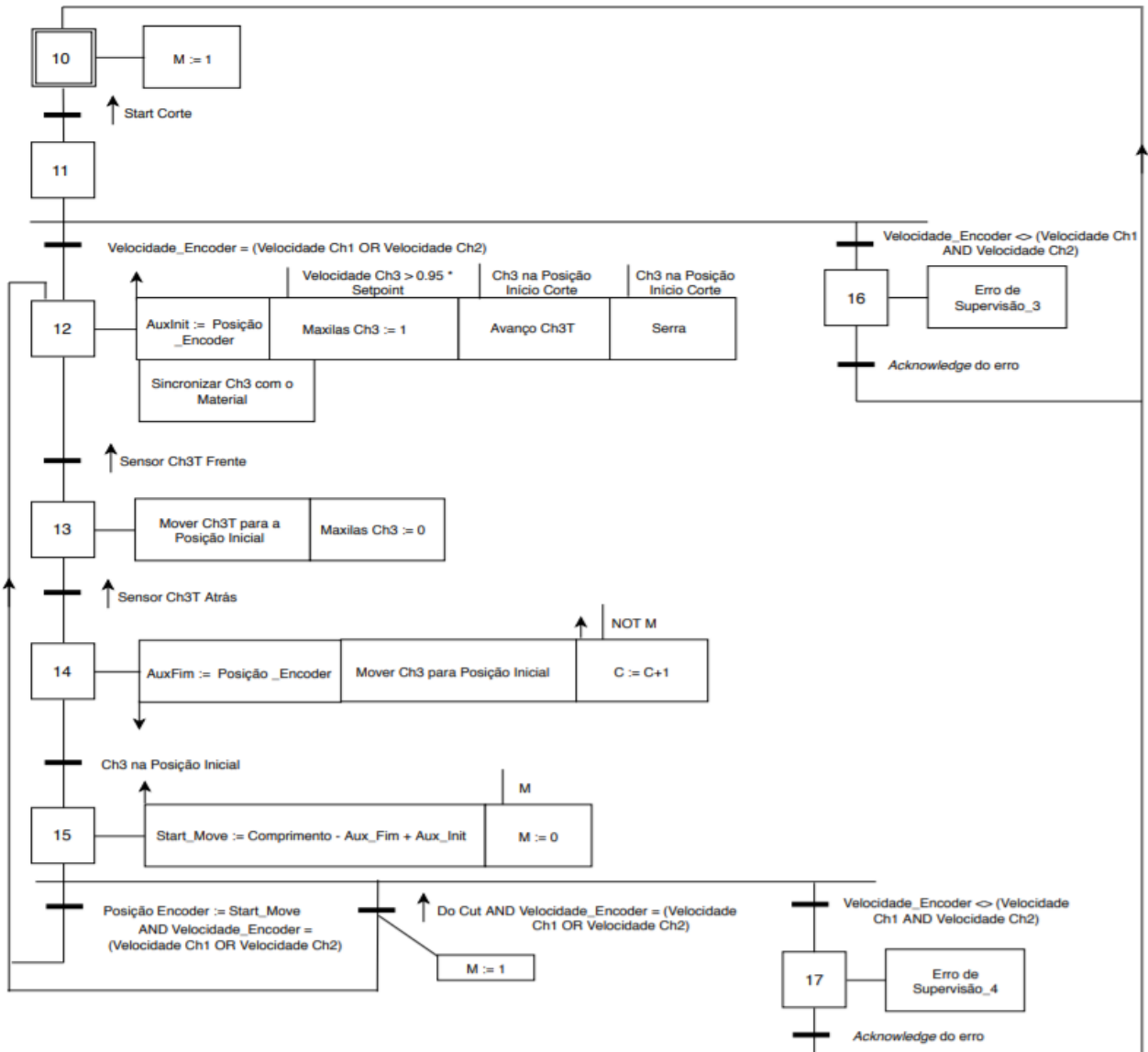


Figura 5.7 – Modelo comportamental do ciclo de corte.

5.3 Interfaces e Funcionalidades de Operação

Para a configuração e desenvolvimento das soluções de visualização, o TIA Portal possui como ferramenta integrante o SIMATIC WinCC. Esta integração simultânea com a aplicação de programação dos elementos de controlo SIMATIC STEP 7 resulta numa implementação rápida e intuitiva da solução global.

De modo a cumprir as especificações do projeto, desenvolveram-se vários ecrãs na interface homem-máquina anteriormente referida, cuja hierarquia se encontra representada na árvore de ecrãs da Figura 5.8.

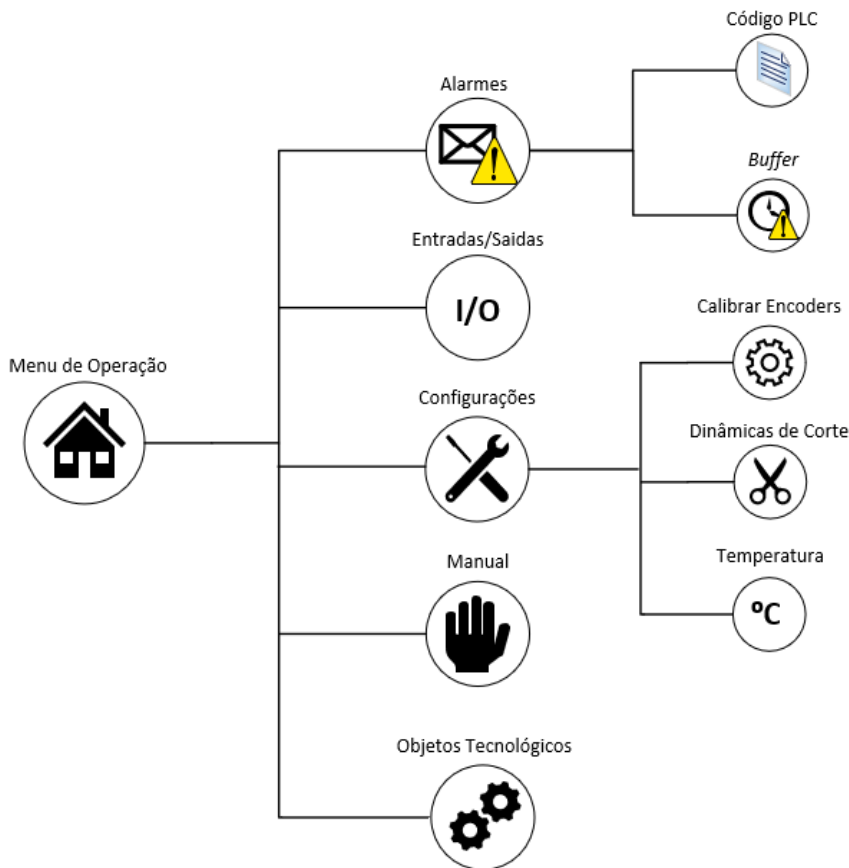


Figura 5.8 - Árvore de ecrãs desenvolvidos.

Antes de mais interessa referir que, transversalmente a todos os ecrãs apresentados, em caso de disparo de alarmes devido à ocorrência de erros no normal funcionamento do processo, surge imediatamente uma sinalética informativa da presença destes.

De seguida serão apresentados os diversos ecrãs desenvolvidos, sendo realizado paralelamente, sempre que se considere oportuno, apontamentos em relação à configuração deste. Da mesma forma, será também apontado o que foi desenvolvido em termos de programa.

5.3.1 Menu de Operação

A Figura 5.9 apresenta aquele que pode ser considerado como o menu principal de operação de toda a unidade de pultrusão, uma vez que dá acesso à configuração e manipulação das principais variáveis de interesse do processo. A partir deste, é também possibilitado o acesso aos restantes ecrãs presentes na árvore hierárquica anteriormente apresentada.

Para além da especificação do *setpoint* de velocidade para o movimento do material, e do comprimento segundo o qual este deve ser cortado, o operador tem ainda acesso ao estado do seletor automático/manual e à evolução do número de peças cortadas durante o *loop* do ciclo automático. Os restantes comandos individuais de habilitação dos motores (*power*), início e fim dos ciclos de *pulling* e de corte, assim como de corte imediato, encontram-se também acessíveis neste ecrã de operação. Ainda neste campo, o *reset* do número de peças cortadas pode ser realizado a qualquer momento por parte do operador.

