

José Daniel Ferreira Neves

PEDALEIRA ASSISTIDA ELETRICAMENTE PARA REABILITAÇÃO DOS MEMBROS INFERIORES

Dissertação de Mestrado em Engenharia Eletrotécnica e de Computadores

Setembro de 2017



UNIVERSIDADE DE COIMBRA



FACULDADE DE CIÊNCIAS E TECNOLOGIA DA
UNIVERSIDADE DE COIMBRA

Dissertação de Mestrado em Engenharia Eletrotécnica e de Computadores

**Pedaleira Assistida Eletricamente para Reabilitação
dos Membros Inferiores**

José Daniel Ferreira Neves

Orientadores:

Prof. Doutor António Paulo Mendes Breda Dias Coimbra (ISR-UC, DEEC-UC)

Prof. Doutor João Paulo Morais Ferreira (ISR-UC, ISEC-IPC)

Membros do Júri:

Presidente: Prof. Doutor Rui Pedro Duarte Cortesão

Orientador: Prof. Doutor António Paulo Mendes Breda Dias Coimbra

Vogal: Prof. Doutor Marco Alexandre Cravo Gomes

Setembro de 2017

Agradecimentos

Aproveito para expressar a minha gratidão aos meus orientadores Doutor Paulo Coimbra, Doutor João Ferreira e ao Doutor Manuel Crisóstomo pela oportunidade proporcionada, pela constante disponibilidade, paciência e vontade para que este projeto chegasse a bom porto.

Agradecer à Fundação para a Ciência e Tecnologia e ao programa COMPETE 2020 pelo apoio financeiro ao projeto “Automatic Adaptation of a Humanoid Robot Gait to Different Floor-Robot Friction Coefficients” (PTDC/EEI-AUT/5141/2014).

Dirijo também uma palavra de agradecimento aos meus pais e irmã pelo apoio incondicional durante todo este período, bem como à minha namorada, Cátia, que fez questão de estar presente em todos os momentos dando sempre o seu contributo em prol do meu sucesso.

Por fim mas não menos importante, um cumprimento especial à Sara, ao Telmo, ao Cavaleiro e a todos aqueles que direta ou indiretamente participaram não só neste projeto, seja em testes ou com sugestões, mas também neste capítulo que agora termina com a entrega desta dissertação. Obrigado!

Resumo

Os membros inferiores permitem-nos movimentar de forma natural e ergonómica e como tal são essenciais à qualidade de vida e ao bem-estar do ser humano. Recorrentemente são fustigados por diversas patologias que acabam por prejudicar essa mesma capacidade de agilidade no movimento. A pensar nisso surge a ideia de poder ajudar as pessoas com essas patologias através da criação de um sistema de reabilitação que possa reduzir ao máximo o tempo de recuperação de um doente.

Partindo de um sistema constituído por uma bicicleta estática ativa, com um motor em corrente contínua (motor DC) que é controlado em velocidade (malha fechada), sensores de força nos pedais, um sensor de batimento cardíaco e com o auxílio de um computador, procedeu-se à criação de novos modos de funcionamento com vista a reabilitação de doentes com patologias nos membros inferiores.

Dos diversos modos desenvolvidos, um permite a um paciente com perda parcial ou total da funcionalidade num dos membros inferiores, efetuar o movimento normal da pedalada, sendo este compensado pelo motor consoante a percentagem de funcionalidade em falta (modo Perna Imóvel). Com vista a abranger um leque maior de possíveis utilizadores, foi desenvolvido o modo Bicicleta Estática Normal em que é possível regular a resistência ao movimento. De forma a possibilitar ao paciente/utilizador aquecer e realizar os respetivos alongamentos, imprescindíveis antes e após o exercício físico, foi desenvolvido o modo Estimulação e Relaxamento Muscular. Com o objetivo de contornar problemas de imprecisão e saturação dos sensores de força foi desenvolvido o modo de Calibração que permite calibrar o valor dos sensores de força de acordo com o peso do paciente. Foram ainda feitos diversos ajustes na interface do paciente com o objetivo de a tornar mais funcional e intuitiva.

Estes modos de funcionamento bem como toda a interface gráfica, foram desenvolvidos com o *software* Matlab/Simulink da MathWorks.

Palavras-Chave: Bicicleta Estática, Sensores de Força, Interface Gráfica, Matlab/Simulink, Dispositivo de Reabilitação, Membros Inferiores

Abstract

The lower limbs allow us to move naturally and ergonomically and as such are essential to the life quality and welfare of the human being. Recurrently, these limbs are hampered by several pathologies which harm its agility capacity when moving. For that reason, it arises the idea to help people with those pathologies through the creation of a rehabilitation system that could substantially reduce the duration time of the patient's recovery.

Starting from a system consisting of an active static bicycle with a continuous current motor (DC motor) with speed control (closed loop), force sensors on the pedals, a heartbeat sensor and the aid of a computer, we proceed to create new operation modes to rehabilitate patients with lower limbs pathologies.

Several operation modes were developed, one allows a patient with partial or total loss of the functionality of the lower limbs to perform the usual movement of the cycling, being compensated by the motor depending on the missing functionality percentage (Weakened Leg mode). Wishing to cover a wider range of possible users or purposes, the Normal Static Bicycle mode was developed in which it is possible to regulate the resistance against movement. In order to allow the patient/user to warmup and perform its corresponding stretches, indispensable before and after workout, the Stimulation and Muscle Relaxation mode was developed. To bypass imprecision and saturation problems of the force sensors, the calibration mode was developed which can calibrate the value of the force sensors accordingly with the weight of the patient. In addition to these modes, several adjustments to the patient interface were made in order to make it more functional and intuitive.

These modes as well as the graphical interface, were developed with Matlab/Simulink software by MathWorks.

Keywords: Static Bicycle, Force Sensors, Graphical Interface, Matlab/Simulink, Rehabilitation Device, Lower Limbs

Índice

Lista de Figuras	v
Lista de Tabelas.....	vii
Lista de Acrónimos.....	ix
Capítulo 1	
Introdução.....	1
1.1 Motivação	1
1.2 Objetivos.....	2
1.3 Estrutura da Tese	2
Capítulo 2	
Estado da Arte	5
2.1 Patologias que Levam à Necessidade de Utilizar a uma Pedaleira de Reabilitação.....	5
2.2 Outros Sistemas para Reabilitação dos Membros Inferiores.....	6
2.2.1 “ <i>An Intelligent Thermal Sensing System for Automatic, Quantitative Assessment of Motion Training in Lower-Limb Rehabilitation</i> ”	6
2.2.2 “ <i>A Wheelchair Integrated Lower Limb Exercise/Rehabilitation System: Design and Experimental Results on the Knee Joint</i> ”	7
2.3.3 “ <i>Lower Limb Rehabilitation Evaluation System Based on Virtual Reality Technology</i> ”	9
Capítulo 3	
Arquitetura do Sistema e Interfaces	11
3.1 <i>Hardware</i>	12
3.1.1 Motor EC 90 <i>flat</i> e Controlador Maxon ESCON	12
3.1.2 Caixa de Velocidades	13
3.1.3 Sensores	13
3.1.3.1 Sensores de Força	13
3.1.3.2 Sensor de Monitorização da Frequência Cardíaca	15

3.1.4	Placa de Aquisição de Dados.....	16
3.2	Interfaces	17
3.2.1	Interface Direcionada ao Terapeuta.....	17
3.2.2	Interface Direcionada ao Paciente	18
Capítulo 4		
	<i>Software</i> Desenvolvido	19
4.1	Alterações na Interface do Terapeuta	19
4.1.1	Interface em Inglês	19
4.1.1.1	Dados do Paciente, Modo de Funcionamento e Dados Atuais.....	20
4.1.1.2	Mensagens de Erro	20
4.1.2	Reorganização do Painel <i>Operating Mode</i>	21
4.1.3	Opções do Perfil do Paciente.....	22
4.1.4	Botão de Ajuda	24
4.1.5	Instruções ao Escolher o Modo de Funcionamento.....	25
4.2	Modo Velocidade Constante	27
4.3	Modo Bicicleta Estática Normal.....	27
4.3.1	Implementação.....	29
4.4	Modo Estimulação e Relaxamento Muscular.....	30
4.5	Modo Perna Debilitada.....	30
4.5.1	Implementação.....	31
4.6	Modo Calibração dos Sensores de Força.....	32
4.6.1	Implementação.....	36
4.7	Modo Configurações	37
Capítulo 5		
	Testes e Resultados	39
5.1	Testes ao Modo de Calibração.....	39
5.2	Teste ao Modo Bicicleta Estática Normal	41
5.3	Testes ao Modo Perna Debilitada.....	43

Capítulo 6

Conclusões e Sugestões de Trabalhos Futuros.....	47
6.1 Conclusões.....	47
6.2 Sugestões de Trabalhos Futuros	48
6.2.1 Sugestões de <i>Hardware</i>	48
6.2.2 Sugestões de <i>Software</i>	48
Referências Bibliográficas	51

Anexo A

Datasheets.....	55
Motor EC 90 <i>flat</i> [24].....	55
Controlador Maxon ESCON [25]	56
Sensores de Força – FlexiForce A201 [10].....	59

Anexo B

Código Matlab Desenvolvido.....	61
Opções do Perfil do Paciente	61
Instruções ao Escolher Modo de Funcionamento	61
Modo Velocidade Constante	62
Painel de Regulação de Força	63
Modo Perna Debitada – Painel “Sick Leg Functionality (0-100%)”	64
Modo Configurações	66
Testes e Resultados	66

Anexo C

Simulink – Esquemático.....	67
-----------------------------	----

Lista de Figuras

Figura 2.1 - Arquitetura do Sistema [3]	7
Figura 2.2 - Configuração experimental (com KineAssist robot treadmill) e os sensores [3]	7
Figura 2.3 - Imagens térmicas e o esqueleto das pernas extraído na atividade experimental [3] ...	7
Figura 2.4 - Sistema de reabilitação de membros inferiores com cadeira de rodas [4]	8
Figura 2.5 - Teste funcional e avaliação do sistema desenvolvido [4]	8
Figura 2.6 - Modelo eclético simplificado com sensores [5]	9
Figura 2.7 - Sistema experimental [5]	9
Figura 3.1 - Estrutura da Bicicleta Estática Ativa	11
Figura 3.2 - Motor EC 90 flat [6]	12
Figura 3.3 - Controlador ESCON 50/5 [7]	12
Figura 3.4 - Caixa de Velocidades (Planetary Gearhead GP 52 C) [8]	13
Figura 3.5 - Sensor FlexiForce Modelo A201 [10]	14
Figura 3.6 - Esquema de como estão montados os sensores nos pedais da Bicicleta	14
Figura 3.7 - Circuito de aplicação dos sensores [10]	15
Figura 3.8 - Sensor de medição de Batimentos Cardíacos - Fototransistor SFH 303FA	16
Figura 3.9 - DAQ USB-6008 [13]	16
Figura 3.10 - Interface que será manipulada pelo terapeuta [12, 14]	17
Figura 3.11 - Interface gráfica (jogo) direcionado ao paciente [14]	18
Figura 4.1 - Exemplos de janelas com mensagens de erro apresentadas pelo software	21
Figura 4.2 - À esquerda antes e à direita após as melhorias	21
Figura 4.3 - Menu Opções de Perfil / Profile Options	22
Figura 4.4 - Janela da opção SAVE e respetiva mensagem de confirmação	23
Figura 4.5 - Janela da opção LOAD e respetiva mensagem de confirmação	23
Figura 4.6 - Janela da opção DELETE	24
Figura 4.7 - Janelas de confirmação exibidas após escolher o ficheiro a eliminar	24
Figura 4.8 - Botão Help e respetiva janela que surge quando se carrega	25
Figura 4.9 - À esquerda o Help em Português e à direita em Inglês	25
Figura 4.10 - Alerta ao iniciar o Modo Bicicleta Estática Normal (imagem adaptada de [21]) ...	26
Figura 4.11 - Alerta ao iniciar o Modo Perna Debitada (imagem adaptada de [21])	26
Figura 4.12 - Painel da interface referente ao modo Velocidade Constante	27
Figura 4.13 - Painel que permite regular a força resistente exercida pela bicicleta	27
Figura 4.14 - Modo Bicicleta Estática Normal em funcionamento	28

