

# ENGENHARIA FÍSICA

---

## Enunciado

### Simulações de controlo em Simscape Multibody e Aplicação em Robot Real

---

Cristina P. Santos - [cristina@dei.uminho.pt](mailto:cristina@dei.uminho.pt)

Nuno Ferrete Ribeiro – [nuno.fribeiro@dei.uminho.pt](mailto:nuno.fribeiro@dei.uminho.pt)

---

## Resumo

Tendo em conta que a teoria de controlo é extremamente importante na robótica, este trabalho visa principalmente dotar os alunos da capacidade de utilizarem ferramentas de simulação do Matlab (Simscape™/Simscape Multibody™) que permitam controlar robots, fazendo com que estes possam caminhar. Assim, é esperado que os alunos numa fase inicial se ambientem ao Simscape para que percebam como funciona o modelo. De seguida, os alunos devem ajustar o modelo da simulação tendo em conta o modelo real. A partir deste ponto, é esperado que os alunos atinjam os valores ideais para o seu modelo através de várias iterações na simulação. Uma vez obtidos estes valores, os alunos devem implementar a estratégia de controlo no seu robot real de forma a que este possa caminhar de forma semelhante ao simulado. É importante que os alunos consigam identificar diferenças, caso existam, entre o modelo real e o modelo simulando, justificando devidamente a ocorrência dessas mesmas alterações. Este trabalho é uma oportunidade para os alunos implementarem, em termos práticos, estratégias nas áreas de Controlo e Robótica.

## Informação Geral

### Apresentação Final

*Novembro/Dezembro/Janeiro na UM – Gualtar*

### Objectivos

O foco deste projeto é a marcha de um robot bípede usando trajetórias simuladas e que possam ser controladas. O projeto fornece as seguintes tarefas:

1. Compreender o funcionamento do Arduino IDE através dos exemplos fornecidos

(<https://www.arduino.cc/en/software>).

2. Compreender o funcionamento do Simscape Multibody para os Walking Robots (Código disponível em: <https://www.mathworks.com/videos/modeling-and-simulation-of-walking-robots-1576560207573.html>).
3. Ajustar as características do modelo da simulação com as informações do robot real.
4. Parameterizar de forma a que o modelo Robot da simulação se possa mover.
5. Implementar a estratégia de controlo no robot real e comparar com o resultado da simulação (Se necessário efetuar ajustes e explicar a razão destas alterações).
6. Conhecer as diferentes estratégias de controlo.
7. Compreender as vantagens e desvantagens do PID para controlo de motores.
8. Conhecer diferentes formas de gerar trajetórias em robots.
9. Compreender as vantagens e desvantagens do método utilizado para gerar trajetórias.
10. Implementar/Utilizar outro método existente para geração de trajetórias.
11. Comparar e discutir resultados e trajetórias.
12. Identificar as limitações que ocorreram ao longo do trabalho e as ações necessárias para as ultrapassar, sempre que possível.

### **Materiais do Curso**

- Software: Matlab - Simscape Multibody.
- Arduino UNO (+ IDE).
- Robot Bípede.
- (Latex): MikTex & TexMaker ou Overleaf / Word.

## **Descrição do Problema**

O presente trabalho está inserido no contexto do desenvolvimento de trajetórias da marcha através de controlo de sistemas robóticos. Pretende-se a implementação de controlo PID nos motores do robot bípede, utilizando a ferramenta SIMULINK do MATLAB®, onde é possível gerar a trajetória de um modelo dinâmico. O robot bípede encontra-se representado na Figura 1. Assim sendo, o trabalho encontra-se dividido em 2 fases.