Outra particularidade interessante deste e de alguns dos menus apresentados de seguida é o facto de se ter atribuído *Hotkeys* aos botões que se prevê de maior uso pelo operador, reduzindo assim um possível desgaste do ecrã da HMI, não tão pouco recorrente devido a contactos menos cuidados por parte do operador. Os *Hotkeys* utilizados têm ação direta sobre os botões localizados perto de si: por exemplo, o *Hotkey* F4 está associado ao botão “*Stop Cut*” presente no ecrã da HMI.

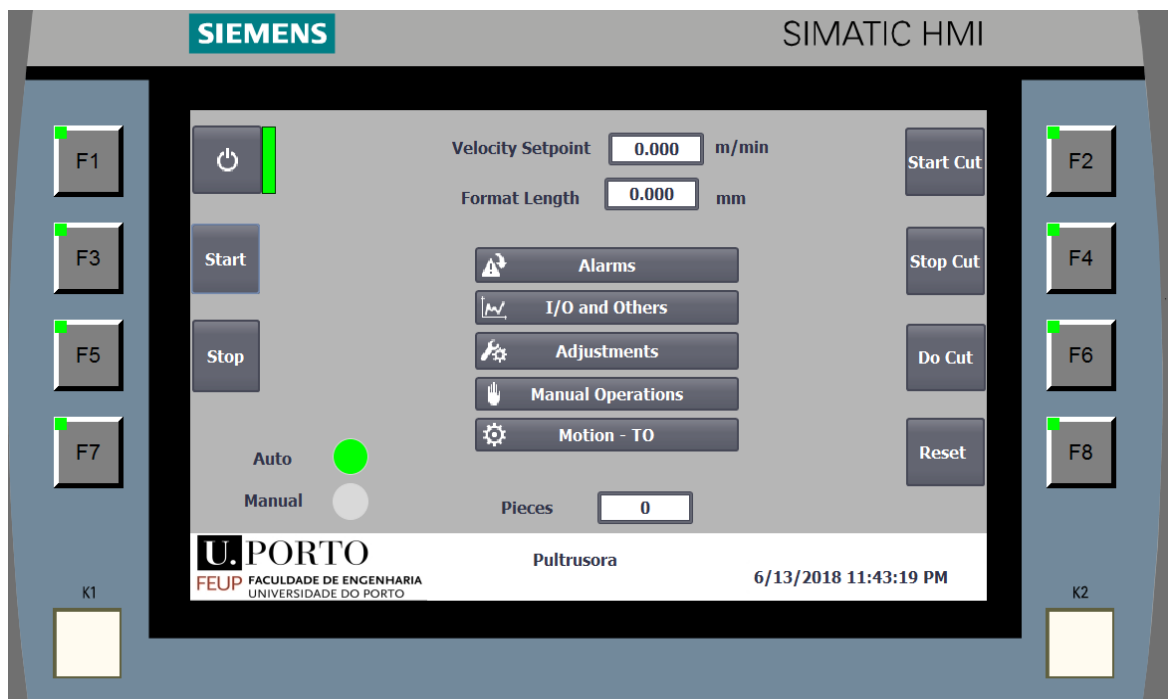


Figura 5.9 – Menu de Operação.

5.3.2 Alarmes

Este é dos ecrãs mais importantes do processo, uma vez que informa ao operador a natureza das anomalias presentes no funcionamento da máquina. Os alarmes configurados tiveram três diferentes origens:

- Provenientes diretamente de entradas ou saídas.
- Associados ao controlo de movimento.
- Erros de supervisão do S7-Graph

Para a sua representação, o WinCC possibilita a configuração de alarmes de natureza discreta ou analógica, e ainda de alarmes provenientes dos sistemas de *ProDiag* inseridos no controlador do processo. É ainda possível atribuir-lhes uma classe (erro, aviso, sistema, entre outros) ou ainda torná-los parte de um determinado grupo, cujo *acknowledge* é partilhado de forma automática pelos constituintes do grupo.

Tendo em vista a geração de alarmes provenientes de atuação das botoneiras de emergência, de informação sobre falhas na alimentação dos motores ou ainda falhas na alimentação das resistências de aquecimento, foram configurados alarmes discretos para a sua representação (Figura 5.10).

ID	Name	Alarm text	Alarm class	Trigger tag	Trigg...	Trigger address	HMI acknowl...	HMI a...	HMI acknowl...	Report
1	Falha_na_Potência	Falha_na_Potência	Errors	error_word	9	Error_Word.x9	<No tag>	0		<input type="checkbox"/>
2	Falha_na_Potência ..	Falha_na_Potência_Aquecimento	Errors	error_word	8	Error_Word.x8	<No tag>	0		<input type="checkbox"/>
3	Falha_ar	Falha Ar	Errors	error_word	10	Error_Word.x10	<No tag>	0		<input type="checkbox"/>
4	Bot Emergência Main	Botoneira de Emergência Main Atua	Errors	error_word	11	Error_Word.x11	<No tag>	0		<input type="checkbox"/>
5	Bot Emergência Hmi	Botoneira de Emergência Hmi Atua	Errors	error_word	13	Error_Word.x13	<No tag>	0		<input type="checkbox"/>
6	Bot Emergência Co...	Botoneira de Emergência Corte Atua	Errors	error_word	12	Error_Word.x12	<No tag>	0		<input type="checkbox"/>
7	Relé de Segurança	Relé de segurança desativado	Errors	error_word	14	Error_Word.x14	<No tag>	0		<input type="checkbox"/>
<Add new>										

Figura 5.10 – Configuração de alarmes discretos.

Para a configuração destes alarmes, são apenas aceitáveis variáveis do tipo INT ou WORD, sendo necessário referir qual é o bit dessa variável que realiza o *trigger* do alarme correspondente. Para tal, é necessário ter em conta que os bits são contados de forma diferente pelos SIMATIC S7 e pelo WinCC (Tabela 5.7).

Tabela 5.7 – Diferença na contagem de bits nos SIMATIC S7 e no WinCC

	Byte 0								Byte 1							
	Byte mais significativo								Byte menos significativo							
PLC S7	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
WinCC	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Os objetos tecnológicos de posicionamento e sincronismo permitem ainda especificar funcionalidades de monitorização de movimento, a atuar, por exemplo, sobre o valor do erro de seguimento. Adicionalmente, permitem a configuração de *Hardware Limit Switches* e *Software Limit Switches* (Figura 5.11a). Os primeiros têm como função limitar o curso máximo disponível para o eixo, a partir do qual a integridade física da máquina é colocada em causa – caso estes sejam atingidos, é cessado o movimento do eixo e o objeto tecnológico é desabilitado.

Com a configuração de *Software Limit Switches* pretende-se uma solução que limite o curso do eixo sem o desabilitar, o que facilita a forma com que é retornado ao seu movimento natural. Não obstante, esta limitação deve ser sempre incluída dentro dos limites impostos por *hardware*. O facto da sua configuração ser à base de *software* permite uma flexibilidade de cursos impostos, uma vez que será relativamente fácil alterar os valores de posição que estes limitam. Para a efetivação da limitação por *software*, o *encoder* do eixo terá, naturalmente, de ser previamente calibrado.

No entanto, a possibilidade de configuração de *Software Limit Switches* não substitui a necessidade de se configurar a limitação por *hardware*, uma vez que uma falha a nível de *software* teria consequências perigosas para a integridade da máquina e do operador. Assim, pode-se concluir que a configuração da limitação por *hardware* é obrigatória, enquanto que a limitação por *software* apresenta-se como complemento à primeira, representando uma boa prática para facilitar o habitual retorno ao funcionamento normal.

