

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

# **Ethernet de tempo real para aplicações robóticas**

**Simão Paulo Marques de Amorim**

**PREPARAÇÃO DA DISSERTAÇÃO**

**PREPARAÇÃO DA DISSERTAÇÃO**

Orientador: Paulo José Lopes Machado Portugal

12 de Fevereiro de 2021

© Simão Paulo Marques de Amorim, 2021

# **Ethernet de tempo real para aplicações robóticas**

**Simão Paulo Marques de Amorim**

**PREPARAÇÃO DA DISSERTAÇÃO**



# Resumo

Vivemos num mundo cada vez mais digital e informatizado onde existe uma constante necessidade de interligação entre tudo e todos. Os sistemas robóticos modernos não escapam a esta necessidade e, por isso, é preciso adaptá-los. Existem no mercado várias soluções de redes de comunicação de tempo real, já bem estabelecidas, mas em todas se encontra a mesma lacuna: a escassez de material educativo acerca delas.

Este documento pretende apresentar as duas soluções propostas para colmatar um pouco esta lacuna na rede *EtherCAT* e demonstrar o trabalho de pesquisa e estudo preliminar efetuado para suportar e servir de ponto de partida para o desenvolvimento aprofundado de uma das soluções. Nos capítulos seguintes será explicitado o contexto e motivação para a realização deste projeto, os objetivos que propomos alcaçar, uma descrição do problema incluindo a sua caracterização, uma apresentação não exaustiva da tecnologia por detrás da rede *EtherCAT*, a explicação das soluções propostas e, por fim, um planeamento de tarefas e objetivos com calendarização dos mesmos.



*“Tell me and I forget. Teach me and I remember. Involve me and I learn.”*

Benjamin Franklin



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto . . . . .	1
1.2	Motivação . . . . .	1
1.3	Objetivos . . . . .	2
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>3</b>
2.1	Definição e caracterização do problema . . . . .	3
2.2	<i>EtherCAT</i> . . . . .	5
2.2.1	Princípio de funcionamento . . . . .	5
2.2.2	Arquiteturas de rede . . . . .	6
2.2.3	Sincronização de relógios . . . . .	6
2.2.4	Conclusão . . . . .	7
2.3	Soluções propostas . . . . .	7
2.3.1	Controlo de discos perfurados . . . . .	9
2.3.2	Seguimento de um traçado com um braço robótico . . . . .	9
<b>3</b>	<b>Plano de trabalho</b>	<b>13</b>
3.1	Tarefas, objetivos e metodologia de abordagem . . . . .	13
3.1.1	Calendarização . . . . .	13
3.1.2	Tecnologias e ferramentas a usar . . . . .	13
	<b>Referências</b>	<b>15</b>



# **Lista de Figuras**

2.1	Estrutura de um <i>frame EtherCAT</i> . . . . .	5
2.2	Configuração mista de topologias suportadas pela rede <i>EtherCAT</i> . . . . .	6
2.3	Arquitetura da rede <i>EtherCAT</i> pretendida . . . . .	8
2.4	Simulação do demonstrador - Posição 1 . . . . .	10
2.5	Simulação do demonstrador - Posição 2 . . . . .	10
2.6	Simulação do demonstrador - Posição 3 . . . . .	11
3.1	Diagrama de Gantt das tarefas propostas . . . . .	14



# **Abreviaturas e Símbolos**

FEUP	Faculdade de Engenharia da Universidade do Porto
ESC	EtherCAT Slave Controller
I/O	Input/Output
PLA	Polylactic Acid
MDF	Medium Density Fiberboard
PID	Proportional/Integral/Derivative
MAC	Media Access Controller



# Capítulo 1

## Introdução

### 1.1 Contexto

A recente digitalização da indústria introduz exigências cada vez maiores de comunicações robustas, fiáveis e determinísticas na área da automação industrial, em especial nas aplicações com requisitos de tempo-real. Nos últimos anos tem-se assistido a uma crescente utilização de redes industriais baseadas na tecnologia *Ethernet*. Geralmente denominadas redes de *Ethernet Industrial* e de *Tempo-Real*, estas tecnologias são adaptações da tecnologia *Ethernet* original (IEEE 802.3) com o objetivo de interligar os dispositivos de um sistema com requisitos de temporais exigentes. As soluções *EthernetIP*, *Profinet* e *EtherCAT* são os exemplos mais comuns deste tipo de redes de comunicação.

A área da robótica industrial é um ótimo exemplo de que a utilização de redes *Ethernet* de tempo-real faz sentido. A sincronização de ações entre componentes e a entrega e/ou difusão de mensagens cumprindo intervalos de tempo determinísticos são apenas duas características que tornam a utilização destas redes quase indispensável em sistemas robóticos modernos.

### 1.2 Motivação

A forte presença das redes *Ethernet* de tempo-real na indústria introduz a necessidade de dar formação acerca deste tema aos novos técnicos e engenheiros da área. É, portanto, necessário colmatar a falta de formação acerca deste tema na oferta formativa do curso de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto. A escassez no mercado de material didático de formação restringe o tipo de formação possível a bases puramente teóricas. Assim pretende-se desenvolver um sistema de demonstração intuitivo capaz de evidenciar as capacidades e limitações destas redes, que será posteriormente incluído nas atividades letivas do curso.