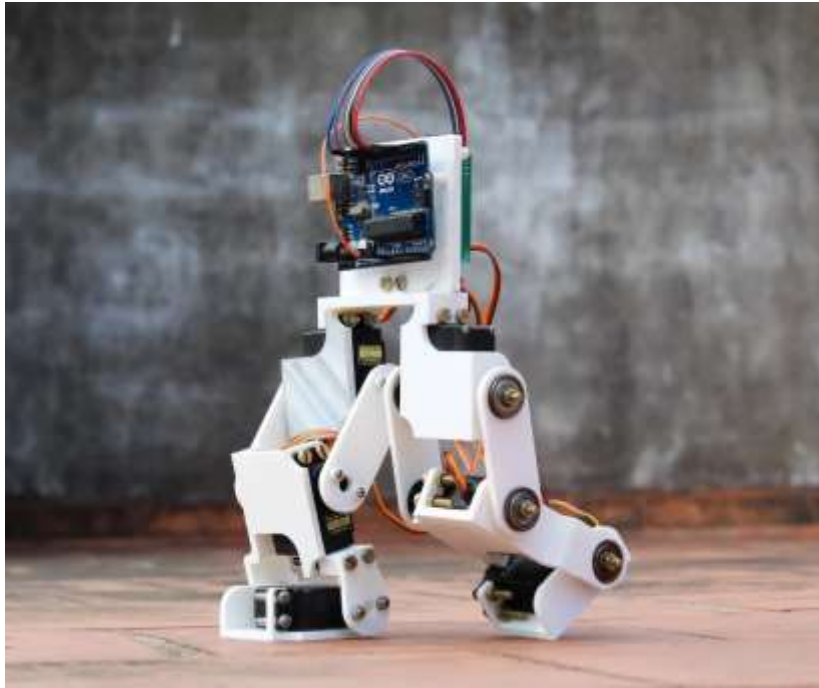


Figura 1 - Robot bípede implementado no semestre anterior.

### **Fase 1**

A primeira fase compreende a familiarização com as ferramentas já desenvolvidas em Matlab/Simulink para os Walking Robots. Os alunos devem compreender o fluxo do processo existente e perceber onde e o que podem alterar na simulação em relação ao seu projeto real. Para tal, necessitam obter as dimensões do robot real, bem como as especificações dos motores que se encontram nas juntas do robot, como por exemplo, torque (Nm), velocidade angular (rpm), *Rotor Inertia* ( $gcm^2$ ) ou *Rotor Damping* ( $m.s.N/deg$ ). Desta forma, os alunos poderão aproximar o seu modelo virtual ao modelo real, aumentando a fiabilidade das suas simulações, assim como os resultados obtidos. No fim da 1ª fase, é expectável que os alunos consigam cumprir as primeiras três tarefas, sendo que como resultado devem apresentar os resultados mais importantes da simulação com o modelo virtual o mais próximo possível do modelo real. A Figura 2 mostra o ambiente onde os alunos poderão ver o robot a caminhar.