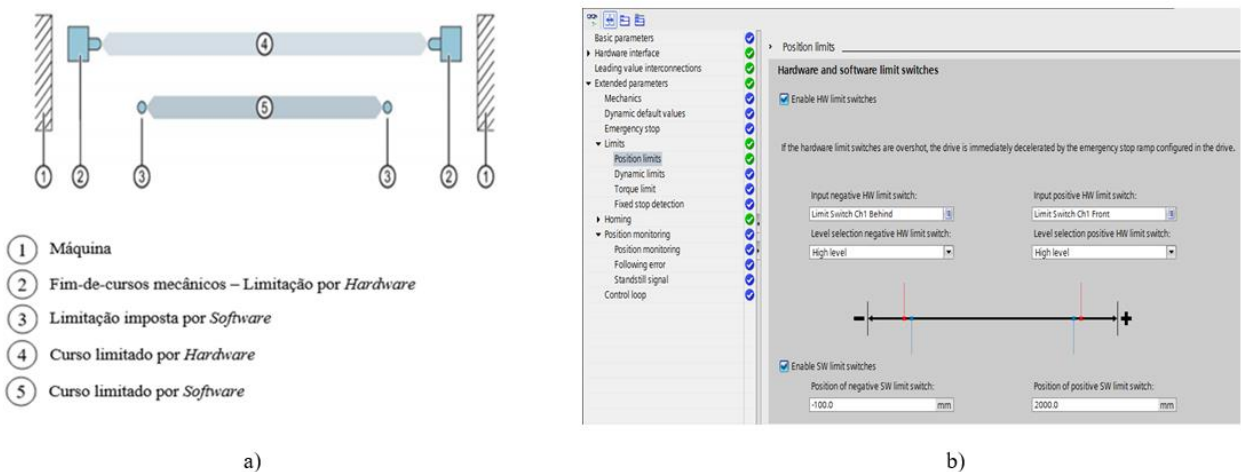


Figura 5.11 - a) Limitação imposta por estas funcionalidades e b) Configuração no TIA Portal.

Todos os erros despoletados pelo incumprimento das restrições acima referidas, assim como resultantes, por exemplo, de uma não calibração prévia de *encoders* quando é especificado um comando de posicionamento absoluto, são disponibilizados nos blocos de dados dos objetos tecnológicos. A questão que se coloca prende-se em como fornecer essas informações ao operador, na interface homem-máquina.

Uma das soluções passaria pela leitura dos valores das *words* de estado dos respetivos objetos tecnológicos (Figura 5.12). No entanto, o rastreio da causa do erro seria um processo moroso, uma vez que o operador teria que recorrer a uma tabela auxiliar que fizesse corresponder os valores obtidos com as informações de erro disponibilizadas pela Siemens. Foi nesta perspetiva que se descobriu uma das funcionalidades mais interessantes do S7-1500 – a disponibilização do bloco de função *Get_Alarm*.

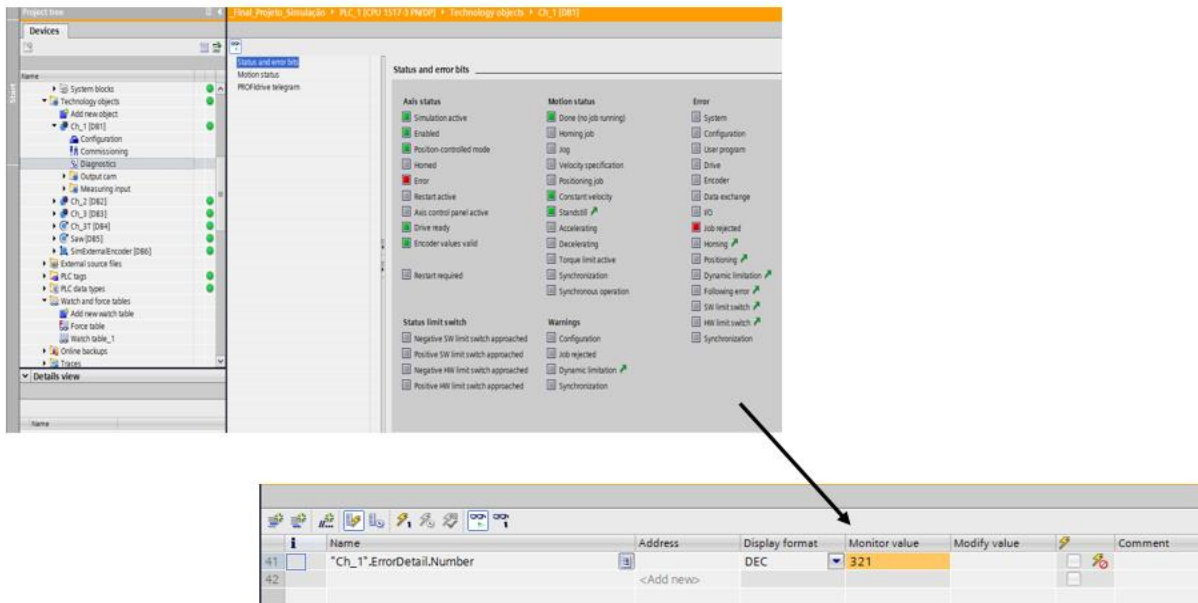


Figura 5.12 – Exemplo de um código de erro disponibilizado nas *words* do bloco de dados do objeto tecnológico.

Através desta instrução, torna-se possível aceder a alarmes provenientes dos elementos de controlo de movimento e copiá-los para um bloco de dados próprio no CPU do PLC (Figura 5.13a), o que, por sua vez, permite a visualização dos mesmos no *Alarm View* da HMI. O interessante é que, para além de minimizar tempo e esforço de configuração por parte do utilizador, o WinCC lança imediatamente a informação sobre o erro em questão. Com o consequente *Acknowledge* do erro na lista de alarmes, o erro do objeto tecnológico desaparece, e este fica pronto a realizar uma nova tarefa. A título de exemplo, a Figura 5.13b expõe o erro originado por uma ordem de movimento para posicionamento absoluto sem que o *encoder* do objeto tecnológico tivesse sido calibrado.

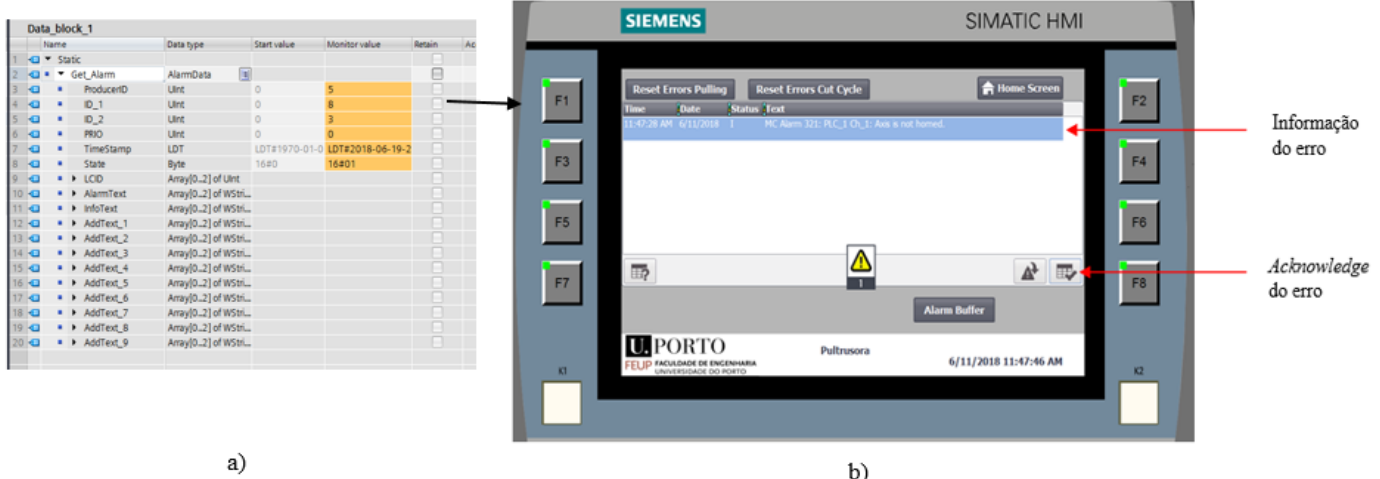


Figura 5.13 – Alarme proveniente da não calibração do *encoder* associado ao servomotor de acionamento do *charriot* 1:
 a) Armazenamento no bloco de dados e b) Representação na HMI.