Atendendo à atual disponibilidade de material de redes *EtherCAT* presente na FEUP, decidiu-se utilizar esta tecnologia para o desenvolvimento do demonstrador em questão.

### 1.3 Objetivos

O objetivo fundamental desta dissertação é o efetivo desenvolvimento de uma plataforma robótica de demonstração prática e intuitiva da rede *EtherCAT* e da sua respetiva aplicação de controlo. Para isto, um estudo aprofundado da tecnologia e das funcionalidades deste protocolo é fundamental para que o sistema final possa apresentar um leque alargado das suas capacidades e limitações.

Como o foco da demonstração é a utilização destas tecnologias em aplicações robóticas, a plataforma deverá ser baseada no controlo de trajetórias de eixos, com uma configuração que permita, entre outros aspetos, demonstrar as características da rede *EtherCAT* ao nível da sincronização temporal (sincronização de ações) e da resposta temporal (tráfego em tempo-real) da aplicação de controlo.

## Capítulo 2

# Revisão Bibliográfica

### 2.1 Definição e caracterização do problema

Com a modernização constante da indústria e consequente crescente exigência de conetividade de todo o tipo de equipamentos, é imperativa a introdução de novas soluções tecnológicas que satisfaçam estes requisitos modernos mas que também mantenham a compatibilidade com as exigências satisfeitas pelos sistemas atualmente em uso.

Quando nos referimos à área da robótica, exigências temporais como o período, a latência e a periodicidade da malha de controlo são fulcrais para o funcionamento estável do sistema. Ora, a crescente exigência de conetividade moderna instiga à utilização de redes de comunicação no ambiente industrial. Quando o controlo de diversas áreas num equipamento é feito de modo central num PLC ou micro-controlador, a sincronização de alteração de estados de saídas é trivial, pois basta garantir que ambas são atualizadas no mesmo ciclo de processamento. Quando se pretende introduzir uma rede de comunicação entre o processamento central e o controlo das saídas e/ou a aquisição das entradas, é crucial garantir que não existem atrasos significativos na transmissão da informação na rede e garantir a sincronização entre as atualizações das saídas nos diversos escravos.

Nos últimos anos, várias implementações e estudos já foram realizados com manipuladores robóticos utilizando redes de comunicação na malha de controlo como [Zhang et al. \(2018\)](#), [Lei Wang et al. \(2010\)](#) ou muito recentemente [Deremetz et al. \(2020\)](#). No entanto, todas elas se focam na vertente mais industrial e técnica da solução e praticamente não existem implementações focadas no ensino.

Um demonstrador focado no ensino deve ter características apelativas de modo a que a sua utilização possa ser o mais generalizada possível. O primeiro objetivo deverá ser o desenvolvimento de um produto de baixo custo, de forma a que este possa ser adquirido em larga escala pelos estabelecimentos de ensino. É também preciso considerar que no ambiente de ensino, é mais provável o acontecimento de situações de utilização indevida de um equipamento comparativamente a um ambiente industrial, onde geralmente apenas pessoal qualificado está autorizado a interagir

com este, e portanto uma possível avaria do mesmo não pode trazer prejuízos avultados à instituição e/ou ao estudante. Naturalmente, esta redução no custo implicará sempre uma redução na qualidade do produto final face a um desenvolvimento de nível industrial, mas essa não é uma característica fundamental de um demonstrador didático, pois o seu principal objetivo é a transmissão e/ou auxílio na compreensão do conhecimento.

Tipicamente no momento em que nos é apresentada uma tecnologia desconhecida através de um demonstrador didático surgem questões acerca do demonstrador propriamente dito e não na tecnologia que ele pretende demonstrar. É, portanto, necessário desenvolver equipamentos focados no ensino que não imponham este tipo de barreira ao processo de aprendizagem. Assim, para minimizar este tipo de interferências existem duas possíveis soluções: utilizar um conceito que seja tão simples quanto possível, de modo a não permitir qualquer tipo de dúvida; ou que o público alvo tenha um conhecimento aprofundado do conceito de base do demonstrador. Considerando que este demonstrador será direcionado a estudantes de mestrado na área da robótica, o controlo de movimento e posição é um tema bem conhecido.

Complementando a característica económica de um demonstrador educativo, a utilização de componentes e sub-sistemas genéricos, denominados componentes *off-the-shelf*, facilita a aquisição dos mesmos tanto para efeitos de fabrico como para efeitos de reparação. Assim, é possível que o próprio cliente faça uma reparação do produto, sendo que a solução mais simples e prática será a troca do componente ou sub-sistema danificado.

A utilização de componentes e sub-sistemas genéricos leva-nos a mais uma característica importante de demonstradores educativos: a modularidade. Um sistema dividido em secções simples que se focam numa única tarefa é um sistema modular. Em engenharia dá-se o nome de sub-sistema a cada uma dessas secções. Cada sub-sistema por si é simples, fácil de implementar, interpretar e diagnosticar. A inter-ligação dos diferentes sub-sistemas permite obter um sistema mais complexo e o seu desenvolvimento e eventual diagnóstico pode ser feito por partes, simplificando o processo, e proporcionando também a possibilidade de este ser paralelizado. Esta modularidade permite que, no contexto de aprendizagem, seja mais fácil e rápido interpretar o funcionamento de cada sub-sistema e, consequentemente, o funcionamento geral do sistema.