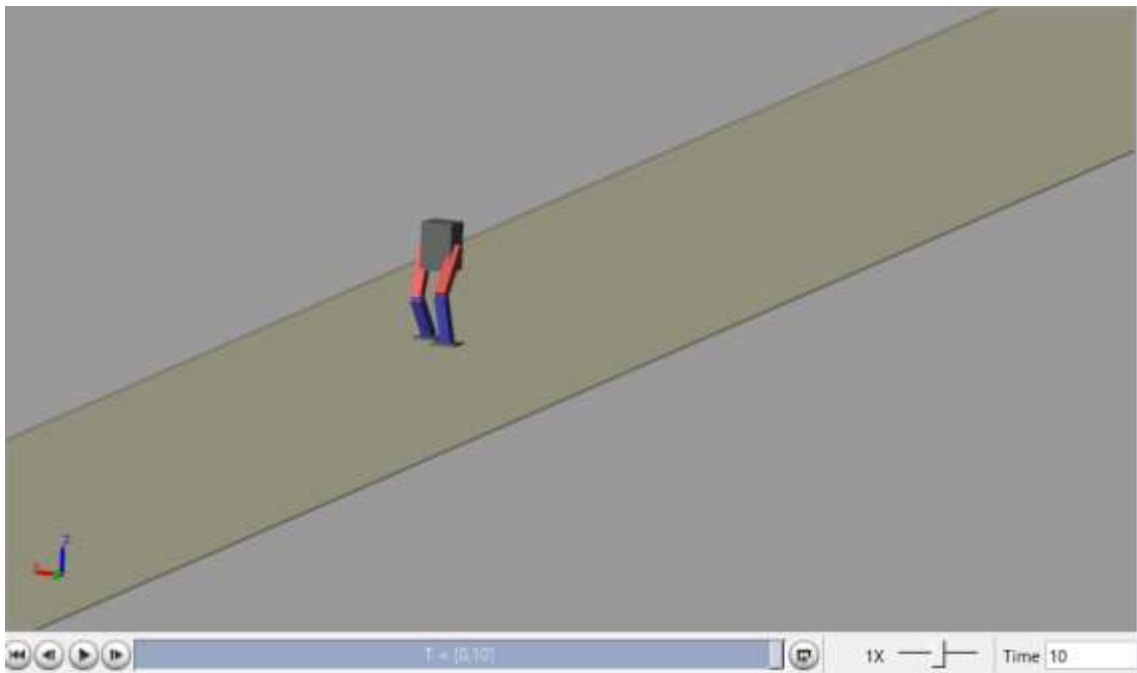


Figura 2 - Ambiente de simulação do *Walking Robot*.

Nesta primeira fase, os alunos devem começar por gerar a trajetória do Centro de Massa (COM), executando o script *"animateLIPM.mlx"*. Neste script, os alunos deverão ajustar os valores da simulação, como por exemplo, o *"swing height"* (altura do pé) ou a altura do pêndulo, tendo em conta o robot real, por forma a que a trajetória gerada esteja o mais próxima possível da realidade. Para além disso, também serão capazes de visualizar durante o período de tempo que quiserem a trajetória do COM bem como as posições dos pés. A Figura seguinte ilustra o output visual gerado por este script.

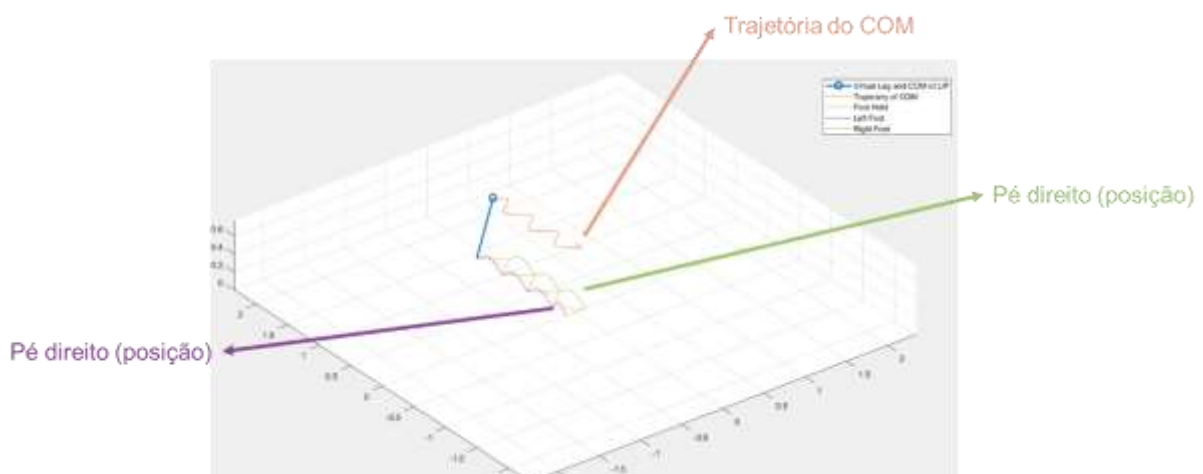


Figura 3 - Ilustração da Trajetória do COM com base nas informações introduzidas no script.

A informação resultante deste script fica guardada na variável “stepinfos”, que por sua vez não deve ser eliminada para correr o script seguinte, nomeadamente o script “*animateLIPMLocal.mlx*”. Este último script é responsável por gerar as trajetórias dos pés, guardando a informação na variável “footinfos”. Mais uma vez, esta variável não deve ser apagada para o script seguinte. A Figura 4 ilustra a trajetória dos pés resultante da execução do script “*animateLIPMLocal.mlx*”.