Para além disso, este bloco de função permite ainda enviar para o WinCC alarmes gerados pelas instruções de *Supervision* e *Interlock* gerados no programa do utilizador. Embora tal pudesse ser feito utilizando o *ProDiag* (que permite a utilização de até 25 instruções de supervisão sem a necessidade de licenças adicionais), isto minimiza as configurações necessárias por parte do utilizador. Quando são gerados alarmes relativos ao programa do utilizador, um *clique* levará a que o operador consiga ver em que zona do programa se deu o erro – através do PLC Code View (Figura 5.14). Esta questão dá ainda mais ênfase a esta funcionalidade do S7-1500.

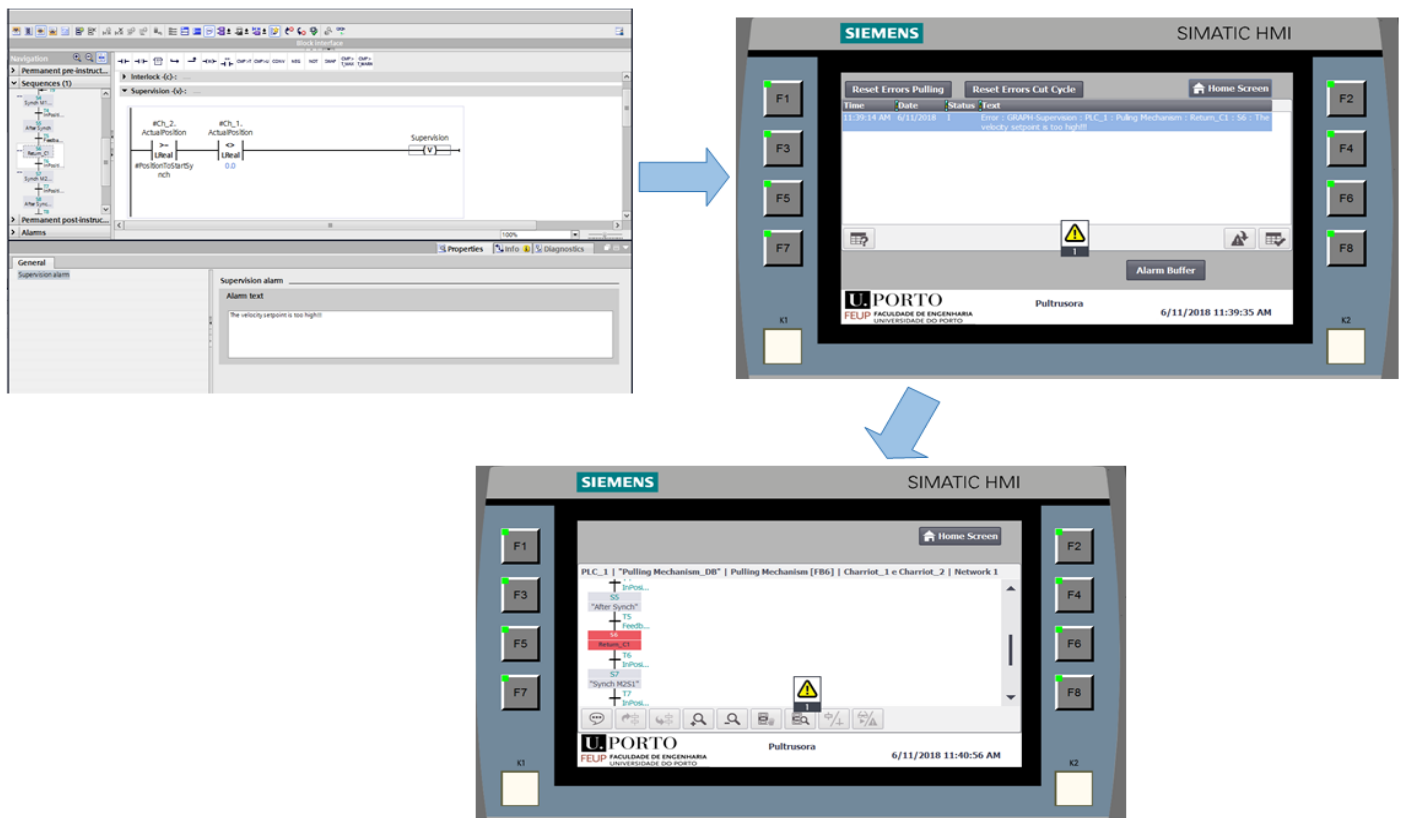


Figura 5.14 – Erro de supervisão originado pela especificação de uma velocidade demasiado alta para o processo.

No entanto, o *Acknowledge* destes erros não funciona de forma semelhante ao *acknowledge* dos erros de *Motion Control*, sendo necessário criar variáveis booleanas que realizam o *acknowledge* diretamente na interface dos blocos de função dos S7-Graph – trata-se pois das variáveis booleanas associadas ao botões “*Reset Errors Pulling*” e “*Reset Errors Cut Cycle*” presentes no ecrã de alarmes.

Para além disto, no ecrã de alarmes encontra-se disponível o acesso a um ecrã que disponibiliza o histórico de alarmes gerados.

5.3.3 Entradas e Saídas

Neste ecrã o operador tem acesso ao estado das entradas e saídas do sistema, assim como à evolução da posição e velocidade dos *charriots* 1, 2 e 3. A temperatura fornecida pelos termopares são também alvo de exibição neste ecrã (Figura 5.15).

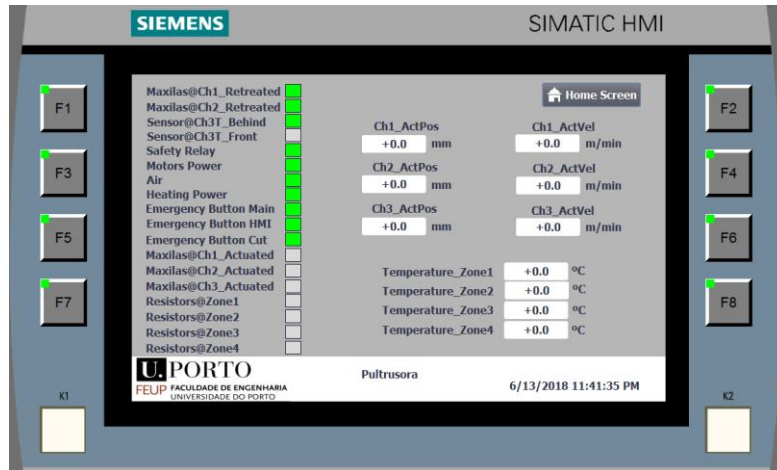


Figura 5.15 – Ecrã de entradas e saídas do sistema.

5.3.4 Configurações

Os ecrãs de configurações (Figura 5.16) têm como objetivo possibilitar ao operador o acesso à configuração de parâmetros que, por não serem tão frequentemente alterados, ou não tão específicos do processo, não estão presentes no menu de operação. Nestes é possibilitada a calibração dos *encoders* dos servomotores de acionamento dos *charriots* 1, 2 e 3, a definição das velocidades de rotação dos motores do carro transversal e da serra, e a imposição dos valores de referência de temperatura que permitam obter a curva de temperatura desejada dentro do molde/feira.