A agregação das últimas três características (simplicidade, modularidade e utilização de componentes genéricos) origina mais uma característica importante de um demonstrador didático: a reprodutibilidade. Para muitas instituições de ensino é importante que este tipo de equipamentos seja facilmente reproduzível, pois muitas preferem produzir os seus próprios equipamentos, escolhendo os componentes a usar. Isto permite a cada instituição gerir a quantidade de equipamentos existentes e, inclusivamente, proporciona a possibilidade de efetuar reparações no local. O próprio ato de montagem de um equipamento educativo é interessante para auxiliar o lecionamento do tema abordado.

Por fim, uma das características mais importantes de qualquer demonstrador educativo, e razão pela qual estes existem, é o seu caráter intuitivo. Qualquer utilizador deve ser capaz de, através do próprio funcionamento do demonstrador, entender o conceito base em exposição. Em ambiente

educacional é muito importante fornecer este tipo de contacto com a tecnologia para que os estudantes possam sedimentar os conhecimentos com mais facilidade e de uma forma mais duradoura.

## 2.2 EtherCAT

A rede *EtherCAT* é uma rede de *Ethernet* industrial que usa a especificação padrão IEEE 802.3 (iee, 2018) para definir o formato dos *frames* e camada física a utilizar, mas introduz uma maneira diferente de os processar.

Esta nova forma de processamento permite uma comunicação com todos os dispositivos presentes na rede com apenas um *frame*. *EtherCAT* utiliza uma tipologia de comunicação *Master/Slave*, tipicamente implementada numa arquitetura de rede encadeada (*daisy-chain*), mas permite várias outras arquiteturas.

### 2.2.1 Princípio de funcionamento

A rede *EtherCAT* utiliza como base um *frame Ethernet* padrão, fazendo uso do endereçamento MAC e inserindo os seus datagramas de comunicação no conteúdo do *frame Ethernet*. Estes são marcados com o identificador 0x88A4 no campo *Ethertype*. Isto permite que as comunicações *EtherCAT* não dependam da pilha de protocolos de comunicação tradicionais como *TCP/IP* ou *UDP/IP*. A figura 2.1, retirada da página de apresentação da tecnologia (Group), demonstra o aspeto genérico da estrutura de um *frame EtherCAT*.

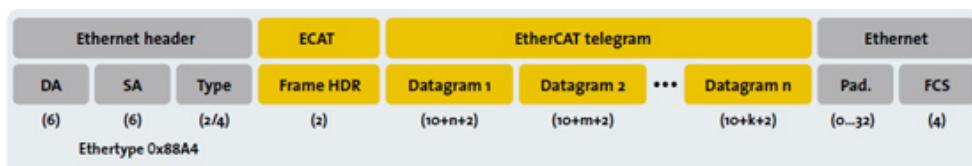


Figura 2.1: Estrutura de um *frame EtherCAT*

Vários datagramas de *EtherCAT* podem ser transmitidos no mesmo *frame Ethernet* de modo a otimizar a eficiência da transferência de informação. Durante a configuração da rede, o dispositivo mestre atribui um ou mais endereços a cada escravo, de maneira que é possível endereçar cada escravo individualmente ou endereçar grupos de escravos em simultâneo. Este processo é denominado endereçamento lógico e permite separar os dados relativos a diferentes tarefas geridas pelo dispositivo mestre. Cada datagrama enviado no *frame Ethernet* terá um cabeçalho que indicará qual o tipo de acesso que o dispositivo mestre pretende efectuar com o(s) escravo(s) identificado(s) no datagrama. Os acessos disponíveis são apenas leitura, apenas escrita ou leitura e escrita em simultâneo.

Apenas o dispositivo mestre pode iniciar um *frame* de comunicação e os dispositivos escravos limitam-se a ler a parte informação contida no *frame* que lhes é endereçada. Ao mesmo tempo, cada dispositivo escravo pode introduzir informação sua no *frame* antes de o enviar para o dispositivo seguinte.

Através destas funcionalidades de endereçamento lógico e da definição do modo de acesso em cada datagrama, o dispositivo mestre pode decidir quando pretende ler ou escrever dados de cada escravo, permitindo assim que se configurem diferentes períodos de atualização para cada tarefa, como por exemplo, fazer uma atualização dos parâmetros dos controladores dos motores com período de 1ms e fazer a atualização dos comandos manuais do utilizador com período de 10ms.

### 2.2.2 Arquiteturas de rede

Como referido em 2.2, esta rede é muito flexível em termos da topologia de rede que suporta. As mais comumente utilizadas são a encadeada, em árvore ou em estrela e é possível fazer uma combinação de topologias diferentes em diversas secções da rede. A imagem 2.2, retirada da página de apresentação ([Group](#)), demonstra estas topologias numa configuração mista.