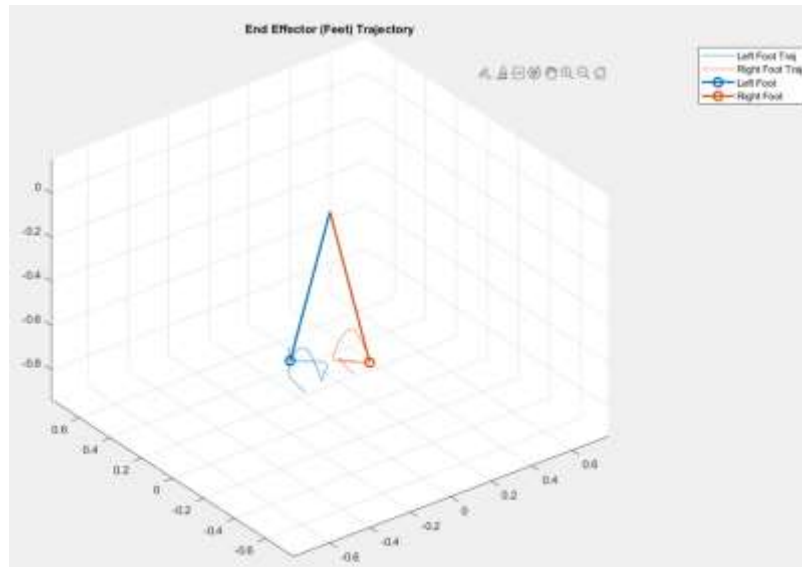


Figura 4 - Ilustração da animação gerada que exibe a trajetória dos pés, estimada a partir dos dados do script anterior.

Este é o último passo antes de se gerar as trajetórias das juntas através do script “*animateInverseKinematics.mlx*”. Este script utiliza a cinemática inversa para gerar a trajetória das juntas tendo por base a trajetória dos pés e a trajetória do COM. Neste script, os alunos deverão alterar os parâmetros Denavit-Hartenberg com base no robot real, nomeadamente, os tamanhos dos segmentos do robot. Estes parâmetros também devem ser alterados na função “*invKinBody2Foot*”. A figura seguinte ilustra uma vez mais a animação resultante da simulação deste script.

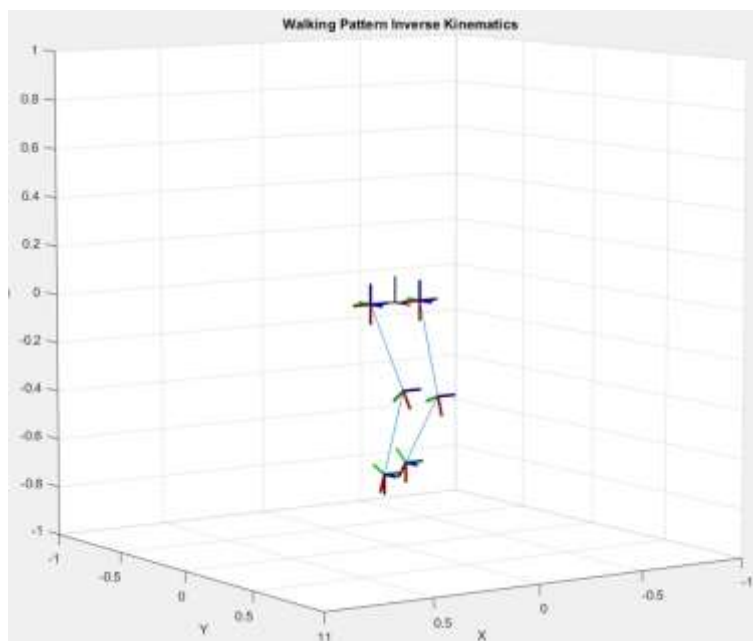


Figura 5 - Animação resultante da execução do script “animateInverseKinematics.mlx”.

De seguida, uma vez compreendido o processo (Figura 6) até esta fase, os alunos devem executar o ficheiro “*walkingRobotLIPM.slx*”. Este ficheiro de Simulink será responsável pela obtenção dos ângulos das juntas que posteriormente serão utilizados no robot real. Isto porque também é aqui onde os alunos poderão introduzir informações sobre os motores, alterar a forma física do robot, entre outras informações relevantes que aproximarão ainda mais a simulação do modelo real.



Figura 6 - Processo Completo da Fase 1 com os scripts correspondentes.

No Simulink, espera-se que os alunos utilizem e percebam os atuadores “Motion” e “Motor+Servo”. Para além disso, é esperado que alterem o bloco “Walking Robot” para uma versão mais próxima do modelo real. Posto isto, após a execução do modelo de Simulink, surge uma variável denominada “*simout*” que possui informação relativa aos ângulos das juntas do robot. A figura seguinte exemplifica o modelo de simulink. É possível aceder aos vários blocos, clicando sobre eles. Todas as alterações e testes devem ser registados para posterior análise.