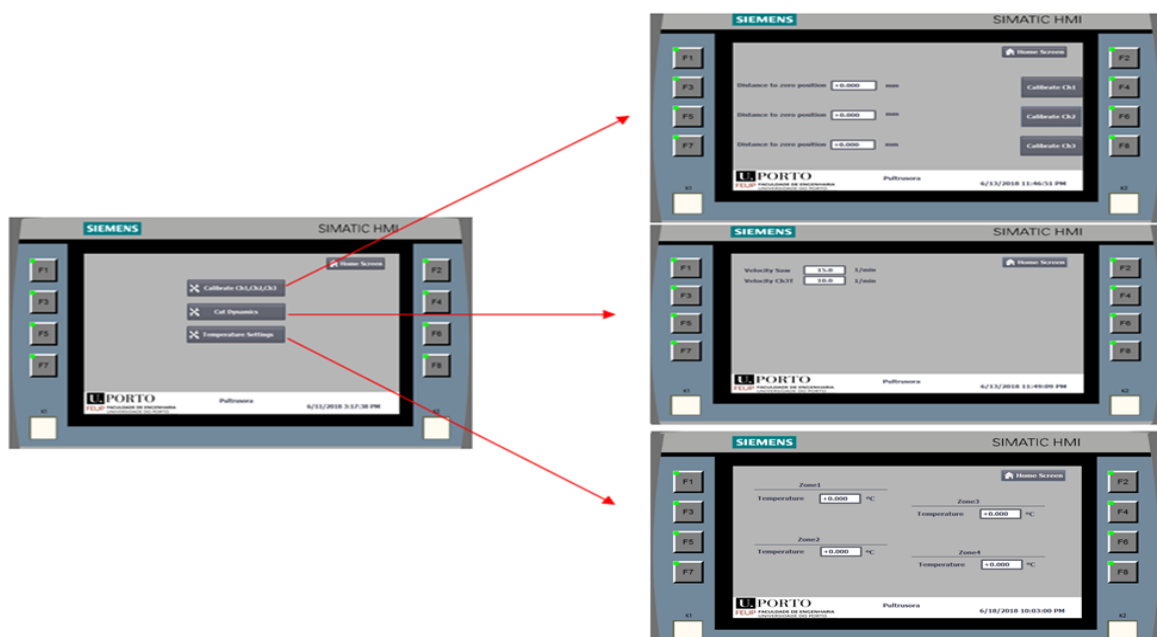


Figura 5.16 – Ecrãs de configurações disponíveis.

Os servomotores de acionamento dos *charriots* 1, 2 e 3 estão equipados com *encoders* absolutos *multi-turn*, o que garante que os seus valores de posição sejam guardados mesmo quando o PLC é desligado, contrariamente ao que acontece com os *encoders* incrementais, que necessitam de um processo de calibração específico após o *Power On* do PLC. Posto isto, é seguro admitir que a frequência do procedimento de calibração será baixa. Deste modo, a solução pensada baseia-se numa medição, por parte do operador, através de uma fita métrica ou de um qualquer outro dispositivo de medição equivalente, da distância do *charriot* ao ponto requerido como zero-máquina. Após a introdução destes valores na HMI, a calibração é concretizada através da instrução MC_Home (Figura 5.17), com o modo 7, que permite armazenar o valor da posição retentivamente na CPU do PLC.

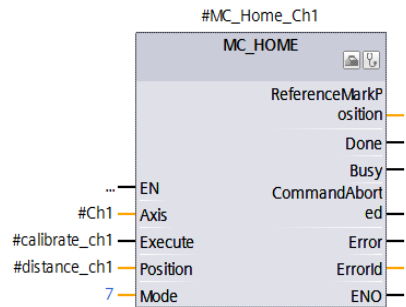


Figura 5.17 – Instrução MC_Home para calibração/referência dos *encoders*.

5.3.5 Movimento Manual

A partir deste ecrã (Figura 5.18) é possibilitado ao utilizador realizar o comando individual dos respetivos eixos de movimento. Para tal, aos *charriots* 1, 2, 3 e 3T são aplicados os comandos MC_MoveJog (Figura 5.19), enquanto que no caso da serra estar-se-á interessado em promover uma rotação permanente, e não um movimento discreto como nos anteriores. Assim, a instrução de controlo de movimento é MC_MoveVelocity, sendo que para realizar a sua cessação de movimento utiliza-se a instrução MC_Halt.

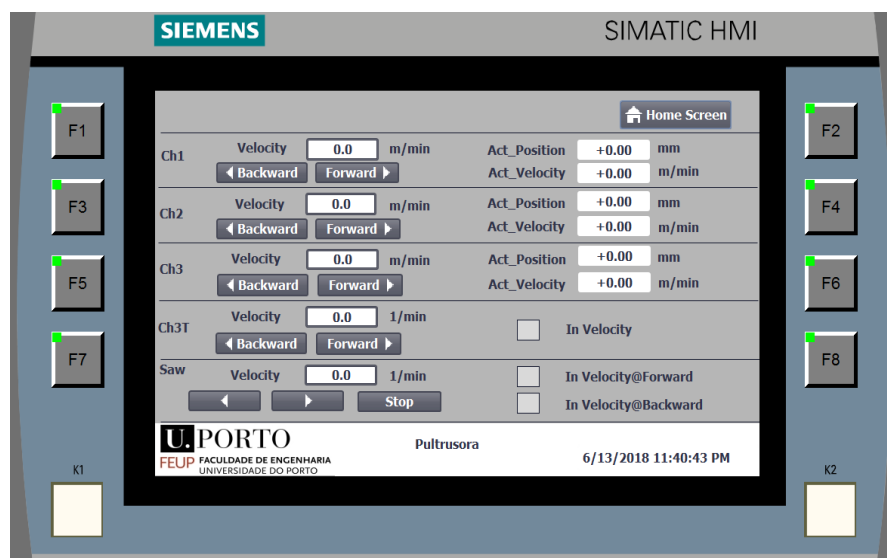


Figura 5.18 – Ecrã de operação manual.

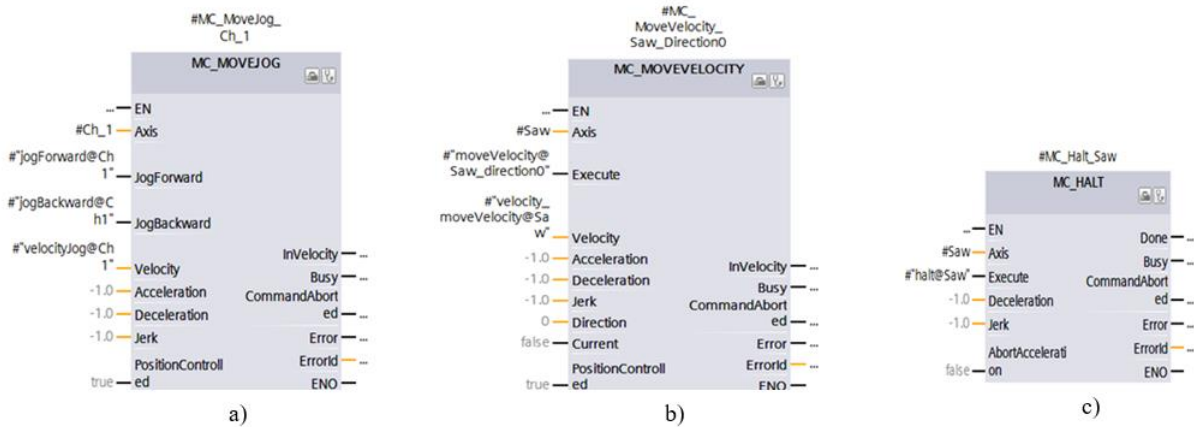


Figura 5.19 – a) MC_MoveJog, b) MC_MoveVelocity e c) MC_Halt.

5.3.6 Objetos tecnológicos

Ainda no âmbito de diagnósticos por parte do operador, é-lhe disponibilizado um ecrã (Figura 5.20) que disponibiliza informações de interesse sobre os objetos tecnológicos utilizados. Por exemplo, permite inferir sobre as condições de habilitação dos respetivos *drives*, se estes se encontram em condição de erro, ou se algum objeto tecnológico possui algum tipo de erro. Para além disso, o estado da comunicação cíclica entre o *drive* e o PLC é também avaliada.