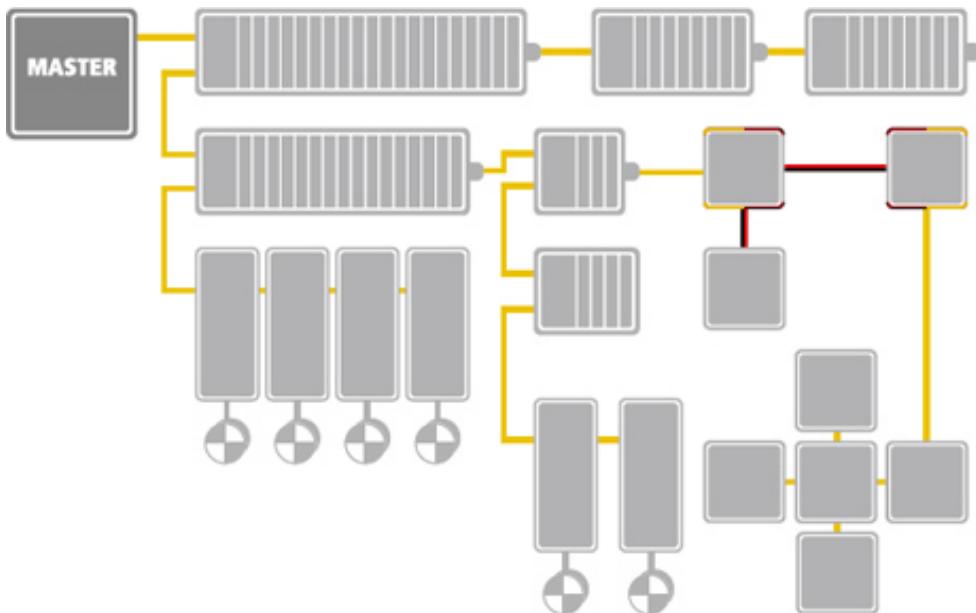


Figura 2.2: Configuração mista de topologias suportadas pela rede *EtherCAT*

### 2.2.3 Sincronização de relógios

Em aplicações onde é necessário garantir a simultaneidade de ações, como discutido em 2.1, a utilização de uma arquitetura distribuída com base em redes de comunicação dificulta o processo de manter o bom funcionamento de ações síncronas. Um bom exemplo é, exatamente, um sistema que envolve a sincronização de movimentos multi-eixo.

A rede *EtherCAT* colmata esta dificuldade através da utilização de um sistema de sincronização de relógios distribuídos (*distributed synchronised clocks*). A calibração dos relógios é implementada na íntegra no *hardware*, fazendo a distribuição cíclica do valor do relógio do primeiro dispositivo escravo por toda a rede, resultando numa sincronização dos relógios de todos

os dispositivos escravo com um valor de *jitter* abaixo de 1 microsegundo. Este valor significa que o relógio a que ele diz respeito têm um desvio relativamente ao relógio de referência de, no máximo, 1 microsegundo. Durante o período de inicialização da rede, esta operação de sincronização de relógios mede o tempo de transmissão dos dados no meio de transporte e aplica a devida compensação.

Quando todos os dispositivos têm os relógios sincronizados, estes podem atualizar as saídas no mesmo instante e fornecer um registo temporal muito preciso da aquisição das entradas. Como já foi apresentado em 2.1, esta funcionalidade é fundamental para o funcionamento de sistema robóticos que façam uso de redes de comunicação nas suas malhas de controlo. Mais à frente, quando se fizer a apresentação das soluções propostas para esta dissertação na secção 2.3, ir-se-á dar exemplos concretos desta necessidade com base no conceito de cada um dos sistemas de demonstração.

#### 2.2.4 Conclusão

Estas características permitem que o dispositivo mestre seja implementado em qualquer tipo de dispositivo que contenha uma porta de comunicação *Ethernet*. Os dispositivos escravo utilizam um *EtherCAT Slave Controller* (ESC) que processa os *frames*, fazendo com que a velocidade e tempos de resposta da rede sejam previsíveis e independentes do hardware em que os dispositivos escravo se baseiam. Assim, é possível a utilização de dispositivos escravo implementados em arquiteturas de computação diferentes dentro da mesma rede *EtherCAT*, mantendo as características e o desempenho da rede previsíveis.

### 2.3 Soluções propostas

Para atingir os objetivos propostos por esta dissertação, foram propostas duas possíveis soluções. Ambas são apresentadas de seguida sendo que maior ênfase será dada na primeira, pois é a proposta que se mostra mais adequada ao estudo em questão. A segunda proposta é mais avançada mas terá um custo bastante mais elevado. Como ela também depende de um manipulador do tipo braço robótico, a sua reprodutibilidade é bastante inferior.

Ambas as soluções têm por base o controlo de movimento através da velocidade e/ou posição de um sistema robótico de múltiplos eixos. Fazendo uso de uma arquitetura de controlo distribuída, interligada por uma rede *EtherCAT* em tipologia encadeada (ver secção 2.2.2). Esta arquitetura será constituída por um dispositivo mestre implementado num micro-computador *Raspberry Pi* programado através das linguagens descritas no padrão IEC 61161-3. Os dispositivos escravo, que farão a interface com os atuadores, sensores e interface de comando, serão implementados através de placas *Arduino UNO* agrupadas com adaptadores *EasyCAT* da [AB&T](#). Um esquema simplificado da arquitetura proposta é mostrado na figura 2.3.