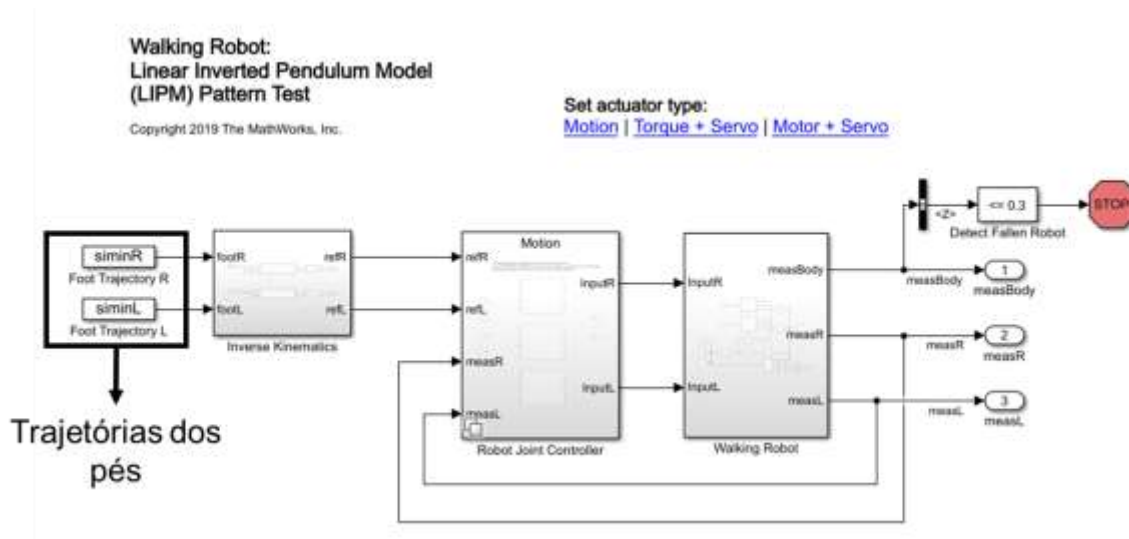


Figura 7 - Construção em Simulink para Simulação da Marcha do Robot.

## Fase 2

A segunda fase tem como objetivo utilizar os dados obtidos na simulação e implementá-los no robot real. Esta fase apresenta vários desafios, nomeadamente: i) alteração do código no Arduino para ativação dos motores; ii) implementação em código do controlo PID; iii) implementar uma estratégia para que os dados dos ângulos das juntas obtidos na fase anterior da simulação possam ser usados agora no robot real de uma forma eficiente; e iv) comparar a marcha do robot com e sem controladores PID.

Com a obtenção dos ângulos das juntas através da simulação, o código de ativação dos motores prevê-se mais simples, contudo, é necessário efetuar esta alteração de forma a que o código seja o mais compacto e eficiente possível, tendo em conta as demais funcionalidades do robot. É importante continuar a usar as informações dos demais sensores para que a marcha do robot continue a possuir alguma inteligência associada.

O controlo PID deve ser implementado em Arduino. Para tal já existem bibliotecas online que podem ser usadas neste sentido, devendo-se também implementar um filtro similar ao que se encontra na simulação. O processamento do Arduino pode eventualmente levantar algumas limitações, pelo que os alunos devem ser capazes de contornar estes problemas com outras estratégias igualmente válidas. A eficiência do código passa igualmente por uma baixa utilização de recursos, como é o caso da memória. Finalmente, é preciso perceber se o controlo PID consegue trazer benefícios para a marcha robot. Em caso afirmativo, os alunos devem indicar o que esta estratégia de controlo trouxe de benéfico. Com esta etapa, os alunos deverão ser capazes, então, de exibir um robot bípede com a funcionalidade de caminhar baseando-se na informação dos seus servomotores, sensores de força e sonar. Na pasta “Material”, os alunos podem encontrar um exemplo da implementação de controlo PID em Arduino.