Figura 4.15 - Diagrama de Blocos, em Simulink, do Modo Bicicleta Estática Normal (com legenda)	29
Figura 4.16 - Função Matlab no bloco “Bicicleta Normal” da figura 4.15	29
Figura 4.17 - Painéis que permitem alterar os níveis de Regulação de Força e Funcionalidade ..	31
Figura 4.18 - Diagrama de Blocos, em Simulink, do Modo Perna Debilitada (com legenda).....	31
Figura 4.19 - Bloco (1) da figura 4.18.....	32
Figura 4.20 - Função matlab do bloco (2) da figura 4.19.....	32
Figura 4.21 - Material utilizado no processo de calibração	33
Figura 4.22 - Reta de calibração para os sensores do pedal esquerdo	34
Figura 4.23 - Interface com o Modo de Calibração selecionado e painel de calibração.....	35
Figura 4.24 - À esquerda o paciente coloca todo o seu peso sobre o pedal esquerdo e à direita sobre o pedal direito	36
Figura 4.25 - a) Diagrama de blocos em Simulink do Modo de Calibração. b) Interior do bloco Calibração.....	36
Figura 4.26 - Mensagem de aviso exibida ao selecionar o modo Configurações/Settings	37
Figura 4.27 - Interface com Modo Configurações (Settings) selecionado.....	37
Figura 5.1 - Valor de força medido em cada pedal (à esquerda sem calibração e à direita já com calibração)	39
Figura 5.2 - Todo o peso aplicado sobre o pedal direito (à direita sem calibração e à esquerda após a calibração).....	39
Figura 5.3 - Quando se aplica todo o peso sobre o pedal esquerdo (à direita sem calibração e à esquerda após a calibração).....	40
Figura 5.4 - Peso dividido por ambos os Pedais	40
Figura 5.5 - Quando se está a dividir todo o Peso sobre os dois pedais.....	40
Figura 5.6 - Resultados do teste ao Modo Bicicleta Estática Normal com a Pessoa 1	41
Figura 5.7 - Resultados do teste ao Modo Bicicleta Estática Normal com a Pessoa 2	42
Figura 5.8 - Resultados do teste ao Modo Bicicleta Estática Normal com a Pessoa 2	42
Figura 5.9 – Resultados do teste ao Paciente 1 que tem 0% da perna direita funcional	43
Figura 5.10 – Resultados do teste ao Paciente 2 que tem 20% da perna direita funcional	44
Figura 5.11 – Resultados do teste ao Paciente 3 que tem 50% da perna esquerda funcional	44
Figura 5.12 – Resultados do teste ao Paciente 3 com 70% da perna esquerda funcional	45

Lista de Tabelas

Tabela 5.1 – Pessoas sujeitas ao teste no Modo Bicicleta Estática Normal.....	41
Tabela 5.2 – Pacientes sujeitos aos testes no Modo Perna Debilitada	43

Lista de Acrónimos

AVC – Acidente Vascular Cerebral

BWS – Body Weight Support

RPM – Rotações Por Minuto

DAQ – Data Acquisition

USB – Universal Serial Bus

IMC – Índice de Massa Corporal

BPM – Batimentos por Minuto

DC – Direct Current

Capítulo 1

Introdução

Considerada por muitos como uma das grandes invenções de todos os tempos, a criação da bicicleta foi realmente um sucesso. É usada como meio de transporte, em competição, em atividades de lazer ou turísticas, para fortalecimento muscular e até para fins medicinais. Ora, tanta utilidade e de tão variadas formas, não espanta quando comparado com a quantidade de benefícios que acarreta: estimula a circulação sanguínea e reduz os níveis de colesterol; é uma poderosa arma de estimulação e tonificação muscular; é usada para perder peso; uma vez que, após a prática de exercício físico, o cérebro liberta endorfinas, pode ser usada para combater o *stress*; e pode também ser usada em reabilitação.

Esta Pedaleira Assistida Eletricamente para Reabilitação dos Membros Inferiores, tem como princípio o uso de uma bicicleta estática provida com um motor elétrico, sensores de força colocados nos pedais e sensores de batimento cardíaco. Através de um computador e do *software* Matlab/Simulink é feita a manipulação do controlador do motor, permitindo assim que sejam desenvolvidos novos modos adaptados a problemas específicos do paciente. Este sistema, visa auxiliar na reabilitação de doentes com patologias nos membros inferiores.

O objetivo desta tese é melhorar a interface direcionada ao terapeuta tanto a nível funcional, com a criação de novos modos de funcionamento que confirmam a este sistema maior utilidade, como a nível estético, tornando a interface mais intuitiva e fácil de utilizar.

1.1 Motivação

Os membros inferiores podem sofrer diversas lesões que podem estar localizadas ao nível da anca, joelho, tornozelo ou pé. Deve-se também ter em conta outros tipos de patologias que possam vir a afetar os membros inferiores como é o caso dos Acidentes Vasculares Cerebrais (AVC). Estas lesões podem provocar grandes dificuldades não apenas físicas, mas também do foro psicológico, alterando o estado emocional de uma pessoa e, deste modo, interferindo na sua qualidade de vida.

Na tentativa de reabilitação e suporte destas patologias, foi desenvolvida a bicicleta com pedaleira ativa em questão de forma a conseguir desenvolver planos de reabilitação específicos de

acordo com o problema do paciente e assim, diminuir o tempo da lesão e fortalecer a estrutura muscular do paciente.

1.2 Objetivos

Esta dissertação tem como objetivo, através do controlo de velocidade e binário do motor DC, criar novos perfis de força ajustando a velocidade de referência do motor da bicicleta estática ativa. Desta forma, pretende-se conceber novos modos de funcionamento adequados a diferentes tipos de pacientes, bem como aperfeiçoar os já existentes.

A grande área de aplicação desta bicicleta ativa será a reabilitação e como tal será esse o grande foco deste projeto. Apesar disso, não se deve deixar de ter em conta a necessidade de abranger o máximo de potenciais utilizadores e como tal deverão ser também criados novos modos de funcionamento para assim tornar a bicicleta tão completa quanto possível. Só desta forma este produto pode ser realmente útil não só em reabilitação mas também por desportistas ou outro tipo de utilizadores.

No final pretende-se que, além de ter modos de funcionamento com grande utilidade e realmente funcionais, tenha uma interface apelativa, simples e acima de tudo intuitiva, que cativa tanto o terapeuta como o paciente a utilizá-la muitas vezes durante todo o processo de reabilitação, não só pelo aspeto mas também pelos bons resultados que possa vir a produzir no paciente.

1.3 Estrutura da Tese

Esta dissertação é composta por 6 capítulos:

Capítulo 1 - Neste capítulo são expostos os fatores que motivaram a realização desta dissertação bem como os objetivos que se visam atingir com a mesma.

Capítulo 2 - Aqui são apresentados tipos de patologias que levam à necessidade de utilizar um sistema deste tipo, bem como os produtos com fins idênticos existentes no mercado.

Capítulo 3 - É feita uma descrição da arquitetura do sistema. A descrição divide-se em duas partes (o hardware e as interfaces) e corresponde ao ponto de situação aquando do início deste projeto.

Capítulo 4 - É neste capítulo que se encontra descrito tudo o que foi desenvolvido e implementado nesta tese. Desde os motivos que desencadearam as alterações até à forma como foram implementados os novos modos, tudo é descrito aqui de forma detalhada.

Capítulo 5 - Nesta secção são descritos os testes que fundamentam a real utilidade e correto funcionamento dos modos de funcionamento que foram desenvolvidos.

Capítulo 6 - No último capítulo são expostas as conclusões obtidas e são também apresentadas propostas que podem ser postas em prática em trabalhos futuros.

A secção de anexos divide-se em 3:

Anexo A - Contém os *datasheets* do motor, controlador e sensores de força.

Anexo B - Onde é apresentado o restante código, desenvolvido em Matlab, que não aparece no capítulo referente ao *software* desenvolvido.

Anexo C - Aqui pode ser consultado o esquemático em Simulink.

Capítulo 2

Estado da Arte

2.1 Patologias que Levam à Necessidade de Utilizar a uma Pedaleira de Reabilitação

Segundo a Sociedade Portuguesa de Medicina Física e de Reabilitação, a Reabilitação define-se como “um processo dinâmico pelo qual o indivíduo que sofreu incapacidade adquire o conhecimento e as competências técnicas necessárias para uma função física, psicológica e social otimizada”. Permite assim apoiar os indivíduos incapacitados, criando soluções para as inevitáveis restrições impostas por lesões, na tentativa da sua reintegração na sociedade, facilitando a sua autonomia e independência [1].

Na maioria dos países europeus, a prevalência da incapacidade atinge cerca de 10% da população. Um dos fatores deste nível de incapacidade pode dever-se ao envelhecimento populacional, com aumento da sobrecarga assistencial e custos aumentados ao nível dos cuidados de saúde e serviços sociais. Tem-se verificado uma melhoria em termos de cuidados de saúde e a população europeia tem cada vez melhores expectativas relativamente à manutenção de um bom estado de saúde. Assim, a prevenção das complicações da imobilização, das lesões cerebrais e da dor proporciona muitos benefícios, havendo necessidade de procurar novas formas de tratamento [2].

A reabilitação de sequelas ao nível do membro inferior é uma das importantes áreas de intervenção tanto das especialidades médicas de Medicina Física de Reabilitação e Reumatologia, como da especialidade de Ortopedia, onde o equipamento desenvolvido apresenta uma enorme vantagem em termos de reabilitação nas diversas patologias ou situações clínicas que podem surgir. Estas lesões e patologias podem estar associadas a défices motores que produzem fraqueza muscular e perda de funcionalidade do indivíduo, incluindo perda da capacidade de marcha e de autocuidados [2]. Nestas lesões e patologias pode incluir-se [2]:

- Doenças do foro musculoesquelético, tais como: ruturas musculares, tendinites, bursites, artrose, lesão ligamentar, muitas vezes associadas a lesões ocupacionais e desportivas, com necessidade de imobilização.

- Problemas neurológicos com sequelas de imobilização, como AVC, patologias neurológicas incapacitantes crónicas, nomeadamente esclerose múltipla e doença de Parkinson, paralisia da medula espinhal; alterações neuromusculares e miopatias; neuropatias periféricas, incluindo de Guillain-Barré;
- Períodos prolongados em que o doente se encontra acamado, que descondicionam os doentes e provocam a perda de funcionalidade física e psicológica;
- Doenças do foro reumatológico associadas a patologia inflamatória, como a Artrite Reumatoide e Artrite Psoriática;
- Reabilitação de pós-operatório de fraturas ósseas, associadas a síndromes de imobilização com descondicionamento.

2.2 Outros Sistemas para Reabilitação dos Membros Inferiores

A reabilitação pode-se considerar como uma combinação de medicação, manipulação física e exercícios terapêuticos adaptados a cada paciente. O exercício específico que melhor se enquadra ao paciente bem como o doseamento dessa carga, são atribuídos por profissionais de saúde responsáveis por ir acompanhando e fomentando a sua evolução.

Com o intuito de retirar ideias e perceber o que pode ser realmente útil a este sistema, fez-se uma pesquisa sobre sistemas desenvolvidos com a finalidade de reabilitar os membros inferiores:

2.2.1 *“An Intelligent Thermal Sensing System for Automatic, Quantitative Assessment of Motion Training in Lower-Limb Rehabilitation”*

É um sistema “ciber-físico” desenvolvido para ser utilizado em casa do paciente. Visa melhorar a capacidade de coordenação de movimentos através do treino físico. Este sistema avalia o estilo de movimento das pernas, a periodicidade da trajetória e o equilíbrio dos pés mediante acompanhamento do fisioterapeuta. Os movimentos são registados por câmaras térmicas e a pressão plantar é medida com base em sensores de pressão colocados na planta do pé. Foram

também desenvolvidos algoritmos para extrair, das imagens térmicas, os esqueletos das pernas e implementar auto segmentação (ver figuras 2.1 a 2.3) [3].