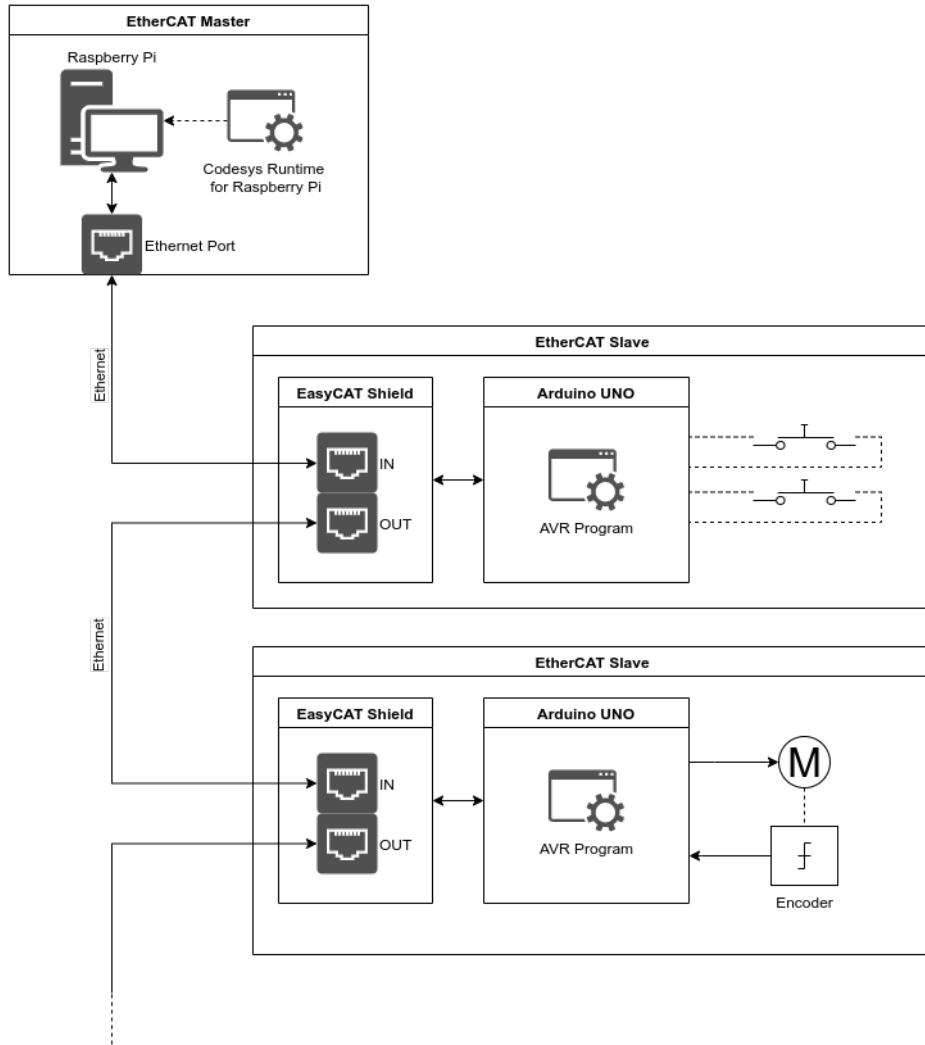


Figura 2.3: Arquitetura da rede *EtherCAT* pretendida

### 2.3.1 Controlo de discos perfurados

A primeira solução idealizada no contexto desta dissertação baseia-se num sistema de controlo e sincronização de discos rotativos independentes. Estes são perfurados na extremidade de maneira a que possam ser momentaneamente atravessados por um feixe de laser. Este último será fixo numa das extremidades do demonstrador com orientação que permita atravessar todos os discos presentes no sistema, ficando visível na extremidade oposta do demonstrador. Assim, com estes discos acoplados a motores DC com codificadores de posição (*encoder*), controlados de forma independente por placas *Arduino UNO* a funcionar em modo de *EtherCAT slave*, é possível criar diferentes cenários de controlo cujo objetivo seja periodicamente alinhar as furações dos discos com o feixe laser, permitindo que este seja viaje até à extremidade oposta.

Para complementar este conceito, será necessário permitir que o estado do ponteiro laser (ligado ou desligado) seja controlado pelo sistema, criando uma camada adicional de complexidade que ajuda a entender a importância da sincronização de relógios nas redes de comunicação de tempo real. Desta forma é possível ativar o ponteiro laser apenas quando o sistema determinar que os discos estão na orientação correta, fazendo com seja mais perceptível a importância da sincronização de relógios para que seja possível uma atualização síncrona do estado das saídas.

Idealmente esta solução deverá permitir efetuar o mesmo controlo mas não usando as capacidades da rede *EtherCAT* e usando apenas comunicação *Ethernet* simples, o que permitirá efetuar uma comparação efetiva destes protocolos e demonstrar que em ambientes de rede de comunicação tradicionais não existem considerações de tempo real nem de sincronização. Esta funcionalidade implicará verificar se os adaptadores *EasyCAT* permitem comunicação *Ethernet* simples ou até de outros protocolos de *Ethernet* industrial (p.ex. *Modbus TCP*), mas até ao momento ainda não o foi possível determinar.

De forma a proporcionar um melhor entendimento sobre o funcionamento do demonstrador aqui proposto, foi realizada uma simples simulação 3D, em computador. Desta resultaram três imagens de pré-visualização para os três possíveis estados do demonstrador, uma onde as furações dos discos se alinharam e o feixe de laser é visível na extremidade oposta, na figura 2.4, e duas onde as furações não se encontram alinhadas, fazendo com que o feixe seja interrompido por um dos discos, nas figuras 2.5 e 2.6.

Desta forma evidenciam-se as capacidades da rede *EtherCAT* no que diz respeito à resposta temporal, sincronização de relógios e capacidade de transferência de dados num sistema cujo conceito é acessível a qualquer estudante de engenharia eletrotécnica.