# Calendário – Tarefas

Tarefa	Duração	Entrega da Tarefa	Descrição
T1	1 semana	N/A	Compreensão da ferramenta Arduino IDE para programação dos Robots
T2	2 semanas	Sim.	Compreensão da ferramenta Simscape Multibody No simulador, alterar de forma a obterem um modelo virtual que se aproxime do modelo real.
T3	1 semanas	Sim. Exibir resultados principais das simulações.	Efetuar simulações de forma a testar o modelo virtual e obter dados que possam ser usados futuramente.
T4	5 semanas	Sim. Exibir o robot real a caminhar.	Utilizar os dados da simulação no modelo real. Efetuar ajustes se necessário e compreender o porquê das diferenças.
T5	1 semana	Sim.	Entrega do relatório e apresentação do trabalho em ppt.

## Descrição das tarefas:

### Tarefa 1

- Compreender a ferramenta apresentada para controlar os Robots.
- Introdução das noções básicas sobre o Arduino IDE.

### Tarefa 2

- Compreender a ferramenta apresentada em Matlab para geração de trajetórias.
- Identificar, nos *scripts* fornecidos, as variáveis a alterar que possam ajudar a aproximar o modelo da simulação ao modelo real.
- Efetuar alterações nas variáveis dos *scripts*, considerando o robot real.
- Ao nível do Simulink, efetuar alterações no modelo e ajustar parâmetros do robot como as dimensões ou o peso, de forma a aproximar a simulação à realidade.
- Efetuar as alterações dos valores do modelo da Simulação no *script robotParameters.m* para um correto uso da simulação no Simulink.

### Tarefa 3

- Obter os ângulos das juntas a partir da simulação (variável *simsout*) para os vários tipos de atuadores (1:motion controlled, 2:torque controlled, 3:motor controlled).
- Explicar o significado de cada um dos tipos de atuadores existentes e que alterações introduzem no ângulo das juntas.
- Iterativamente, alterar os parâmetros da simulação para ajustar a marcha do robot real de forma a que este consiga caminhar (de preferência sem apoio).



- Utilizar os ficheiros CAD do robot disponibilizados para uma simulação mais realista.
- Comparar, em termos de simulação, o robot constituído por blocos (robot inicial) com o robot construído com as peças CAD.
- **Implementar outra estratégia para a geração de trajetórias à escolha dos alunos (ex: Central Pattern Generator - CPG).**
- Comparar as trajetórias obtidas em ambiente simulação a partir das diferentes estratégias de geração de trajetórias.

#### Tarefa 4

- Ajustar a frequência de amostragem (downsampling) dos sinais obtidos em simulação dos ângulos das juntas para uso no Arduino, por forma a respeitar o tempo de atuação dos motores.
- Comparar os ângulos das juntas do trabalho do 1º semestre com os ângulos das juntas obtidos em simulação.
- Implementar, em Arduino, código para a atuação dos motores. Criar uma função única, cujos parâmetros de entrada possam ser alterados de forma a que a função seja sempre chamada para a ativação de qualquer motor. Mostrar evidência do seu funcionamento.
- Explicar o processo de calibração para remoção de offsets dos ângulos das juntas/motores.
- Utilizar os ângulos das juntas da simulação no robot real. Explicar a estratégia utilizada a partir de diagramas.
- Implementar controlo PID em Arduino para os motores do robot real.
- Evidenciar as diferenças obtidas com e sem o controlo PID.
- Perceber e indicar as vantagens da utilização de um controlador PID.
- Identificar outras estratégias de controlo que poderiam ser utilizadas neste trabalho, identificando as vantagens e desvantagens de cada uma.
- Indicar qual das simulações (robot original da simulação vs robot ficheiros CAD) resultou melhor na prática, ou seja, numa marcha mais eficiente no robot real.
- Compreender as vantagens e desvantagens das estratégias de geração de trajetórias usadas e indicar qual a mais adequada para o robot em questão.

#### Tarefa 5

- Efetuar uma discussão ampla sobre as várias tarefas desempenhadas e baseada em factos objetivos.
- Escrever um relatório e preparar uma apresentação do trabalho.

#### Notas:

- N/A – Não Aplicável.
- As entregas das tarefas descritas na tabela não requerem a entrega de um documento. Apenas tratam-se de avaliações junto do professor, onde os alunos exibem os seus resultados e mostram que se encontram dentro do cronograma previsto.
- Tempo para escrever o relatório deve ser distribuído ao longo do trabalho. Contudo, encontra-se definido um tempo específico para esta parte da avaliação.