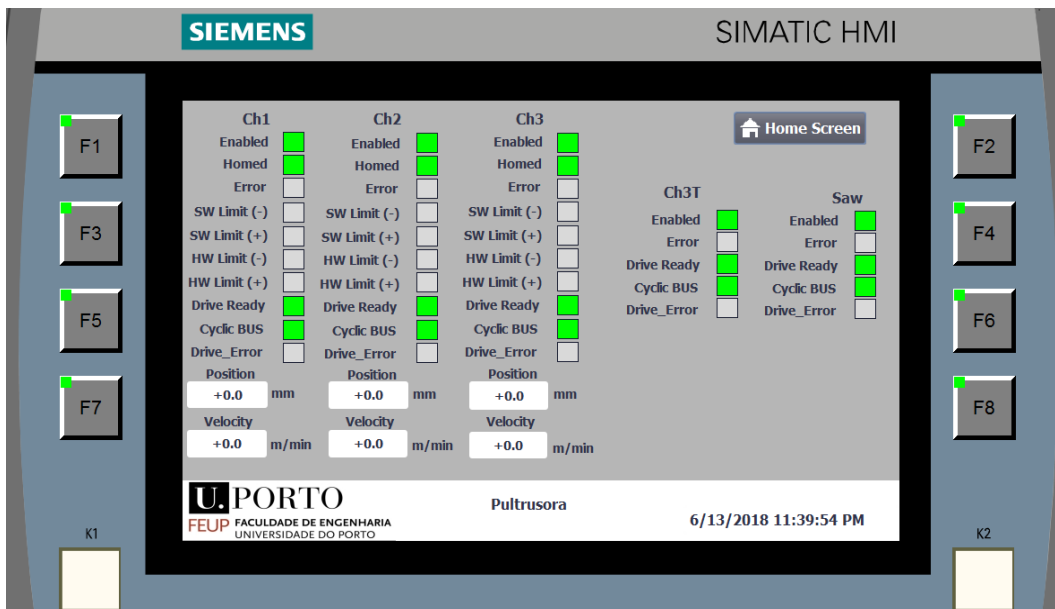


Figura 5.20 – Ecrã com informações relevantes dos objetos tecnológicos.

5.4 Testes

Como referido anteriormente, o facto de a Gislotica não ter presente nas suas instalações a máquina de pultrusão em estudo obrigou à realização paralela de dois projetos diferentes, tendo em vista que um teria de sofrer as adaptações necessárias para permitir a simulação com a ferramenta PLCSIM V14.

5.4.1 PLCSim

Contrariamente, por exemplo, ao *TwinCat*, cujo *run-time* permite simular a evolução de variáveis, o TIA Portal necessita de uma ferramenta auxiliar responsável pela simulação do funcionamento de um PLC na memória principal do dispositivo de programação. Para tal, a Siemens disponibiliza o S7-PLCSIM (Figura 5.21a), que permite simular os eixos controlados pelo PLC, e torna assim possível testar a evolução das variáveis do programa do utilizador sem necessitar de *hardware* adicional.

Para simular os objetos tecnológicos neste ambiente, é preciso declará-los como sendo virtuais e disponibilizá-los para simulação (Figura 5.21b).



Figura 5.21 – a) Ferramenta para simulação *PLCSim* e b) Configuração a realizar nos objetos tecnológicos.

Ao assinalar um objeto tecnológico como sendo virtual, informa-se o sistema que este não controla um *drive* real. Isto permite, por exemplo, ter um eixo fictício a gerar valores para múltiplos eixos reais em operações de sincronismo. Já ao ativar a simulação de um objeto tecnológico, permite-se a sua emulação com a ferramenta de simulação.

Em ambos os casos, não são enviados *setpoints* para o *drive* nem são lidos valores atuais fornecidos pelos respetivos encoders, são antes sim calculados, tendo em conta os valores de *setpoint* fictícios a enviar para o *drive*.

No entanto, nem todos os objetos tecnológicos oferecem a possibilidade de simulação. Apenas o de posicionamento, o de velocidade e o de sincronismo permitem essa situação. No caso da aplicação em estudo, surgiu então o problema de como se iria simular os valores

recebidos pelo *encoder*, uma vez que o objeto tecnológico *ExternalEncoder* não permite a sua simulação. Para tal, a solução passou por utilizar um objeto tecnológico de posicionamento para simular os valores de posição dados pelo *encoder*.

Outras questões problemáticas prenderam-se com os sinais provenientes das entradas e saídas, que foram resolvidas maioritariamente através de código, incluindo os sinais das botoneiras de emergência, o *feedback* de retorno das maxilas, entre outros.

A veracidade e validade do programa desenvolvido foi realizado através de *watch tables*, ou ainda recorrendo à ferramenta *trace*, para visualização de possíveis mudanças de variáveis, que, a olho nu, não seriam visíveis nas primeiras. Permitiu, por exemplo, confirmar a manutenção da velocidade do *charriot 2* na fase posterior ao sincronismo de movimento que, em caso contrário, danificaria o material (Figura 5.22).

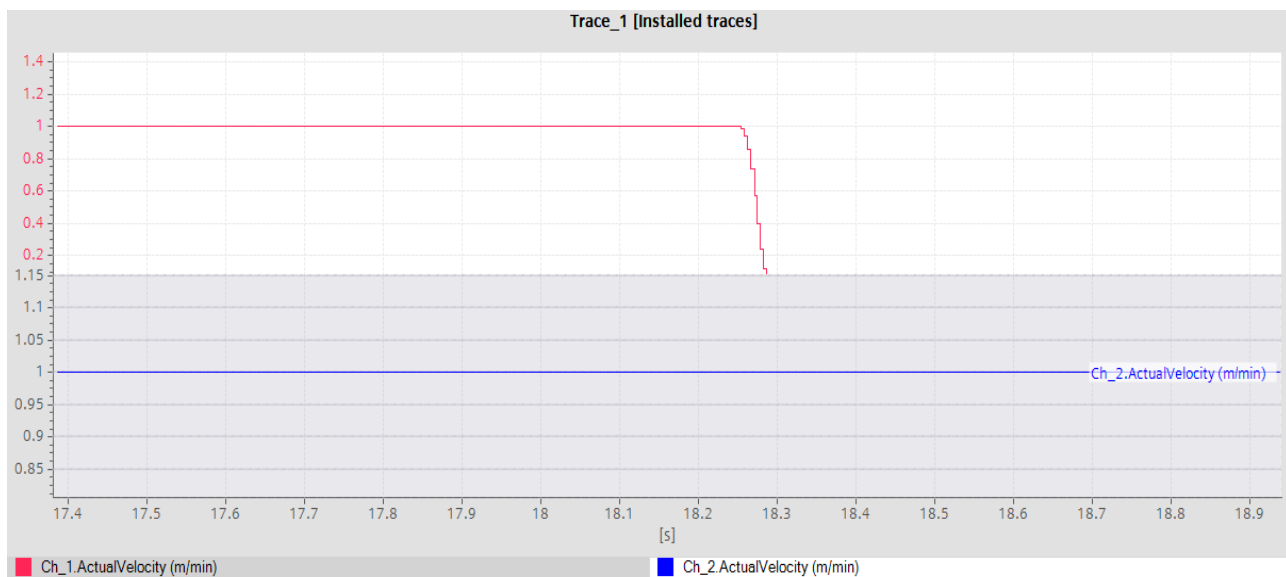


Figura 5.22 - *Tracing* do período final de sincronismo entre os *charriots* 1 e 2.

A existência de uma banca de testes permitiu também suprimir dúvidas circunstanciais que surgiram ao longo do trabalho (Figura 5.23).



a)



b)

Figura 5.23 – Bancada de testes: a) *Drive*, b) PLC.

5.5 Síntese

O capítulo que aqui finda relatou os pontos essenciais desenvolvidos na parte experimental desta dissertação. Inicialmente foram expostas e justificadas as escolhas realizadas a nível de recursos de *hardware* utilizados, seguindo-se a apresentação dos modelos comportamentais desenvolvidos para os ciclos automáticos de *pulling* e de corte.

Numa fase seguinte, referiu-se as funcionalidades adicionais desenvolvidas, assim como as interfaces homem-máquina correspondentes. Por último, foram referidas as adaptações realizadas para permitir a validação das rotinas realizadas.

6 Conclusões e perspectivas de trabalho futuro

A presente dissertação colocou em evidência as possibilidades introduzidas com a integração de funcionalidades de controlo de movimento na CPU de uma linha de controladores lógicos programáveis economicamente interessantes para a resolução de grande parte das exigências de movimento das máquinas construídas pela empresa Gislotica. Esta questão foi não só avaliada num capítulo inicial, no qual foi realizada uma comparação com a solução predecessora, como mais tarde na parte da implementação, onde a sua aplicação permitiu eliminar, por exemplo, necessidades de configuração por parte do utilizador nas questões de habilitação de *drives*, ou ainda de avaliação de estados dos *drives*, uma vez que estas funcionalidades já vêm incluídas nos objetos tecnológicos utilizados e são resolvidas por comunicação.