### 2.3.2 Seguimento de um traçado com um braço robótico

A segunda proposta de solução para a dissertação em estudo trata-se do controlo de um manipulador robótico do tipo 'braço' através da mesma arquitetura de comunicação e de controlo descrita em 2.3 e 2.3.1. O objetivo deste braço robótico será fazer o seguimento de um caminho, do estilo labirinto, com a sua ferramenta. Esta poderá ser, por exemplo, um apontador laser (derivado do conceito descrito em 2.3.1) ou, no caso de verificar viável, uma massa de esparguetes.

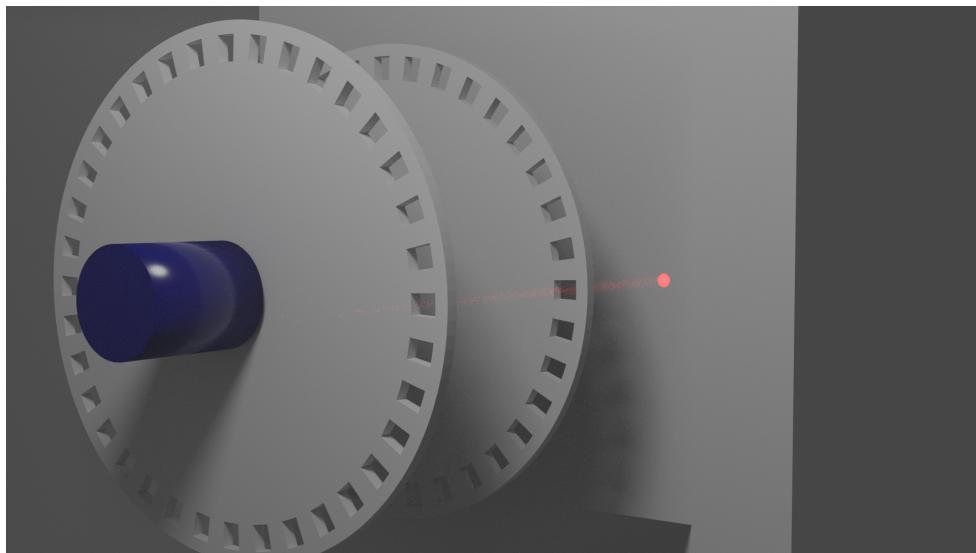


Figura 2.4: Simulação do demonstrador - Posição 1

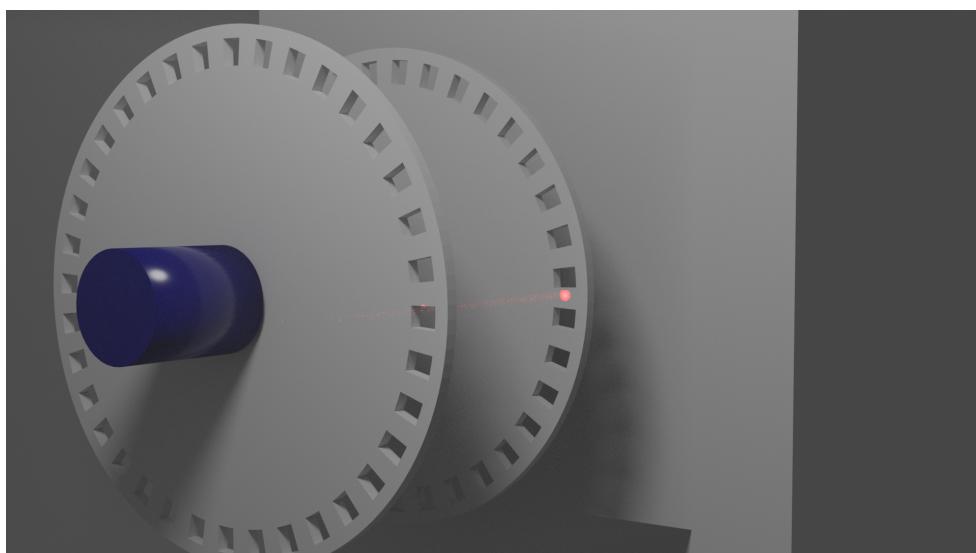


Figura 2.5: Simulação do demonstrador - Posição 2

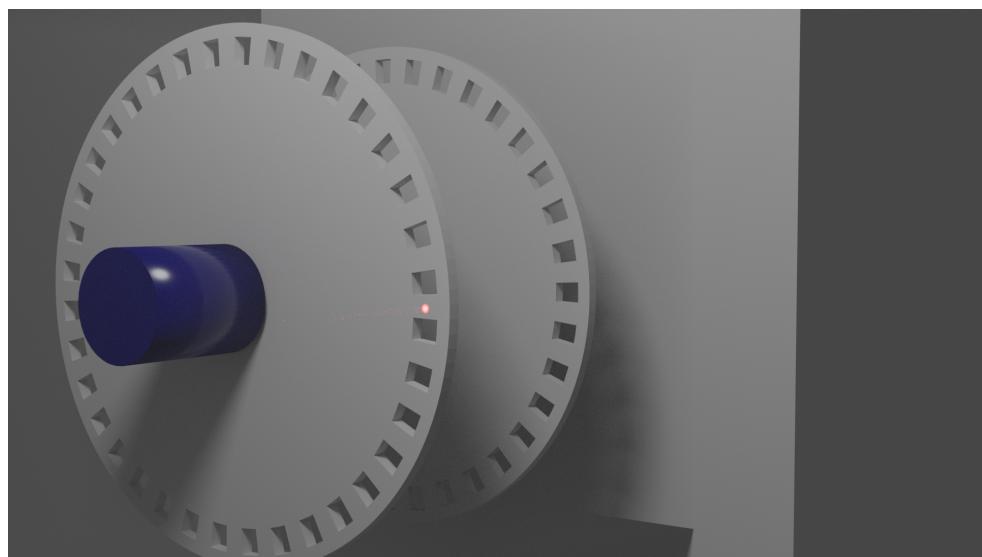


Figura 2.6: Simulação do demonstrador - Posição 3

A utilização de um objeto tridimensional como ferramenta torna a ação de percorrer um trajeto mais interessante e relevante para o objetivo. Quando se considera a utilização de um apontador laser, este só tem representação num plano bidimensional e portanto pode ser usada uma folha de papel como suporte para o trajeto. Considerando a utilização de um objeto tridimensional como ferramenta, como é o caso da massa de esparguete, torna-se necessária a utilização de um suporte tridimensional com canais para definir o trajeto. No âmbito deste projeto, um suporte em material como plástico PLA ou MDF é suficiente face às características de demonstrador didático.

### 2.3.2.1 Manipulador do tipo braço robótico

Um braço robótico é tipicamente composto por um conjunto de servo-motores, onde cada um controla a posição de uma junta do manipulador e onde o controlo simultâneo de todas as juntas permite controlar o manipulador como um todo, podendo posicionar a ferramenta, que se encontra na extremidade, numa posição pretendida. Esta posição é geralmente denominada de ponto comandado (*setpoint*).

O processo de cálculos matemáticos necessários à determinação da posição de cada junta, com o objetivo de posicionar a ferramenta num dado ponto do espaço é denominado cinemática inversa (*inverse kinematics*). A cinemática inversa é geralmente calculada no planeador de trajetória, cuja função é controlar o modo como o movimento é executado. Existem vários tipos de movimentos possíveis, mas todos eles podem ser decompostos em combinações dos seguintes tipos de movimento:

- movimento livre (também conhecido por movimento no espaço das juntas), onde cada junta é controlada de modo independente e é usada quando apenas a posição final é importante e a trajetória que a ferramenta irá percorrer até atingir o ponto comandado não é relevante. Este movimento é caracterizado por ser um movimento rápido, onde todas as juntas iniciam

e terminam o seu movimento em simultâneo, sendo para isso ajustadas as respetivas velocidades;