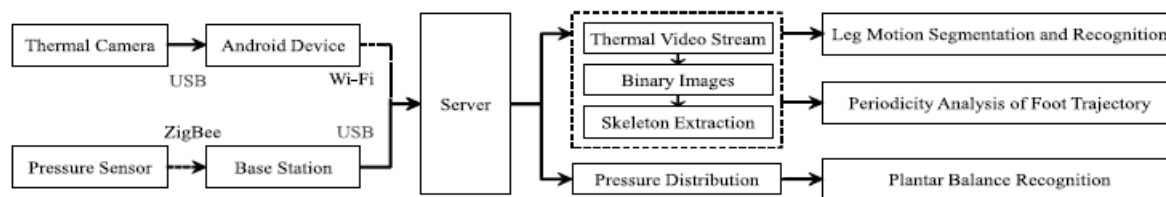


Figura 2.1 - Arquitetura do Sistema [3]

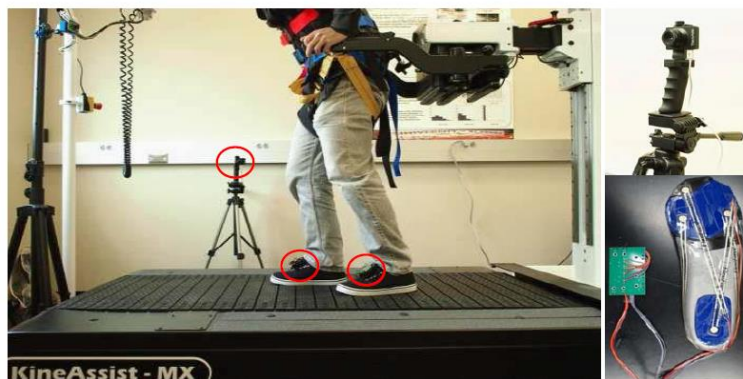


Figura 2.2 - Configuração experimental (com KineAssist robot treadmill) e os sensores [3]

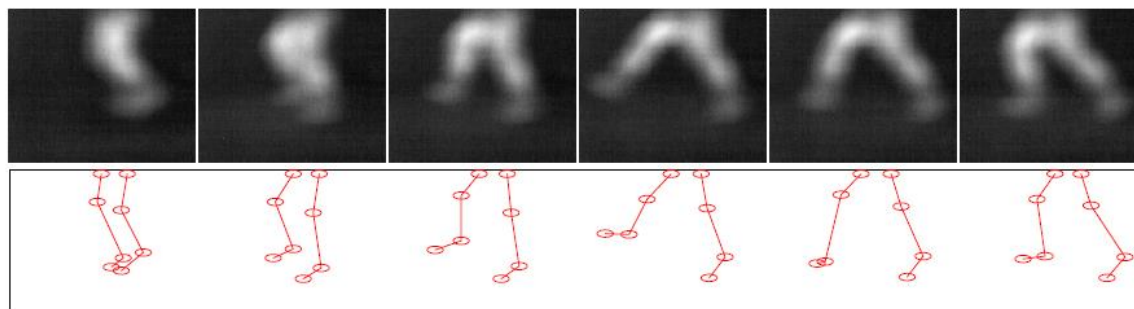


Figura 2.3 - Imagens térmicas e o esqueleto das pernas extraído na atividade experimental [3]

2.2.2 “A Wheelchair Integrated Lower Limb Exercise/Rehabilitation System: Design and Experimental Results on the Knee Joint”

A reabilitação assistida por robots tem como vantagens: 1) Pode reduzir significativamente a carga do fisioterapeuta durante o treino, permitindo assim que os pacientes possam executar uma quantidade suficiente de movimentos repetitivos e de forma intensiva; 2) Permite uma análise e avaliação quantitativa sobre o desempenho do paciente [4].

Esta cadeira de rodas destina-se a pessoas com lesões na medula espinal ou após um AVC. Partindo do princípio que nestas situações os pacientes são facilmente expostos a problemas

secundários provocados pela mobilidade limitada, foi então pensado o sistema integrado de exercícios/reabilitação de membros inferiores com cadeira de rodas. É apresentado um novo conceito de *design* de um sistema robótico, permitindo que as pessoas gravemente incapacitadas realizem treinos automatizados de reabilitação física de forma independente durante as atividades diárias [4].

Este sistema consiste em 3 módulos principais: 1) Uma cadeira de rodas elétrica; 2) Elevador que levanta e suporta o peso corporal do paciente; 3) Um exosqueleto do membro inferior (ver figuras 2.4 e 2.5) [4].



Figura 2.4 - Sistema de reabilitação de membros inferiores com cadeira de rodas [4]

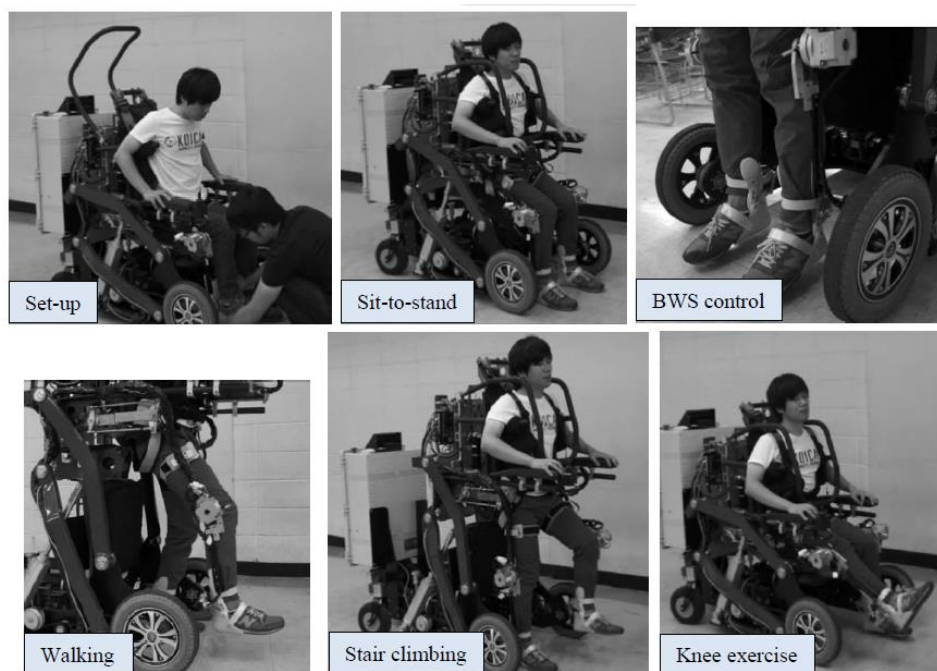


Figura 2.5 - Teste funcional e avaliação do sistema desenvolvido [4]

2.3.3 “Lower Limb Rehabilitation Evaluation System Based on Virtual Reality Technology”

Com a finalidade de compensar as lacunas dos métodos tradicionais, diversos grupos de pesquisa desenvolveram vários tipos diferentes de robôs para treino e reabilitação de membros inferiores. Na sua maioria estes sistemas apenas permitem realizar treino passivo não permitindo adequar a carga física à condição individual do paciente. Tendo em conta esta limitação, foi pensado este sistema de avaliação de reabilitação de membros inferiores para pacientes hemiplégicos, baseado na tecnologia da realidade virtual. Este sistema tem uma interface bastante intuitiva e permite ao médico ajustar o treino de reabilitação para diferentes pacientes e em diferentes níveis com base neste sistema de avaliação (ver figuras 2.6 e 2.7) [5].

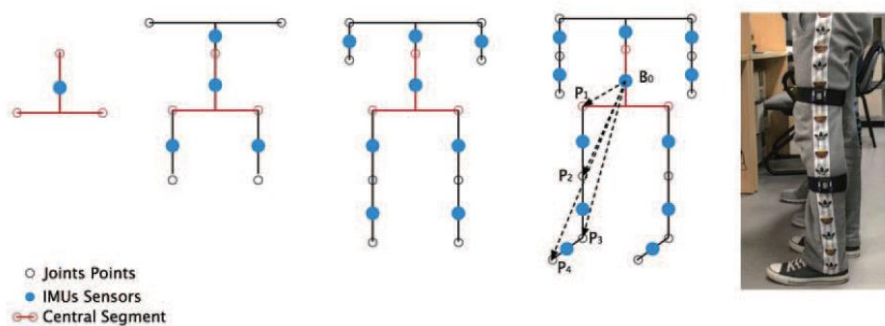


Figura 2.6 - Modelo eclético simplificado com sensores [5]

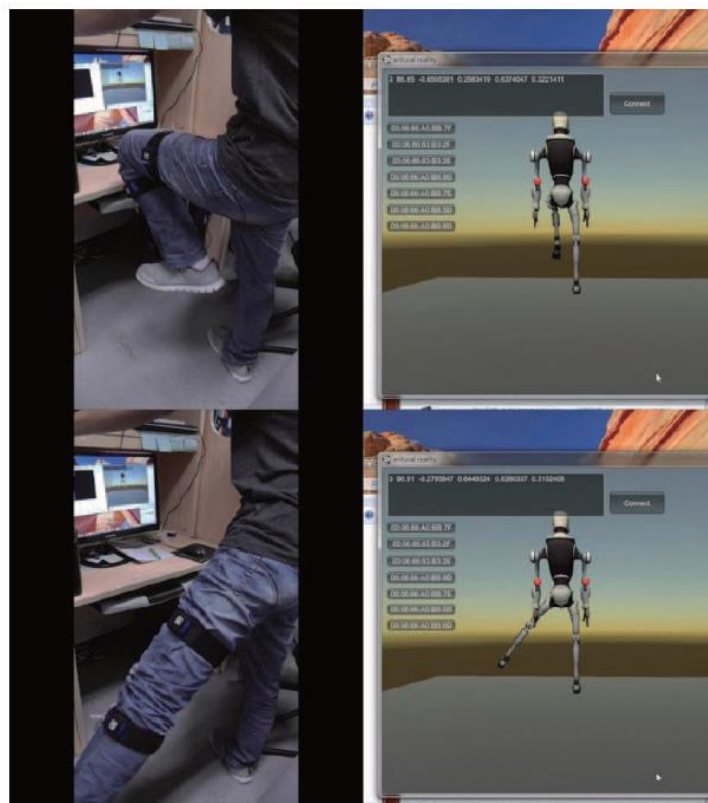


Figura 2.7 - Sistema experimental [5]

Capítulo 3

Arquitetura do Sistema e Interfaces

Foi desenvolvida uma bicicleta estática ativa com o objetivo de fazer reabilitação dos membros inferiores. Nesta bicicleta, a resistência habitual, variada pela caixa de velocidades, foi substituída por um motor elétrico controlado com base em sensores de força e de batimento cardíaco (ver figura 3.1).

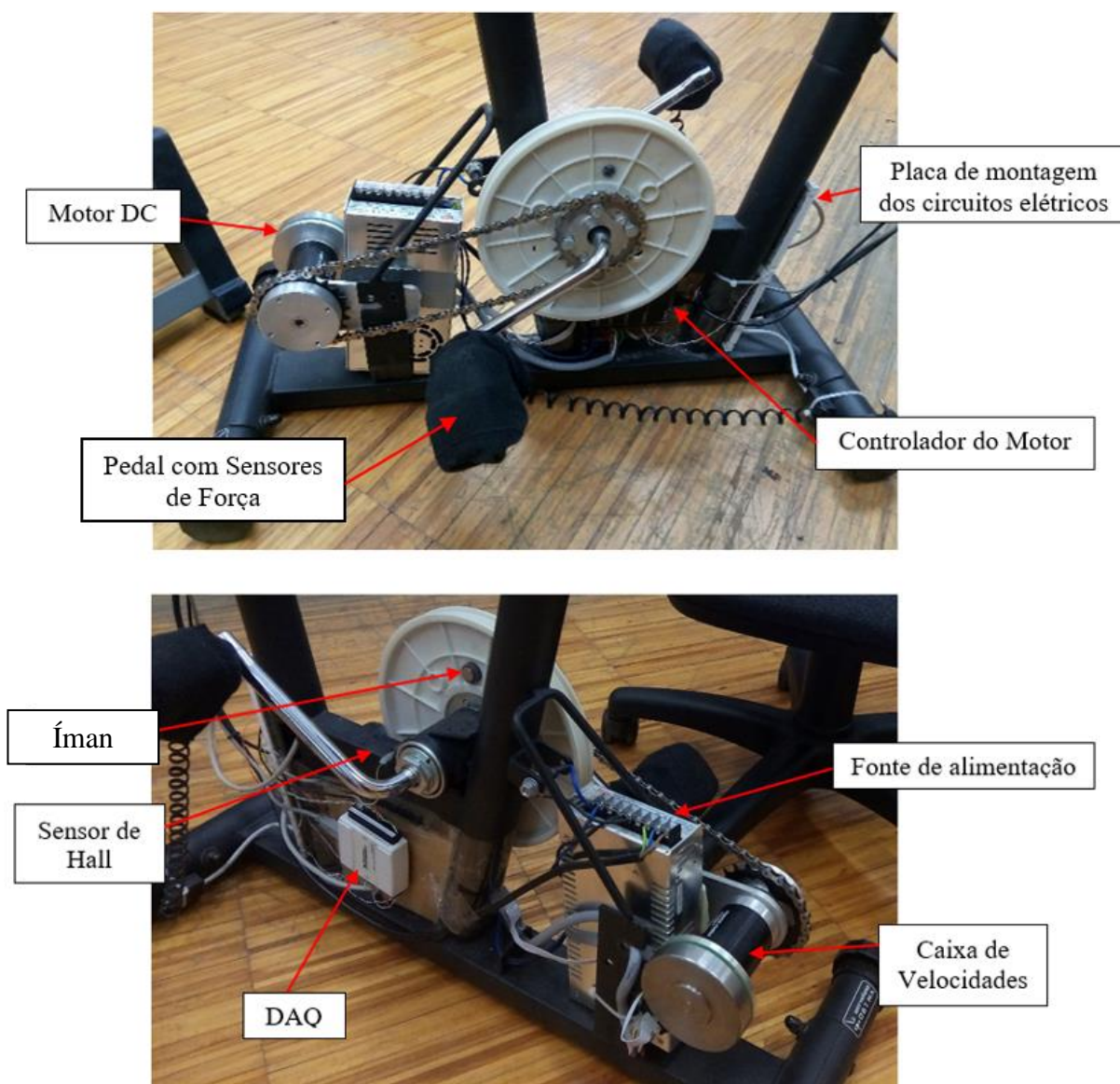


Figura 3.1 - Estrutura da Bicicleta Estática Ativa

3.1 Hardware

3.1.1 Motor EC 90 flat e Controlador Maxon ESCON

A bicicleta contém um motor EC 90 flat da Maxon (figura 3.2). Deste motor, como características principais, podem-se destacar os 24V de tensão, as 5000 rotações por minuto (rpm) de velocidade máxima e os 90 W de potência. No Anexo A tem o *datasheet* onde poderão ser consultadas todas as outras características do motor [6].