Paralelamente, foi possível avaliar as suas limitações a nível de sincronismo de eixos, para a qual a técnica *master/slave* pode ser utilizada. A preocupação nesta fase foi virada para aquilo que o utilizador pode ou não impor na relação entre eixos. Esta questão permitiu inferir sobre a estratégia a utilizar na atualização dos ciclos de funcionamento da unidade de pultrusão em estudo.

Por fim, foram desenvolvidas as rotinas para implementação na unidade de pultrusão, cuja validação foi obtida essencialmente por ferramentas de simulação. A validação do trabalho realizado regeu-se por uma métrica qualitativa, uma vez que uma métrica quantitativa seria claramente influenciada pela ausência das cargas presentes na situação real de funcionamento. A métrica qualitativa utilizada baseou-se em analisar a evolução das variáveis de interesse do programa, tendo em vista sobretudo a ausência de danos nas peças que seriam obtidas nesta unidade.

Embora o autor desta dissertação tenha noção que a estruturação da tese não foi desenvolvida segundo os moldes canónicos – isto é, apresentação do problema, desenho justificado da solução e respetiva experimentação – tal deveu-se ao facto do tema da tese “ter sido mantido em aberto” ao longo da execução da mesma, balizada apenas pela vontade de explorar as potencialidades das soluções Siemens para controlo de movimento.

Concluído o desenvolvimento deste trabalho, as primeiras conclusões retiradas são enunciadas de seguida:

Os controladores S7-1500 vêm com funcionalidades integradas que permitem a especificação de instruções de movimento normalizadas. Esta questão é muito importante, porque num mercado tão competitivo como o da automação, os principais fabricantes deverão não só preocupar-se com as questões de *performance* do seu equipamento, mas também em como colocar as suas ferramentas o mais intuitivas e normalizadas possível de forma a que o utilizador tire o melhor proveito das mesmas.

O abandono da solução antiga praticada pela Gislotica para estas novas ferramentas é uma aposta ganha, uma vez que acompanha uma tendência progressiva da Siemens em integrar funcionalidades diversas e necessárias para um projeto num mesmo *software* de programação.

Aliás, isso ainda foi comprovado com o lançamento da nova versão V15 do TIA Portal, a qual já inclui a possibilidade de parametrização e configuração do S120, algo que não era possível na versão anterior, o que de facto ainda minimiza mais as necessidades de contacto com o *software* Starter para uma grande quota parte das aplicações industriais.

Na área do sincronismo de movimento, foi verificada a sua aplicabilidade numa gama de aplicações mais simples, onde o eixo deverá seguir linearmente o *setpoint* de posição aplicado ao eixo *master*. Aplicações que envolvam relações não lineares, ou ainda *setpoints* provenientes dos valores lidos por sensores de posição, já não estão ao alcance deste controlador. No entanto, as suas funcionalidades mostraram-se à altura das necessidades requeridas na aplicação em estudo.

6.1 Trabalhos futuros

Uma vez considerado o carácter de simulação inerente aos testes realizados, um dos próximos passos seria proceder ao teste das rotinas desenvolvidas no equipamento em questão. Contudo, isto seria limitado apenas ao ciclo de pulling, uma vez que no módulo de corte as adaptações a realizar a nível mecânico necessitariam de ser profundas.

Na possibilidade de um construção futura deste equipamento, para além dos ajustes mecânicos a serem efetuados, um dos aspetos a realizar seria também o dimensionamento eletromecânico do sistema.

7 Bibliografia

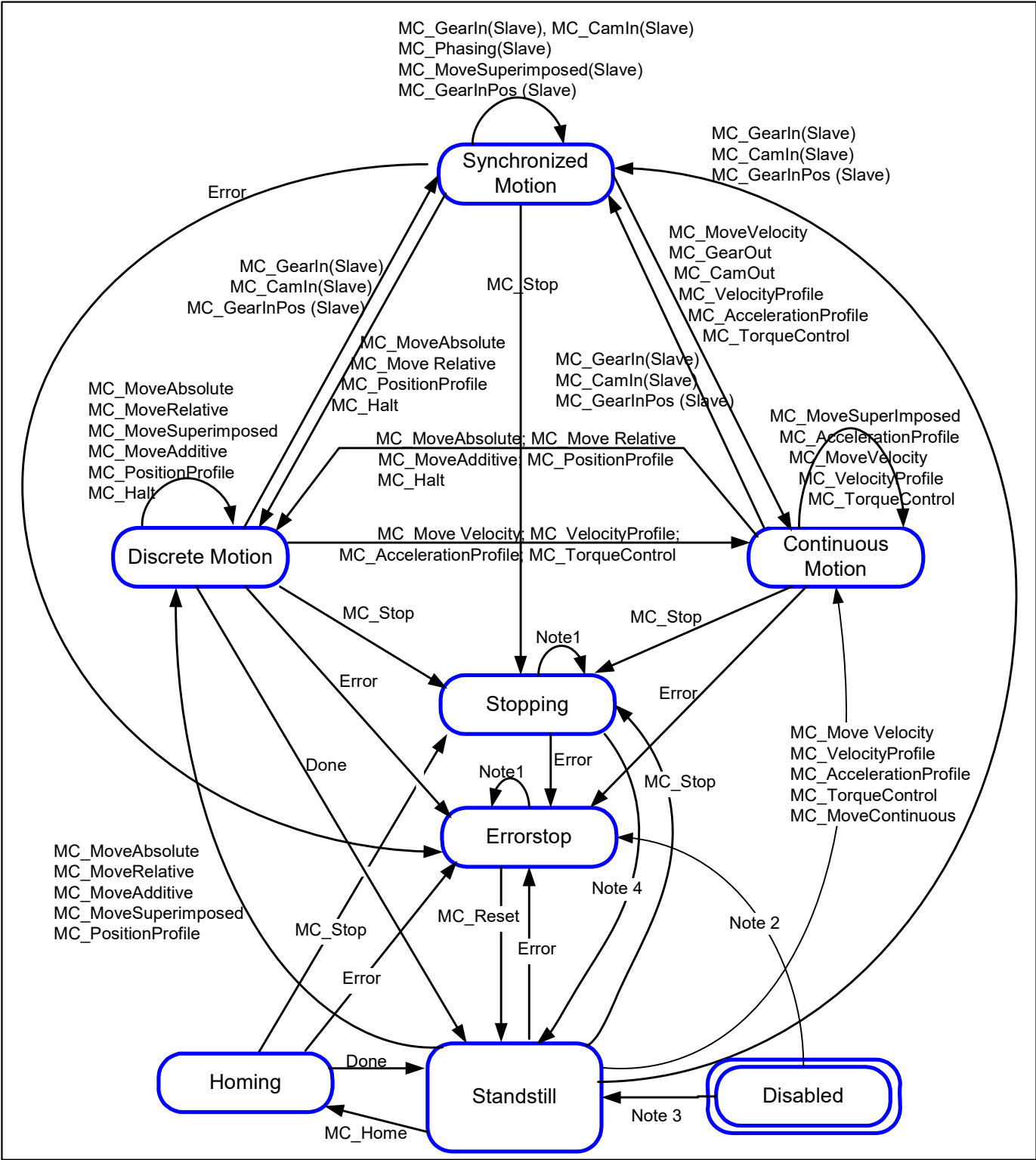
- [1] K. Heinz-John e M. Tiegelkamp, IEC61131-3: Programming Industrial Automation Systems, 2ª Edição ed., Springer, 2009.
- [2] Siemens, *Programming Guideline for S7-1200/S7-1500*, 2014. [Online]. Available: <https://support.industry.siemens.com/cs/document/81318674/programming-guidelines-and-programming-styleguide-for-simatic-s7-1200-and-s7-1500?dti=0&lc=en-WW>.
- [3] Siemens, *Standards Compliance according to IEC 61131-3*. [Online]. Available: https://cache.industry.siemens.com/dl/files/932/8790932/att_82258/v1/norm_tbl.pdf.
- [4] Siemens, "SINAMICS S120 Drive Functions", 2017, pp. 29-60, 999-1001.
- [5] Siemens, *SINAMICS Family and SINAMICS S120*, 2009. [Online]. Available: https://www.slideshare.net/marcosromanholo7/1-sinamics-components?next_slideshow=1.
- [6] Siemens, *SIMOTION, SINAMICS & SIMOTICS*, 2013. [Online]. Available: http://www1.siemens.cz/ad/current/content/data_files/katalogy/pm21/cat_pm_21_2013_en.pdf.
- [7] Siemens, *SINAMICS S120: The flexible drive system for high-performance motion control applications*, 2017. [Online]. Available: <https://www.industry.usa.siemens.com/drives/us/en/electric-drives/ac-drives/high-performance-and-servo-drives/sinamics-s120-modular-drive-system/documents/driv-sinamics-s120-brochure.pdf>.
- [8] Siemens, *How Can You Integrate a Drive into the TIA Portal via the Device Master File (GSD)?*, 2014. [Online]. Available: [https://support.industry.siemens.com/cs/document/73257075/how-can-you-integrate-a-drive-into-the-tia-portal-via-the-device-master-file-\(gsd\)-?lc=en-WW](https://support.industry.siemens.com/cs/document/73257075/how-can-you-integrate-a-drive-into-the-tia-portal-via-the-device-master-file-(gsd)-?lc=en-WW).
- [9] Siemens, *The Technology Objects of SIMATIC S7-1500(T)*, 2017. [Online]. Available: https://cache.industry.siemens.com/dl/files/134/109743134/att_928448/v2/109743134_S7-1500T_TechnologyObjects_DOC_v10_en.pdf.
- [10] Siemens, *SINAMICS G120 Basic Positioner (EPos) for CU250-2 Control Units*, 2017. [Online]. Available: https://cache.industry.siemens.com/dl/files/321/109751321/att_931956/v1/G120_EPos_fct_man_0917_en-US.pdf.
- [11] Siemens, *The Technology Objects (TO) of SIMATIC S7-1500(T)*, 2017. [Online]. Available: [https://support.industry.siemens.com/cs/document/109743134/technology-objects-of-the-simatic-s7-1500\(t\)?dti=0&lc=en-WW](https://support.industry.siemens.com/cs/document/109743134/technology-objects-of-the-simatic-s7-1500(t)?dti=0&lc=en-WW).
- [12] Siemens, *S7-1500T Motion Control V3.0 in the TIA Portal V14*, 2016. [Online]. Available: https://cache.industry.siemens.com/dl/files/326/109481326/att_895991/v1/s71500t_motion_control_function_manual_en-US_en-US.pdf.
- [13] PLCOpen, *TC3 - Certification of IEC 61131-3 Environments*. [Online]. Available: http://www.plcopen.org/pages/tc3_certification/.