- movimento linear, onde é feita uma interpolação linear entre o ponto de partida e o ponto comandado, de maneira a que a ferramenta do manipulador passe por todos os pontos colineares com a reta que liga o ponto de comando e o ponto de partida. Este tipo de movimento é utilizado quando, por algum motivo, se pretende que a ferramenta se move numa linha reta entre o ponto de partida e o ponto comandado.
- movimento circular, onde é executada uma interpolação circular entre o ponto de partida e o ponto de comando, sendo necessário fornecer o raio da circunferência que se pretende percorrer ou, em alternativa, um ponto que deve ser atingido durante a trajetória, que permite calcular internamente o raio da circunferência.

Como conhecemos da geometria, qualquer forma geométrica pode ser representada através da sua decomposição em pontos singulares. Analogamente, um movimento curvilíneo pode ser decomposto numa sequência de pontos discretos. Idealmente usar-se-ia um número infinito de pontos mas, devido a limitações computacionais, é necessário limitar o número de pontos desta decomposição, baixando a precisão do movimento. O objetivo principal do planeador de trajetória é fazer a computação dos pontos de posicionamento intermédio para que seja possível realizar cada um destes tipos de movimento. Estes pontos de comando são enviados para os controladores de posição de cada eixo, geralmente do tipo *PID*, para que estes movam a respetiva junta para a posição pretendida.

### 2.3.2.2 Conclusão

Neste tipo de movimentos é fundamental que o controlo das várias juntas do braço seja preciso e feito de um modo síncrono. É exatamente neste conceito que este projeto se baseia, demonstrando que é necessária uma comunicação rápida, de baixa latência e com capacidade de sincronização de relógios para permitir a implementação de uma arquitetura de controlo distribuída, inter-ligada por uma rede de comunicação de tempo real.

# Capítulo 3

## Plano de trabalho

Neste capítulo apresenta-se o planeamento de tarefas, seus objetivos, faseamento e metodologia de abordagem. No final será feita uma breve apresentação das tecnologias e ferramentas escolhidas que permitirão realizar as tarefas e cumprir os objetivos propostos.

### 3.1 Tarefas, objetivos e metodologia de abordagem

Para poder iniciar os trabalhos com uma rede *EtherCAT* será necessário, naturalmente, ter uma rede provisória funcional onde se possa iniciar o desenvolvimento do software de controlo. Para isto, está planeada a montagem de uma pequena rede *EtherCAT* através de um PC como dispositivo mestre e um *Raspberry Pi* como dispositivo escravo. De seguida, será desenvolvido o código de baixo nível para os *Arduino UNO* em conjunto com os módulos *EasyCAT* como dispositivos escravo. Esta rede de prototipagem permitirá o desenvolvimento de uma parte substancial do código de controlo, tanto do lado do mestre como do lado dos escravos. Do lado do *Arduino UNO* o código deverá controlar a velocidade do motor e fazer a contagem dos impulsos do encoder. Do lado do *Raspberry*, o programa focar-se á no controlo de movimento e sincronização dos mesmos, bem como eventual controlo do estado do laser.