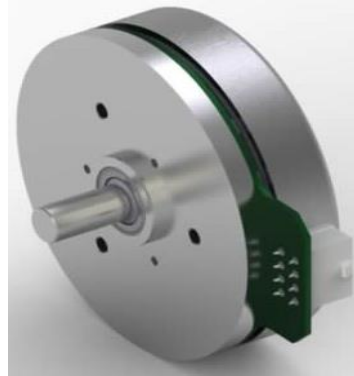


Figura 3.2 - Motor EC 90 flat [6]

Para controlar o motor, isto é, para gerir os valores de entrada existe o Controlador Maxon ESCON 50 (figura 3.3), do qual se poderá consultar todas as características no *datasheet* (presente no anexo A). Este controlador é manipulado através de um *software* da *National Instruments* chamado “ESCON *Studio*”, que permite controlar vários parâmetros como: a velocidade, o binário, as rampas de aceleração/desaceleração e o tipo de controlo desejado. Neste caso, foi utilizado o controlo de velocidade em malha fechada, sendo que os parâmetros são ajustados automaticamente através da opção “*autotuning*”.



Figura 3.3 - Controlador ESCON 50/5 [7]

3.1.2 Caixa de Velocidades

Com a finalidade de ajustar o binário e a velocidade entre a rotação da bicicleta e o motor, foi introduzida uma caixa de velocidades. A Planetary Gearhead GP 52 C (figura 3.4) tem uma redução de 53:1, o que significa que para cada 53 voltas dadas pelo motor a pedaleira da bicicleta efetua apenas uma volta. No Anexo A pode ser consultado o *datasheet completo* deste componente.



Figura 3.4 - Caixa de Velocidades (Planetary Gearhead GP 52 C) [8]

3.1.3 Sensores

3.1.3.1 Sensores de Força

Para este sistema foram implementados os sensores FlexiForce Modelo A201 da Tekscan (figura 3.5). São sensores impressos em circuitos ultrafinos e bastante flexíveis, o que lhes confere diversos tipos de aplicações:

- Detetar e medir força aplicada;
- Identificação de limites de força;
- Detetar e medir taxa de variação;
- Detetar contato.

Estes sensores são constituídos por duas camadas de substrato de poliéster onde, em cada camada, é aplicado um material condutor (prata) seguido por uma camada de tinta sensível à pressão, a ser aplicada na zona de leitura, definida por um círculo de prata na parte superior da tinta sensível. Esta tinta vai-se estender desde a área de sensibilidade aos conetores na outra extremidade do sensor, formando os fios condutores [9].

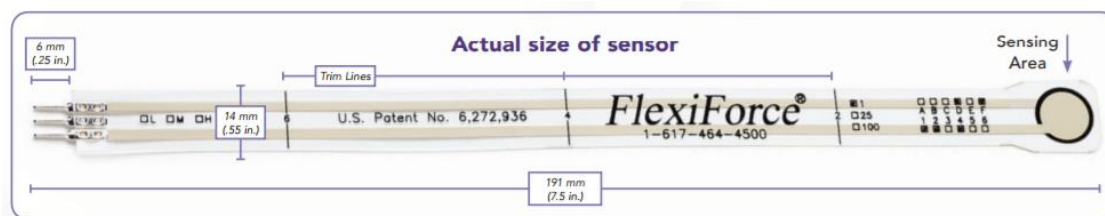


Figura 3.5 - Sensor FlexiForce Modelo A201 [10]

Em cada um dos sensores é colocada uma semiesfera em cima da zona de contacto de modo a melhorar a precisão centrando naquela zona a força exercida. Cada um dos pedais contém 3 destes sensores sendo que cada um deles funciona de acordo com a variação de resistência existente no circuito elétrico interno, ou seja, através da força que se exerce na zona sensível do sensor. A sua gama de força é variável entre os 0 N e os 445 N.

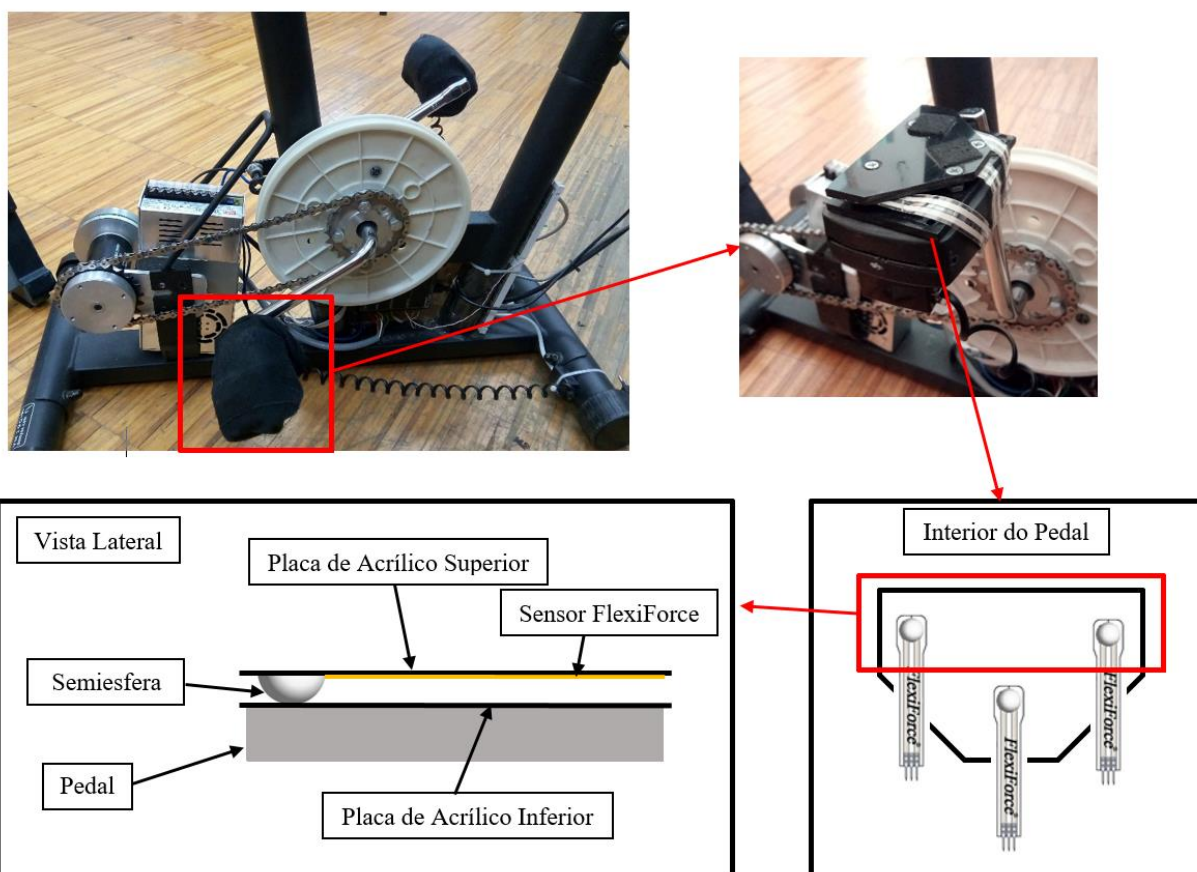


Figura 3.6 - Esquema de como estão montados os sensores nos pedais da Bicicleta

Estes sensores, aplicados a um circuito força-tensão, devolvem à saída (V_{out}) o valor de tensão que é aplicado na área de contato.

O circuito seguinte demonstra como estão aplicados os sensores:

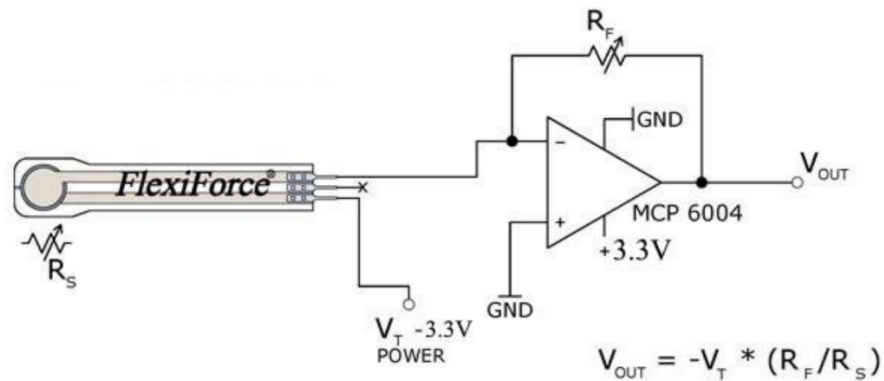


Figura 3.7 - Circuito de aplicação dos sensores [10]

R_s e R_f são a resistência do sensor e a resistência de saída, respetivamente. A tensão de saída é convertida para uma saída digital. O circuito satura caso a tensão de saída seja superior à tensão de referência (2.5V).

3.1.3.2 Sensor de Monitorização da Frequência Cardíaca

Partindo das variações de pressão nos vasos sanguíneos, ocorridos ao longo do ciclo cardíaco, é feita a deteção dos batimentos cardíacos.

Durante o ciclo cardíaco temos dois momentos: quando o sangue é bombeado para todo o corpo, provocando um aumento de pressão nos vasos sanguíneos (denominado como sístole) e durante a fase de relaxamento em que o sangue flui e volta novamente ao coração, através da diminuição da pressão ao nível dos vasos sanguíneos (denominado diástole) [11].

O volume de células no sangue é diretamente proporcional ao batimento do coração, isto é, quando o batimento aumenta o volume das células também aumenta e quando o batimento diminui o volume das células diminui. Esta reflexão é convertida num impulso, (tanto poderá ser de corrente como de tensão, dependendo do sensor) e é então processada [12].

Neste caso foi utilizado o fototransistor SFH 303FA (figura 3.8), sendo que se incidir muita luz o mesmo tende a ter mais condução e deste modo verifica-se uma diminuição de tensão e aumento de corrente. Vai então haver uma proporcionalidade entre a tensão no coletor e a frequência cardíaca, criando a necessidade de um acondicionamento do sinal, uma vez que, sendo uma variação tão pequena desencadeia uma dificuldade acrescida aquando do processamento no *software* [12].

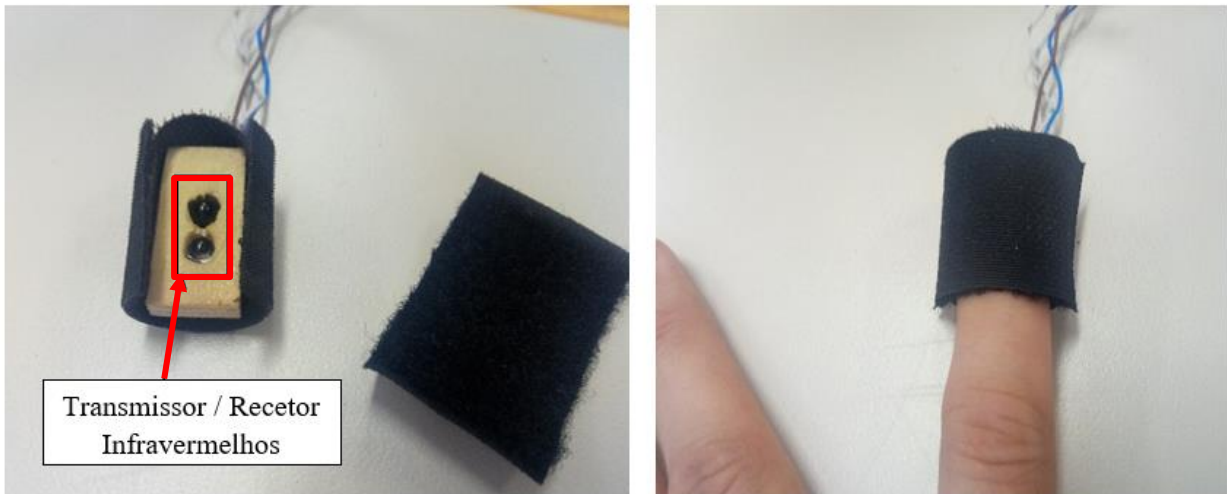


Figura 3.8 - Sensor de medição de Batimentos Cardíacos - Fototransistor SFH 303FA

3.1.4 Placa de Aquisição de Dados

Para adquirir os valores que são obtidos nos circuitos outrora implementados, foi adicionado à bicicleta o *Data Acquisition (DAQ) USB-6008* da *National Instruments* (figura 3.9).