- [14] J. Rathkey, *Multi-Axis Synchronization*. [Online]. Available: <https://www.automation.com/library/articles-white-papers/motion-control/multi-axis-synchronization>.
- [15] M. Woelfel, "Introduction to eletronic cam", vol. 19, *Assembly Automation*, 1999, pp. 17-24.
- [16] H. Gurocak, "Industrial Motor Control", Wiley, 2016, pp. 257-262.
- [17] P. Smith, *Relative Gearing with a External Encoder with a Siemens 1500 PLC*, 2017. [Online]. Available: <https://www.dmcinfo.com/latest-thinking/blog/id/9405/relative-gearing-to-an-external-encoder-with-a-siemens-1500-plc>.
- [18] T. F. Starr, "Pultrusion for Engineers", Woodhead Publishing Limited, 2000, pp. 19-49.
- [19] C. J. M. d. Costa, "Optimização de uma máquina de pultrusão", 2012.
- [20] Pultrex, *Pultrusion Machines*. [Online]. Available: <http://pultrex.com/pultrusion/>.
- [21] Beckhoff, *CX1000 Embedded PC Hardware Documentation*, 2006. [Online]. Available: https://download.beckhoff.com/download/document/ipc/embedded-pc/embedded-pc-cx/cx1000_hwen.pdf.
- [22] SEW-Eurodrive, *Main Technical Data of Asynchronous Drl Servomotors : DRL132S4*, p. 482.
- [23] SEW-Eurodrive, *Informação Técnica - MXA80A-00016-503-00*, 2011, p. 238.
- [24] SEW-Eurodrive, *Technical Data: DRL132S4*.
- [25] Pilz, *Data Sheet: PNOZ X3*. [Online]. Available: <https://www.alliedelec.com/m/d/8f2ac6dac00b4cb9c63fd21287119bd6.pdf>.
- [26] Siemens, *SINAMICS S120 Booksize Power Units*, 2013. [Online]. Available: https://cache.industry.siemens.com/dl/files/012/68042012/att_97659/v1/GH2_0113_eng_en-US.pdf.
- [27] Siemens, *Data Sheet: 6ES7517-3AP00-0AB0*, 2018. [Online]. Available: <https://mall.industry.siemens.com/mall/de/WW/Catalog/Product/6ES7517-3AP00-0AB0>.
- [28] Siemens, *Technical Data: CPU 1517-3 PN/DP*, 2015. [Online]. Available: https://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-1500/cpu/Documents/s7-1500_techn_data_cpu_en.pdf.
- [29] Siemens, *Motion Control with SIMATIC & SINAMICS*, 2014. [Online]. Available: <https://www.ad.siemens.com.cn/club/bbs/upload/file/20160617/6360179917468058614111503.pptx>.
- [30] CCB, *Control Cabinets Bonebuttel*, 2017. [Online]. Available: <https://www.ccb-doorentz.de/index.php/leistungen/automatisierung>.
- [31] Siemens, *ET200 SP distributed I/O System*, 2018.

- [32] Siemens, *SIMATIC ET200*, 2012. [Online]. Available: https://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-et200_en.pdf.
- [33] Siemens, *Data Sheet: 6ES7134-6JD00-0CA1*, 2014. [Online]. Available: https://cache.industry.siemens.com/dl/files/600/59753600/att_71712/v1/et200sp_ai_4x_rtd_tc_2_3_4_wire_hf_manual_en-US_en-US.pdf.
- [34] Siemens, *Data Sheet: 6ES7138-6BA00-0BA0*, 2014. [Online]. Available: https://cache.industry.siemens.com/dl/files/917/89295917/att_866073/v1/et200sp_tm_posinput_1_manual_en-US_en-US.pdf.
- [35] Siemens, *Mobile Panels 2nd Generation*, 2015. [Online]. Available: <https://support.industry.siemens.com/cs/document/109477691/release-for-delivery-mobile-panel-2nd-generation?dti=0&lc=en-KW>.
- [36] Siemens, *Data sheet: 6AV2125-2JB23-0AX0*, 2016.
- [37] Siemens, *SIMATIC HMI Mobile Panels 2nd Generation*, 2015. [Online]. Available: <https://www.appliedc.com/wp-content/uploads/2017/06/Mobile-Panels.pdf>.
- [38] Siemens, *PN Connection Box for Mobile Panel*, 2017. [Online]. Available: <https://support.industry.siemens.com/cs/document/109748296/pn-connection-box-for-mobile-panel?dti=0&lc=en-US>.
- [39] Siemens, *SIMATIC S7-GRAPH*. [Online]. Available: https://w3.siemens.com/mcms/simatic-controller-software/en/step7/simatic-s7-graph/pages/default.aspx#Area_20of_20application.
- [40] International Electrotechnical Commission, *IEC 68048 Ed.2, Specification language GRAFCET for sequential function charts*, 2001.
- [41] PLCOpen, *Function Blocks for Motion Control: Part 2 - Extensions*, 2005.

(Todas as referências em páginas *Web* estavam disponíveis a 25 de Junho de 2018, data de entrega da versão provisória da dissertação).

ANEXO A: Diagrama de Estados PLCOpen [41]



Note 1: In the ErrorStop and Stopping states all Function Blocks can be called, although they will not be executed, except MC_Reset which will generate a transition to the Standstill state. If an error occurs while the state machine is in the Stopping state a transition to the ErrorStop state is generated.

Note 2: Power.Enable = TRUE and there is an error in the Axis.

Note 3: Power.Enable = TRUE and there is no error in the Axis.

Note 4: MC_Stop.Done AND NOT MC_Stop.Execute.

Figure 1: State Diagram