#### 3.1.1 Calendarização

De forma a tornar mais percetível a calendarização de tarefas, criou-se um gráfico do tipo *Gantt*, mostrado na figura 3.1. Este aparece compacto, apenas com distribuição pelo diversos meses, pois a distribuição por semana, embora esteja definida, gerava uma imagem muito larga.

#### 3.1.2 Tecnologias e ferramentas a usar

Como descrito na secção 2.3, ir-se-á utilizar diversos equipamentos para satisfazer requisitos diferentes, como por exemplo o *Raspberry Pi*([Foundation](#)) e dos *Arduino UNO*. O primeiro será programado utilizando a plataforma *CODESYS* ([CODESYS](#)) que permite a utilização das linguagens padrão da indústria de automação, IEC 61161-3. Os *Arduino UNO* serão programados em

TITLE	START	FINISH	CMP	2021					
				2	3	4	5	6	
EtherCAT prototype network	2021-02-15	2021-02-19	0 %	<div style="width: 10%;"></div>					
Basic Arduino libraries	2021-02-15	2021-02-19	0 %	<div style="width: 10%;"></div>					
<b>Arduino software</b>	<b>2021-02-22</b>	<b>2021-03-19</b>	<b>0 %</b>	<div style="width: 0%; background-color: #ccc;"></div>					
Arduino motor control	2021-02-22	2021-02-26	0 %	<div style="width: 10%;"></div>					
Arduino encoder parsing	2021-03-01	2021-03-05	0 %	<div style="width: 10%;"></div>					
Arduino + EasyCAT as Slave	2021-03-08	2021-03-12	0 %	<div style="width: 10%;"></div>					
Integrate all Arduino parts	2021-03-15	2021-03-19	0 %	<div style="width: 10%;"></div>					
<b>Simple EtherCAT Master</b>	<b>2021-03-22</b>	<b>2021-04-23</b>	<b>0 %</b>	<div style="width: 0%; background-color: #ccc;"></div>					
Initialization routines	2021-03-22	2021-03-26	0 %	<div style="width: 10%;"></div>					
- Hollidays -	2021-03-29	2021-04-05	Auto	<div style="width: 10%; background-color: #ccc;"></div>	<div style="width: 10%;"></div>				
Motor control	2021-04-06	2021-04-09	0 %	<div style="width: 10%;"></div>					
Position feedback	2021-04-12	2021-04-16	0 %	<div style="width: 10%;"></div>					
Motion control	2021-04-19	2021-04-23	0 %	<div style="width: 10%;"></div>					
<b>Final EtherCAT Master</b>	<b>2021-04-26</b>	<b>2021-05-28</b>	<b>0 %</b>	<div style="width: 0%; background-color: #ccc;"></div>					
Independent double control	2021-04-26	2021-04-30	0 %	<div style="width: 10%;"></div>					
Multi-Axis Synchronisations	2021-05-03	2021-05-28	0 %	<div style="width: 10%;"></div>					
Thesis Document Writting	2021-05-31	2021-06-25	0 %	<div style="width: 10%;"></div>					

Figura 3.1: Diagrama de Gantt das tarefas propostas

línguagem C, utilizando as bibliotecas e compilador dedicados para os micro-controladores AVR, que o *Arduino UNO* utiliza. Para que este possa funcionar como um escravo de *EtherCAT*, utilizar-se-á um adaptador *EasyCAT* da **AB&T** e as respetivas bibliotecas para desenvolver um programa de controlo capaz de comandar a velocidade de rotação do motor e de fazer a contagem de impulsos provenientes do codificador de posição. O valor desta contagem deverá ser enviado para o *Raspberry Pi* de forma a que este possa fazer o controlo de posição dos motores.

# Referências

- Ieee standard for ethernet. *IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015)*, páginas 1–5600, 2018. doi: 10.1109/IEEESTD.2018.8457469.
- Srl AB&T. Easycat shield for arduino. Available at <https://www.bausano.net/en/hardware/ethercat-e-arduino/easycat.html> (04/02/2021).
- CODESYS. Codesys. Available at <https://www.codesys.com/> (04/02/2021).
- Mathieu Deremetz, Pierre Letier, Gerhard Grunwald, Maximo A. Roa, Bernhard Brunner e Benoit Lietaer. Mosar-wm: A relocatable robotic arm for future on-orbit applications. 10 2020.
- Raspberry Pi Foundation. Raspberry pi. Available at <https://www.raspberrypi.org/> (04/02/2021).
- EtherCAT Technology Group. Ethercat - the ethernet fieldbus. Available at <https://www.ethercat.org/en/technology.html> (04/02/2021).
- Lei Wang, Junyan Qi, Huijuan Jia e Bin Fang. The construction of soft servo networked motion control system based on ethercat. In *2010 The 2nd Conference on Environmental Science and Information Application Technology*, volume 3, páginas 356–358, 2010. doi: 10.1109/ESIAT.2010.5568340.
- G. Zhang, F. Ni, Z. Li e H. Liu. A control system design for 7-dof light-weight robot based on ethercat bus. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, páginas 2169–2174, 2018. doi: 10.1109/ICMA.2018.8484317.