Entre diversas especificações, destacam-se os seus 12 canais digitais (que podem ser de entrada ou saída de acordo com o pretendido), 2 saídas analógicas (12-bits, 150S/s), 8 entradas analógicas (12-bit, 10 kS/s) e uma entrada *Universal Serial Bus (USB)* que serve para alimentação e comunicação com o computador (não é necessária alimentação extra) [13].

Este equipamento vai adquirir dados, que são posteriormente transmitidos para o Simulink (do *software* Matlab) onde são tratados e processados de acordo com o que se pretende implementar.



Figura 3.9 - DAQ USB-6008 [13]

3.2 Interfaces

3.2.1 Interface Direcionada ao Terapeuta

De modo a permitir ao terapeuta definir os parâmetros a aplicar ao paciente com mais facilidade e a consultar os dados que estão a ser adquiridos pelos sensores durante o treino, foi desenvolvida a interface direcionada ao terapeuta:

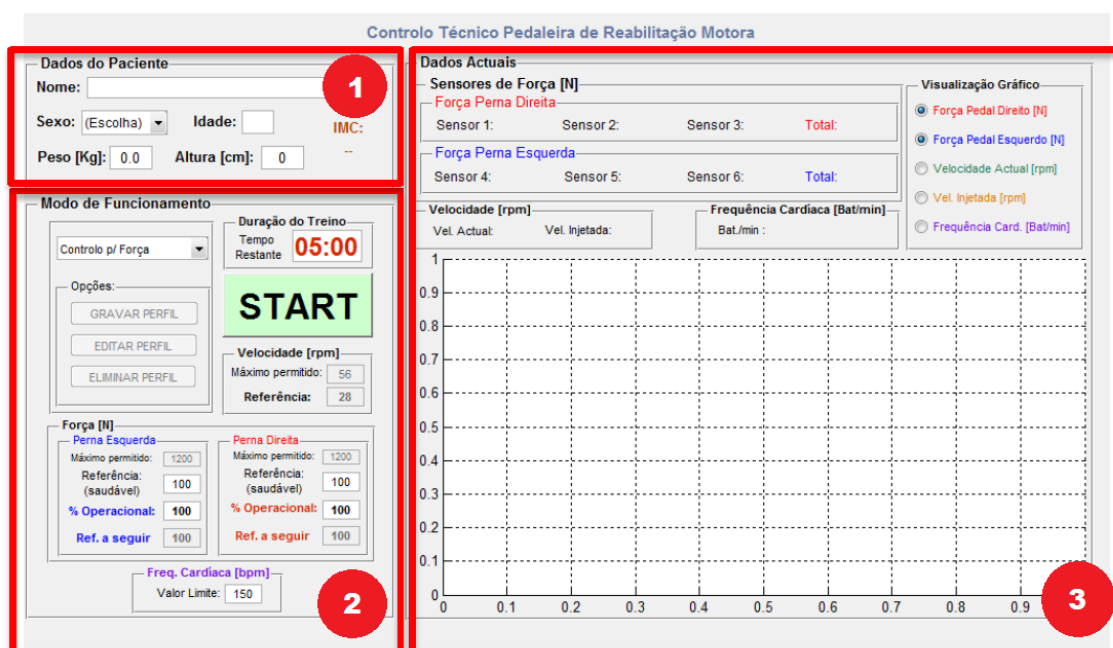


Figura 3.10 - Interface que será manipulada pelo terapeuta [12, 14]

Podemos considerar que a interface se divide em 3 painéis [12, 14]:

- 1) Campo com os dados do paciente: Nome, Sexo, Idade, Peso (kg) e Altura (cm). Com base no peso e altura é calculado o IMC.
- 2) Correspondente ao Modo de Funcionamento. Aqui o terapeuta pode alternar entre os modos de funcionamento existentes (Modo Força e Modo Velocidade). Pode também fazer variar diversos parâmetros de acordo com os modos escolhidos tais como: definir limites de força a aplicar e valores de referência a seguir, a velocidade a aplicar à bicicleta ou até mesmo limitar o valor de frequência cardíaca. Neste painel é ainda possível definir a duração ou iniciar e terminar o treino através do botão “STAR/STOP”. O menu opções que contém os botões com as opções de gravar, editar e eliminar perfil não têm associada nenhuma ação, portanto subentende-se que estão ali como uma possível sugestão para trabalhos futuros.

- 3) No 3º painel, o terapeuta pode consultar os dados atuais do paciente. Tais como, a força que o paciente está a exercer em cada um dos 3 pontos do pedal, a velocidade atual a que se está a mover e o número de batimentos cardíacos por minuto (bpm). Além destes parâmetros, que são exibidos numericamente, tem ainda um gráfico que lhe permite acompanhar e comparar todos esses dados, podendo exibir apenas as variáveis que pretende através do menu com *radio buttons* do lado direito.

3.2.2 Interface Direcionada ao Paciente

Com o intuito de estimular o paciente durante a reabilitação, foi-lhe desenvolvido um jogo (figura 3.11):



Figura 3.11 - Interface gráfica (jogo) direcionado ao paciente [14]

Esta interface gráfica contém um cenário com uma estrada e uma bicicleta numa floresta (a estrada foi desenhada no *software Cinema 4D* e a restante interface foi desenvolvida no *Matlab 3D World Edition* e *Cinema 4D*). A estrada vai dando a sensação de movimento e consoante o paciente, durante a pedalada, faça mais força num pedal do que noutro, a bicicleta tende a inclinar para o lado em que se está a fazer mais força. Quanto mais tempo o paciente conseguir manter a bicicleta na posição vertical, mais pontos ganha. Deste modo pretende-se que o paciente tente coordenar da melhor forma possível a força que está a exercer com cada um dos membros inferiores [14].

Capítulo 4

Software Desenvolvido

4.1 Alterações na Interface do Terapeuta

Como referido anteriormente (Capítulo 3, Interface Direcionada ao Terapeuta) já havia uma interface primitiva desenvolvida para ser manipulada pelo terapeuta e com a finalidade de este poder definir e variar os parâmetros de Força, Velocidade e Batimentos Cardíacos relativos ao paciente.

Após alguma utilização e também pelo *feedback* que foi recebido de pessoas que testaram este sistema, rapidamente se concluiu que a interface tinha diversas lacunas e devia ser melhorada. Foi também tido em conta que os grandes fatores de exclusão no mercado do *software* são a facilidade de habituação por parte dos utilizadores, isto é, o *software* ser intuitivo e de fácil interpretação. Posto isto, numa tentativa de proporcionar uma melhor experiência ao terapeuta ao utilizar esta interface, foram feitas as seguintes alterações:

4.1.1 Interface em Inglês

Cada vez mais as pessoas percebem várias línguas diferentes, o que leva a que haja cada vez menos barreiras linguísticas. O caso “sobe o tom” quando falamos de profissionais de saúde que, pela qualidade do ensino são desde cedo incentivados a falar outras línguas.

Segundo uma notícia de 2017 do *site “Fox News Point”*, a segunda língua mais falada no mundo é o Inglês sendo apenas ultrapassada pelo Chinês que surge em primeiro lugar no top 10 [15]. Posto isto, para dar mais alcance à interface e assim permitir que possa ser utilizada ainda por mais pessoas, optou-se por colocar toda a interface em Inglês.

4.1.1.1 Dados do Paciente, Modo de Funcionamento e Dados Atuais

Todo o texto desta interface foi então alterado para inglês sendo que na imagem abaixo é possível ver o resultado final:

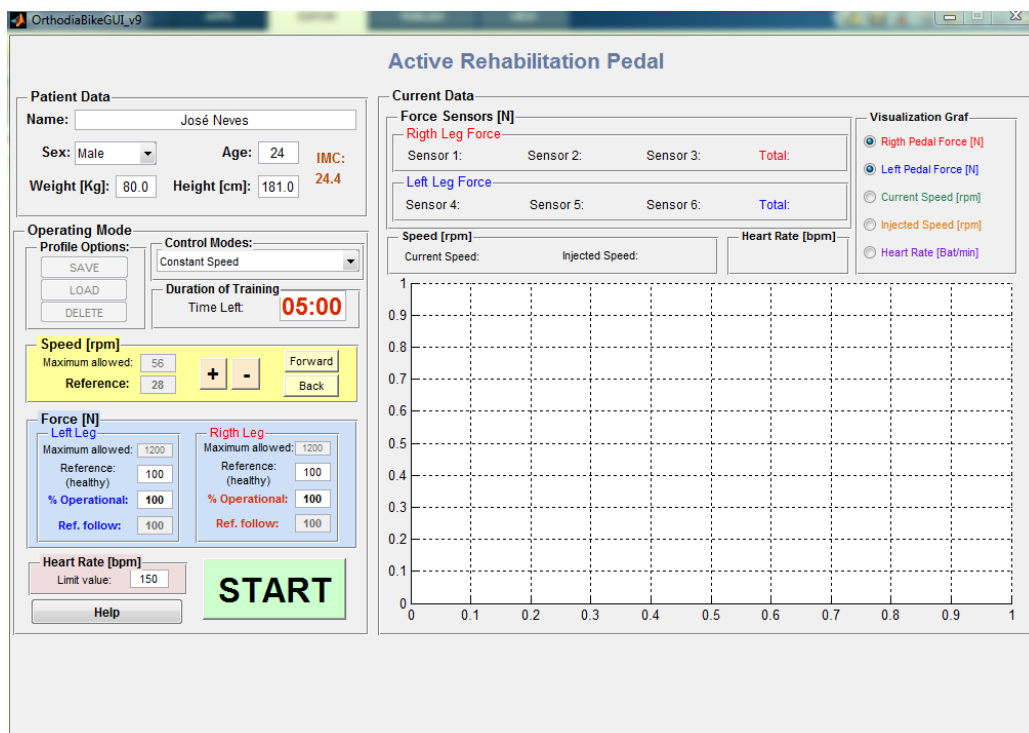


Figura 4.1 - Interface em Inglês

4.1.1.2 Mensagens de Erro

A interface tem vários campos de preenchimento e como em qualquer *software* tem que se precaver a introdução de dados errados que mais tarde possam comprometer o correto funcionamento de todo o sistema. Para tal, as mensagens de erro assumem aqui um papel de relevo, impedindo que o terapeuta introduza dados errados e lembrando quais os dados corretos e o intervalo aceitável a inserir.

Por isto mesmo, foram também traduzidas para inglês todas as mensagens de erro da interface e acrescentadas novas que se consideraram ser necessárias. A figura seguinte ilustra algumas das inúmeras mensagens de erro já em inglês.

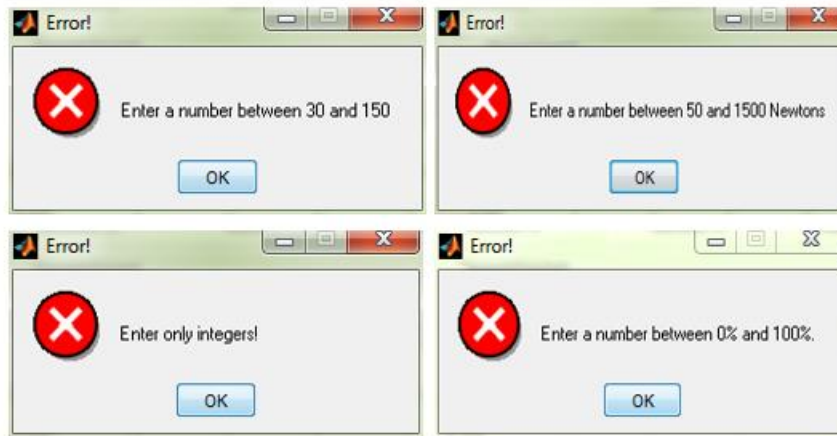


Figura 4.1 - Exemplos de janelas com mensagens de erro apresentadas pelo software

4.1.2 Reorganização do Painel *Operating Mode*

Como foi referido anteriormente (Capítulo 3, Interface Direcionada ao Terapeuta), a interface divide-se em 3 painéis. Concentremo-nos agora no bloco responsável por escolher o modo de funcionamento/*Operating Mode* (figura 4.2):

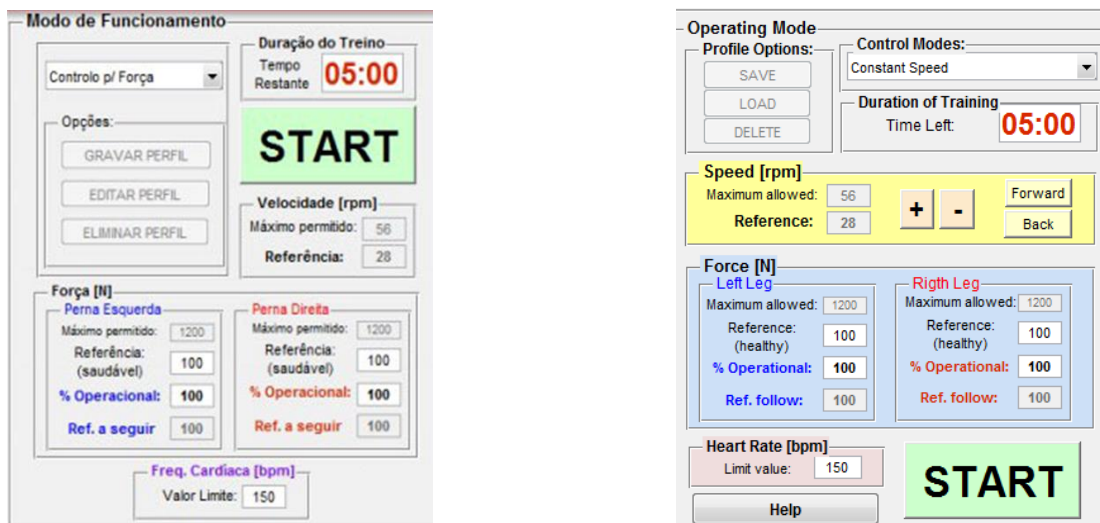


Figura 4.2 - À esquerda antes e à direita após as melhorias

Para tornar este bloco mais intuitivo optou-se por, no interior do painel, reorganizar os parâmetros que se iriam manipular por categorias para assim o utilizador seguir uma ordem lógica. No canto superior esquerdo foram reposicionadas as opções de guardar, carregar ou eliminar perfil, sendo que a escolha do local foi motivada pelo facto de todos os *softwares* terem estas mesmas opções disponíveis no menu “Ficheiro”, sempre disposto no canto superior esquerdo dos respetivos *softwares*. Logo à direita surge o modo de controlo e o tempo de duração, duas opções

essenciais uma vez que os restantes campos (seguintes) só serão manipuláveis consoante o modo escolhido. De seguida, surgem os campos referentes à velocidade, força e ao batimento cardíaco. Foi escolhido o amarelo para a velocidade, o azul para a força e o rosa para o batimento cardíaco pois são essas as cores utilizadas para esses parâmetros no gráfico que ilustra a sua variação. Só no final aparece o botão “*START*” para evitar que se inicie o sistema antes de definir todos os parâmetros pretendidos, problema esse que se verificou em testes com as versões anteriores da interface.

Desta forma, consegue-se que os campos de preenchimento fiquem dispostos de forma a que, aquando do preenchimento, o terapeuta possa seguir uma sequência lógica e estruturada evitando que perca tempo à procura dos campos pretendidos.

4.1.3 Opções do Perfil do Paciente

Ao analisar em pormenor o funcionamento da interface, o seu objetivo e de que forma as opções de guardar, carregar e eliminar perfil podiam ser proveitosas concluiu-se que, como estão a ser adquiridos simultaneamente valores de força, velocidades e batimentos cardíacos, seria muito útil, após o treino, ser possível guardar esses valores para posteriormente serem analisados pelo terapeuta. Outra grande utilidade seria, com o desenvolvimento de um modo de calibração, permitir guardar e carregar os dados de calibração de um paciente sem ter que repetir todo o processo novamente. Poderia também guardar os planos de treino para que mais tarde os pudesse repetir, evitando assim ter que repetir o preenchimento de todos os campos no início de cada treino.

Para ser possível fazer tudo isto foram desenvolvidas as opções seguintes (ver a figura 4.3):

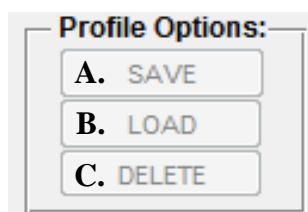


Figura 4.3 - Menu Opções de Perfil / Profile Options

A. SAVE

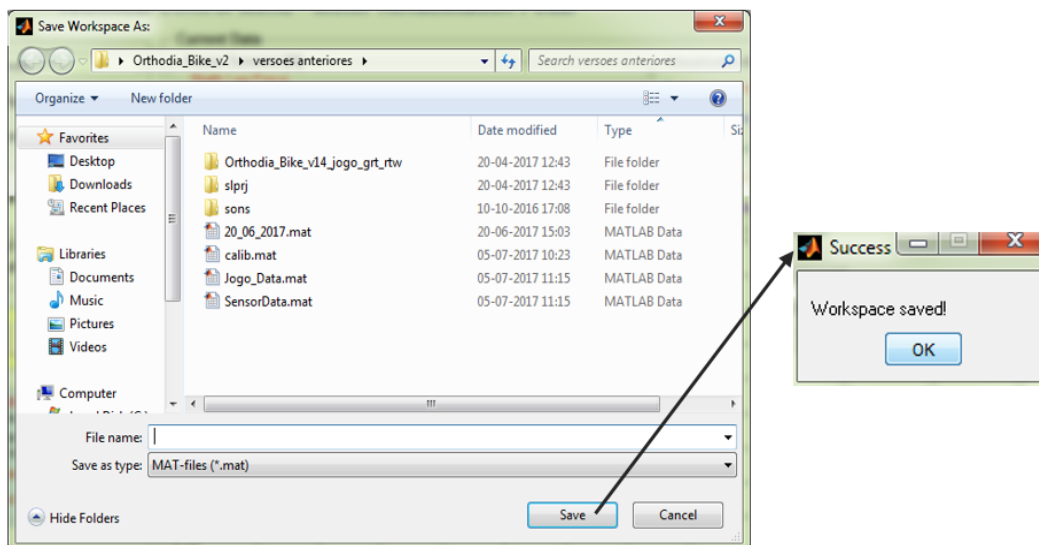


Figura 4.4 - Janela da opção SAVE e respetiva mensagem de confirmação

Esta opção permite guardar todas as variáveis do *Workspace* com um nome e localização pretendidos pelo utilizador [16]. O ficheiro guardado é do tipo “.mat”. Para o utilizador saber que o ficheiro foi guardado corretamente é exibida uma mensagem de confirmação (figura 4.4 à direita) [17].

B. LOAD

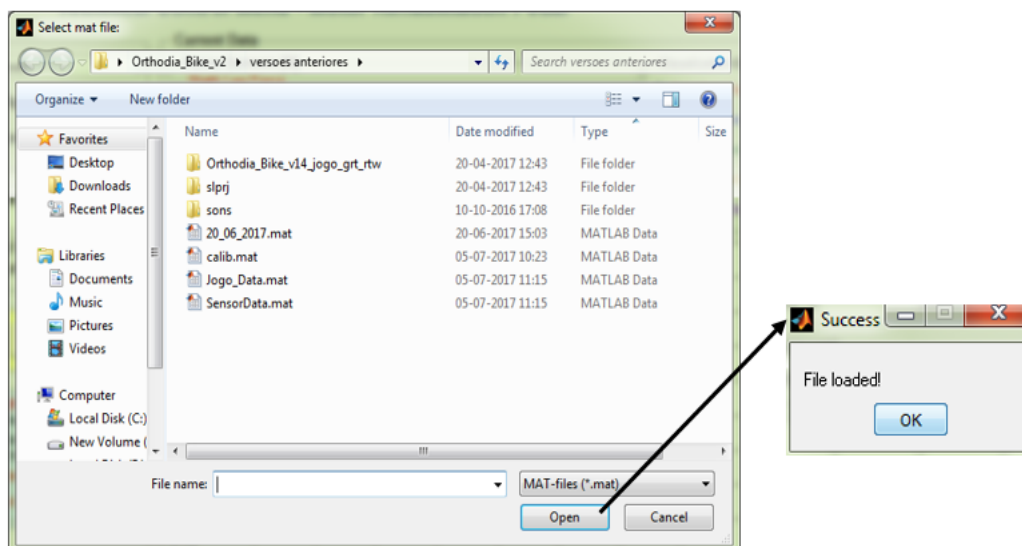


Figura 4.5 - Janela da opção LOAD e respetiva mensagem de confirmação

A opção *LOAD* permite ao utilizador carregar um ficheiro, do tipo guardado na opção *SAVE*, com a localização pretendida. Nesse caso, os campos são atualizados pelos valores do ficheiro que se está a carregar [18]. Após escolher o ficheiro e consequentemente se carregar em “*Open*” é exibida uma mensagem de confirmação.

C. DELETE

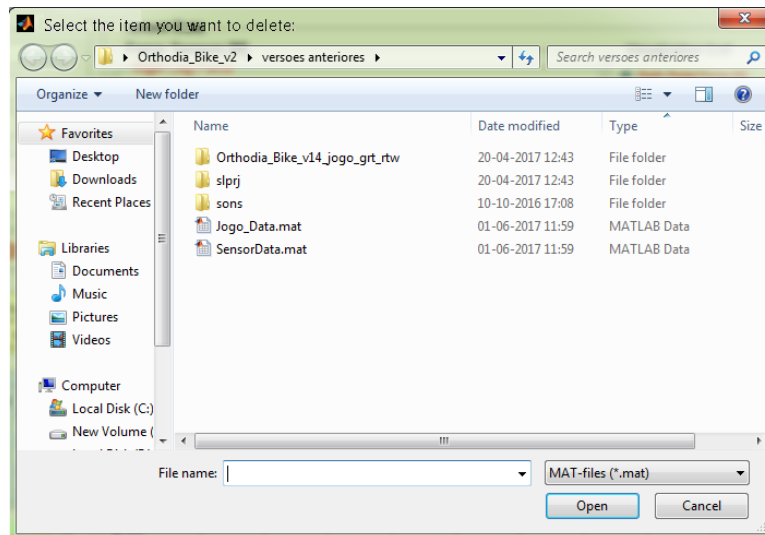


Figura 4.6 - Janela da opção DELETE

Esta opção tem como objetivo eliminar um “*Workspace*” guardado por engano ou que já não é necessário, evitando que o terapeuta tenha que sair da interface para eliminar um ficheiro [19]. Para precaver que se elimine um ficheiro acidentalmente, após se escolher o ficheiro a eliminar, é exibida uma janela de confirmação (figura 4.7) [17]:

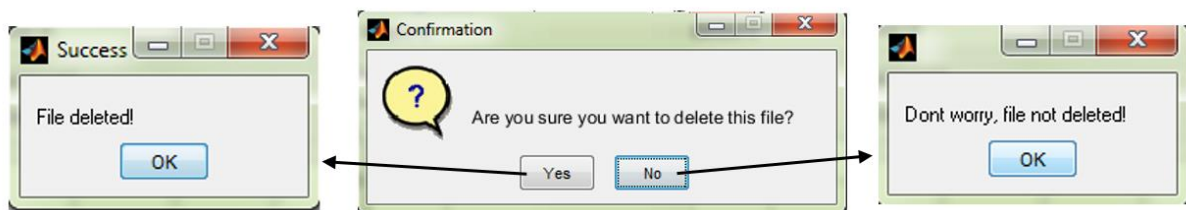


Figura 4.7 - Janelas de confirmação exibidas após escolher o ficheiro a eliminar

4.1.4 Botão de Ajuda

A criação do botão “*Help*” teve como propósito permitir, caso surja alguma dúvida ao terapeuta acerca do funcionamento de um modo específico, que este possa procurar ajuda dentro da própria interface e evitar assim que tenha que sair e ir ler um manual mais detalhado [17].

Esta opção está disponível tanto em Inglês como em Português (figuras 4.8 e 4.9):

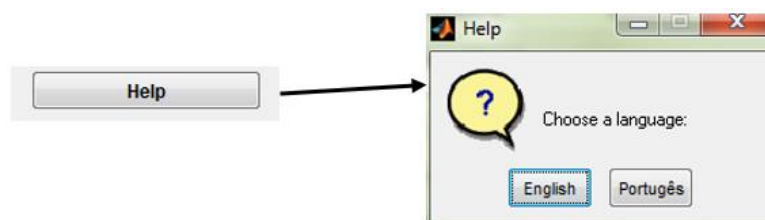


Figura 4.8 - Botão Help e respetiva janela que surge quando se carrega

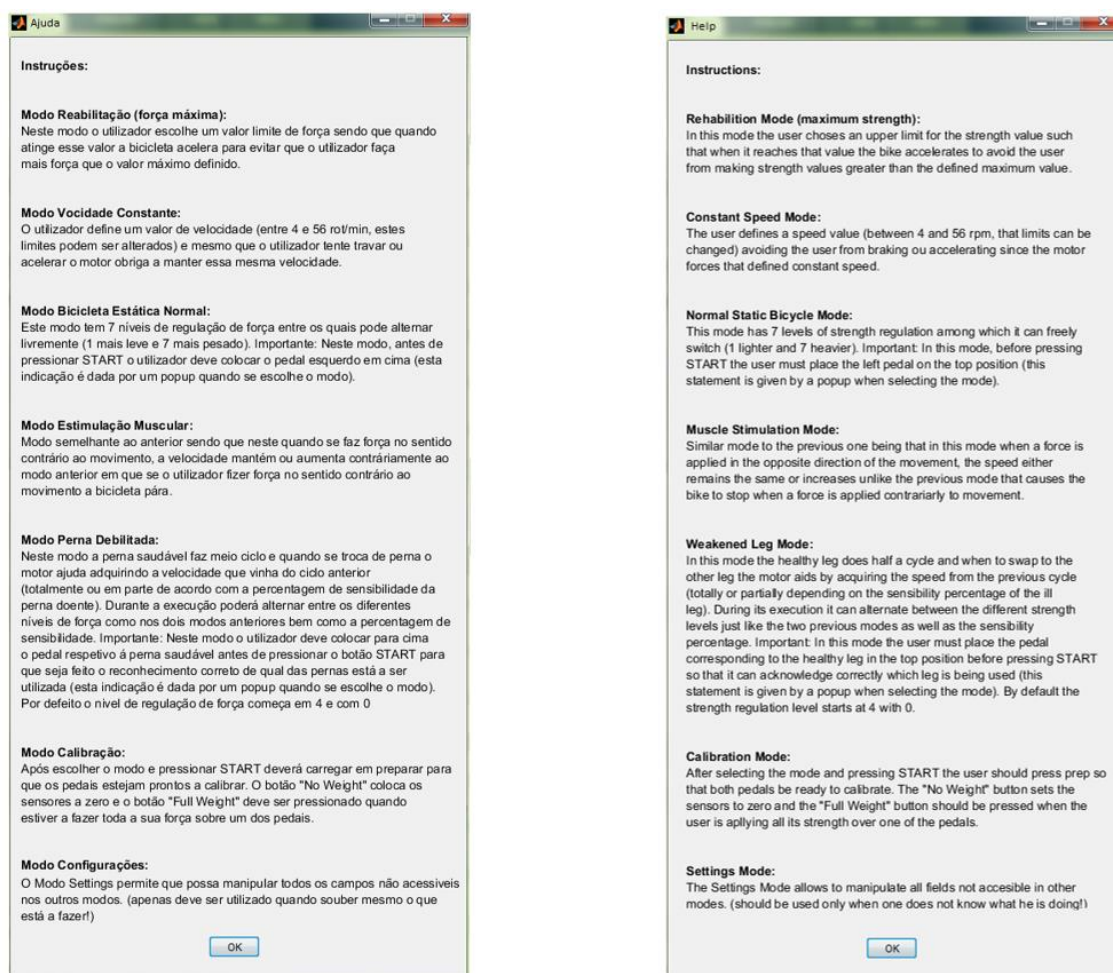


Figura 4.9 - À esquerda o Help em Português e à direita em Inglês

4.1.5 Instruções ao Escolher o Modo de Funcionamento

Para que todos os modos de funcionamento tenham o funcionamento desejado, é importante que o paciente inicie o movimento com os pedais numa posição específica. Para alertar o paciente e tentar assim evitar que posteriormente os modos não tenham o funcionamento desejado, criou-se (nos modos em que a posição de início é importante) um alerta que mostra em que posição devem ser colocados os pedais antes de iniciar o movimento [17, 20]. Estes alertas estão ilustrados nas figuras 4.10 e 4.11.

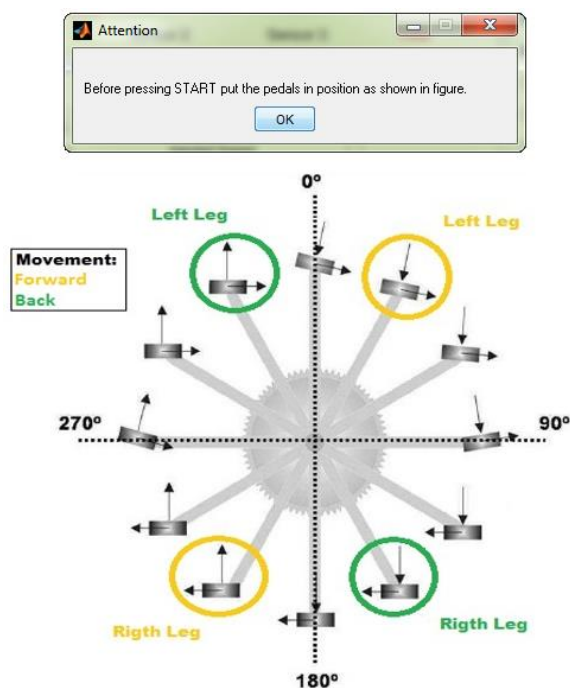


Figura 4.10 - Alerta ao iniciar o Modo Bicicleta Estática Normal (imagem adaptada de [21])

O alerta da figura anterior fica visível até que seja pressionado o botão “OK”. O esquema representa a vista da lateral direita da bicicleta. A amarelo é ilustrada a posição em que se deve colocar os pés caso o movimento seja para a frente e a verde caso o movimento seja para trás.

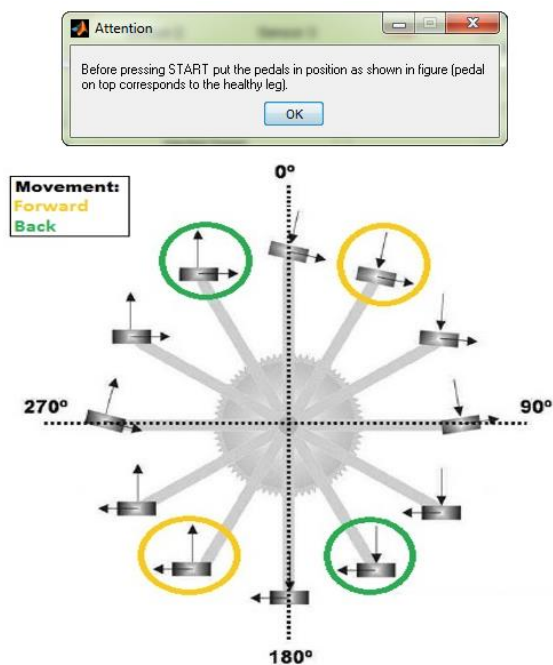


Figura 4.11 - Alerta ao iniciar o Modo Perna Debilitada (imagem adaptada de [21])

Neste caso não é dada a indicação de qual dos pés colocar em cada posição uma vez que no alerta já menciona que no pedal mais em cima (na posição de fazer força) deverá ser colocado o pé correspondente à perna saudável.

4.2 Modo Velocidade Constante

No Modo Velocidade Constante (*Constant Speed*) foram criados 4 novos botões (ver figura 4.12) com o propósito de tornar o modo mais funcional.

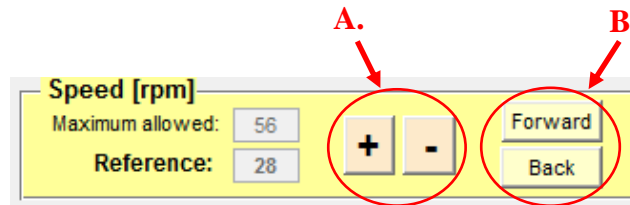


Figura 4.12 - Painel da interface referente ao modo Velocidade Constante

- A. Estes dois botões permitem incrementar (+) ou decrementar (-) velocidade (2 rpm) permitindo assim variar a velocidade sem ter que terminar e voltar novamente a iniciar a execução da interface.
- B. Adicionou-se também a opção de inverter o sentido do movimento, permitindo assim ao terapeuta alterar diretamente na interface o sentido de rotação dos pedais, de forma simples e bastante intuitiva.

Na implementação destes botões foi utilizada a função “set_param” do Matlab [22].

4.3 Modo Bicicleta Estática Normal

Tendo como objetivo aumentar o leque de potenciais utilizadores foi desenvolvido o Modo Bicicleta Estática Normal (Normal Static Bicycle). Neste modo pretende-se que, tal como numa bicicleta estática normal (em que a velocidade aumenta consoante se faz mais ou menos força nos pedais), o utilizador (ou neste caso o terapeuta) possa regular a resistência oferecida pela bicicleta durante o movimento simulando uma caixa de velocidades no percurso por uma estrada.

Relativamente à interface adicionou-se o painel seguinte que é exibido aquando da seleção deste modo:



Figura 4.13 - Painel que permite regular a força resistente exercida pela bicicleta

Quando o utilizador escolhe uma destas 7 opções a velocidade aplicada pelo motor é calculada pela fórmula descrita abaixo, em rpm:

$$velocidade = \left| \sum Forças Pedal_{direito} - \sum Forças Pedal_{esquerdo} \right| \times Regulação de Força \quad (1)$$

O somatório das forças corresponde à soma do valor lido pelos 3 sensores de cada pedal e a Regulação de Força é uma variável em que 1 corresponde a 90, 2 a 70, 3 a 50, 4 a 20, 5 a 5, 6 a 2.5 e 7 corresponde a 1 rpm/N. Estas constantes foram definidas com base em testes efetuados para perceber que valores se iriam enquadrar melhor em cada um dos níveis para que o utilizador sentisse a diferença de resistência imposta pela bicicleta ao variar entre eles. Por defeito colocou-se o sistema a iniciar no nível 4 que corresponde ao nível intermédio.

O valor final da expressão acima corresponde à velocidade no motor (antes da redução). Uma vez que o sistema tem uma caixa de velocidades de 53:1, se quisermos saber as rotações por minuto nos pedais teremos que dividir o valor por 53.

Durante a fase de testes percebeu-se que quando o paciente tentava contrariar o movimento da bicicleta na tentativa de travar, contrariamente ao que o mesmo pretendia, a bicicleta acelerava. Isto deve-se à diferença de forças ser em módulo. Para corrigir isto acrescentou-se uma condição que verifica qual o pedal que está para cima (através do sensor de hall e de 2 ímanes colocados de forma a perceber qual a posição dos pedais) e desta forma, se o paciente fizer mais força no pedal contrário é atribuído o valor zero à velocidade (como o controlador do motor tem definida uma rampa de desaceleração, a velocidade começa então a baixar até atingir o valor de 0 rpm).

A imagem seguinte mostra este modo em funcionamento:

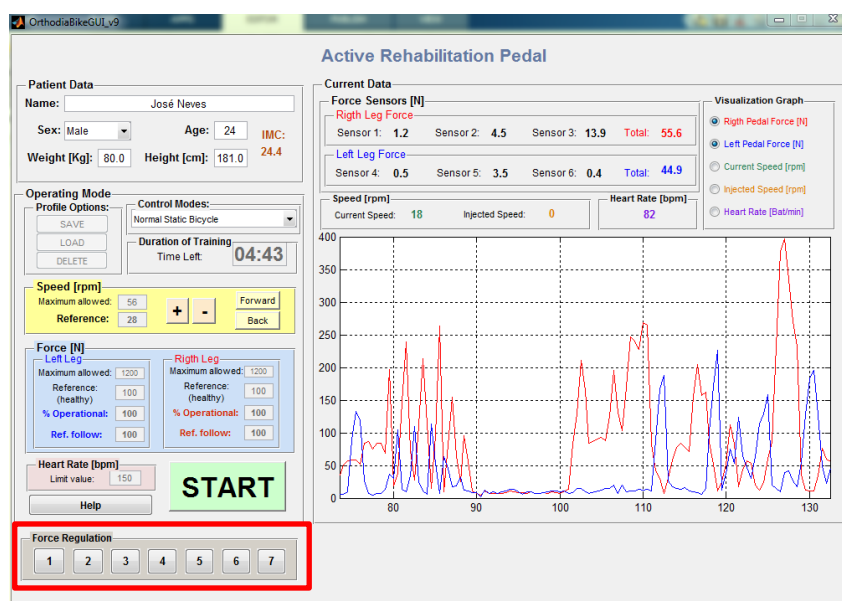


Figura 4.14 - Modo Bicicleta Estática Normal em funcionamento

4.3.1 Implementação

A figura seguinte demonstra como este modo foi implementado em *Simulink*:

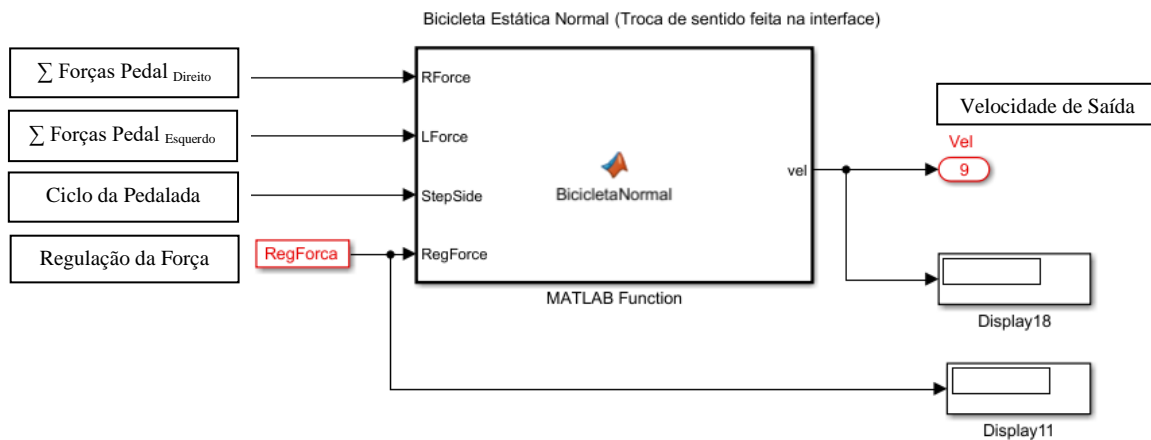


Figura 4.15 - Diagrama de Blocos, em Simulink, do Modo Bicicleta Estática Normal (com legenda)

O Ciclo da Pedalada (ou *StepSide*) pode ter os valores 1 (meio ciclo correspondente a fazer força com a perna esquerda) ou 2 (meio ciclo em que se faz força com a perna direita). Estas variáveis entram no bloco “Bicicleta Normal” (figura 4.15), que contém a equação (1), e da qual sai então o valor da velocidade do motor. É imprescindível que o paciente, antes de iniciar o movimento, coloque o pedal correspondente à perna esquerda para cima (como ilustra a janela de instruções do modo), para que assim a função tenha o funcionamento desejado.

Código contido no interior do bloco “Bicicleta Normal” da figura 4.15:

```
function vel = BicicletaNormal(RForce, LForce, StepSide, RegForce)
%StepSide=1 - Perna Esquerda a fazer força
%StepSide=2 - Perna Direita a fazer força

Force_Dif=abs(RForce-LForce);

%vel=k*(RForce-LForce)*RegForce;
%RegForce=1 -- tem que se fazer - força para obter maior velocidade
%RegForce=90 -- tem que se fazer + força para obter maior velocidade

if (RegForce==1 || RegForce==2.5) && Force_Dif<20           %Force_Dif<20 para que não ande mal se coloquem os pés nos
    vel=0;                                                    %pedais dando a sensação de ter que se fazer força
                                                             %para dar início ao movimento
elseif (RegForce==5) && Force_Dif<25
    vel=0;
elseif (RegForce==20) && Force_Dif<30
    vel=0;
elseif (RegForce==50) && Force_Dif<35
    vel=0;
elseif (RegForce==70 || RegForce==90) && Force_Dif<50
    vel=0;

%Para que se fizer força para trás a bicicleta não acelere
elseif StepSide==1 && RForce>LForce
    vel=0;
elseif StepSide==2 && LForce>RForce
    vel=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
else
    vel=RegForce*Force_Dif;
end
```

Figura 4.16 - Função Matlab no bloco “Bicicleta Normal” da figura 4.15

4.4 Modo Estimulação e Relaxamento Muscular

O modo de Estimulação e Relaxamento Muscular (*Muscle Stimulation*) foi pensado com o intuito de proporcionar ao paciente um modo que lhe permitisse relaxar e alongar os músculos dos membros inferiores tanto antes como após o treino.

Neste modo o paciente vai regulando a força resistente criada pela bicicleta durante a pedalada (os mesmo 7 níveis do modo anterior), mas neste caso, a bicicleta não terá o comportamento normal. Quando o paciente pedala (ou faz força) no sentido do movimento a bicicleta adquire mais ou menos velocidade consoante seja feita mais ou menos força nos pedais respetivamente. Quando tenta contrariar o movimento a bicicleta acelera também dando ao paciente uma sensação de relaxamento e permitindo deste modo alongar os músculos dos membros inferiores.

Este modo foi testado por vários utilizadores/pacientes e em todos eles a sensação de relaxamento e estimulação dos músculos foi unânime.

4.5 Modo Perna Debilitada

O Modo Perna Debilitada (*Weakened Leg*) foi desenvolvido como o objetivo de reabilitar pessoas que tenham sofrido um AVC, ou outra lesão que implique a perda total ou parcial da funcionalidade num dos membros inferiores.

Durante o processo normal da pedalada, cada um dos membros inferiores faz força durante meio ciclo. Para o caso de pacientes que não tenham uma perna 100% funcional, o motor deverá, no meio ciclo correspondente à perna debilitada, compensar com a percentagem em falta para que assim o paciente possa ter um movimento uniforme. Num caso extremo, em que o paciente não tem um dos membros, (ou mesmo tendo ambos os membros, tem 0% de funcionalidade num deles), o motor deverá compensar 100% no meio ciclo correspondente à perna debilitada.

A velocidade relativa ao meio ciclo da perna saudável é calculada pela equação (1), sendo que a velocidade correspondente ao meio ciclo da perna debilitada é calculada da forma seguinte:

$$\mathbf{velocidade}_{\text{Ciclo Perna Debilitada}} = \mathbf{velocidade}_{\text{compensação}} + \mathbf{velocidade}_{\text{Perna Debilitada}} \quad (2)$$

$$Velocidade_{compensação} = Velocidade_{meio\ ciclo\ anterior} \times (1 - Funcionalidade) \quad (3)$$

$$Velocidade_{Perna\ Debilitada} = ((|\sum Forças\ Pedal_{direito} - \sum Forças\ Pedal_{esquerdo}|) \times Regulação\ de\ Força) \quad (4)$$

Em que a Funcionalidade é uma constante entre 0 (correspondente a 0%) e 1 (correspondente a 100%) de acordo com a percentagem atribuída pelo terapeuta à perna debilitada e a velocidade_{meio ciclo anterior} corresponde à velocidade com que o paciente terminou o meio ciclo com a perna saudável.

Em suma, é medida a velocidade com que o paciente terminou o meio ciclo correspondente à perna saudável fazendo com que, de acordo com a percentagem de funcionalidade da perna debilitada, o motor compense com o restante necessário (por exemplo, um paciente com uma perna 40% funcional, terá que fazer 40% de força para manter a velocidade que vinha do ciclo anterior sendo que o motor será responsável por auxiliar com os 60% restantes).

Em termos de interface, o terapeuta pode regular a força relativa à perna saudável e a percentagem de funcionalidade da perna debilitada através dos painéis seguintes:

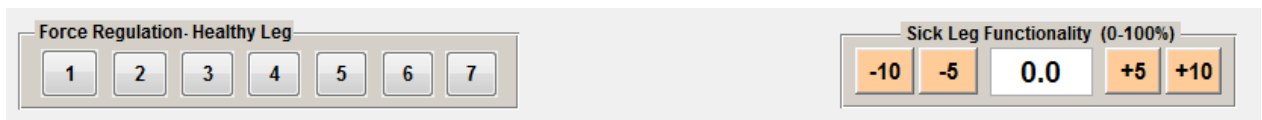


Figura 4.17 - Painéis que permitem alterar os níveis de Regulação de Força e Funcionalidade

4.5.1 Implementação

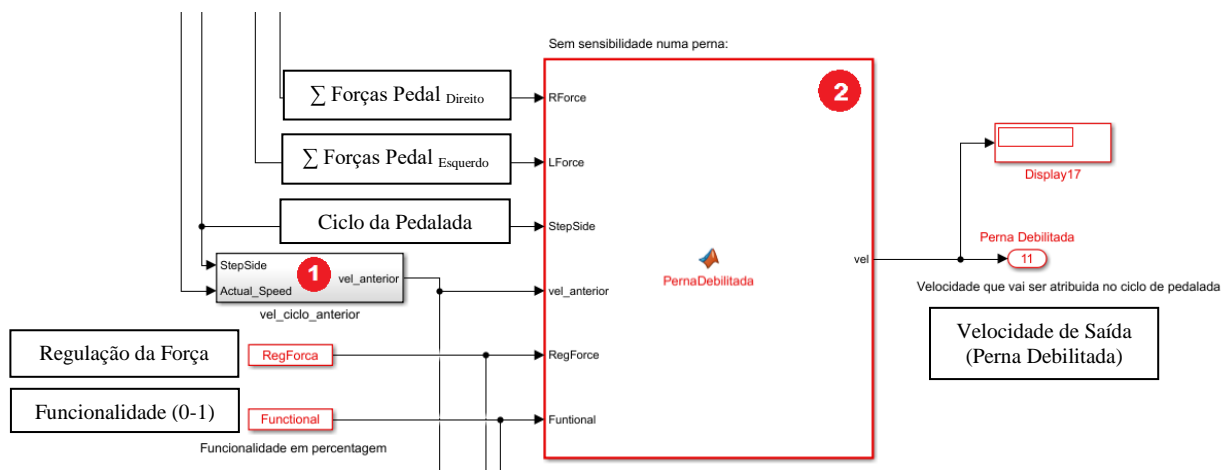


Figura 4.18 - Diagrama de Blocos, em Simulink, do Modo Perna Debilitada (com legenda)

No bloco assinalado com (1) é guardada a velocidade com que se terminou o meio ciclo feito com a perna saudável. Esse valor é armazenado na variável “vel_anterior” e mantém-se igual até que se chegue novamente ao meio ciclo da perna saudável [23].

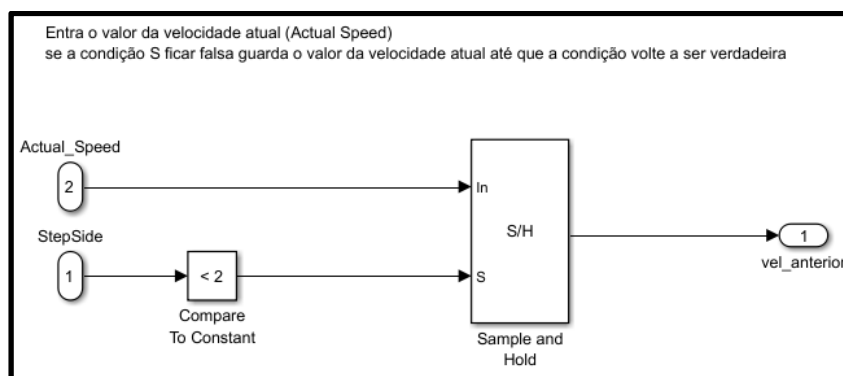


Figura 4.19 - Bloco (1) da figura 4.18

No bloco (2) introduziu-se a equação (2) bem como todas as condições necessárias ao correto funcionamento do modo (figura 4.20):

```
function vel = PernaDebilitada(RForce, LForce, StepSide, vel_anterior, RegForce, Funtional)
%StepSide=1 - Perna Boa em cima
%StepSide=2 - Perna doente a fazer força

Force_Dif=abs(RForce-LForce);
%vel=(Force)*RegForce;
%RegForce=1 -- tem que se fazer - força para obter maior velocidade
%RegForce=90 -- tem que se fazer + força para obter maior velocidade

vel=0;

if (RegForce==1 || RegForce==2.5) && Force_Dif<20 && StepSide==1 %Force_Dif<20 para que não ande mal se coloquem os pés nos
    vel=0; %pedais dando a sensação de ter que se fazer força
                                     % para dar início ao movimento
elseif (RegForce==5) && Force_Dif<25 && StepSide==1
    vel=0;
elseif (RegForce==20) && Force_Dif<30 && StepSide==1
    vel=0;
elseif (RegForce==50) && Force_Dif<35 && StepSide==1
    vel=0;
elseif (RegForce==70 || RegForce==90) && Force_Dif<50 && StepSide==1
    vel=0;

elseif StepSide==1 %quando stepside 1 (PERNA BOA), a vel é dada através da força pela formula abaixo
    vel=abs((Force_Dif)*RegForce);

elseif StepSide==2 %quando é para usar a perna "má"
    vel=(vel_anterior*(1-Funtional))+abs(Force_Dif)*RegForce;
    %o motor adquire a velocidade que vinha do ciclo de pedalada anterior e essa vel é multiplicado a (1 - %sensibilidade na perna doente)
    %Esse valor é somado 'a velocidade adquirida pela força feita pela perna doente * a percentagem de funcionalidade

end
```

Figura 4.20 - Função matlab do bloco (2) da figura 4.19

4.6 Modo Calibração dos Sensores de Força

Durante a realização de alguns testes, os 3 sensores do pedal esquerdo deixaram de funcionar e tiveram que ser substituídos (estes sensores são muito sensíveis e basta um mau contato nos *crimps* (partes metálicas do sensor que ligam as bandas do sensor à área de contato), ou até mesmo um sobreaquecimento durante a soldagem para que acabem danificados). Como os

sensores eram novos, optou-se por determinar uma nova reta de calibração para os sensores do pedal esquerdo. Para tal foi utilizado o material seguinte (figura 4.21):

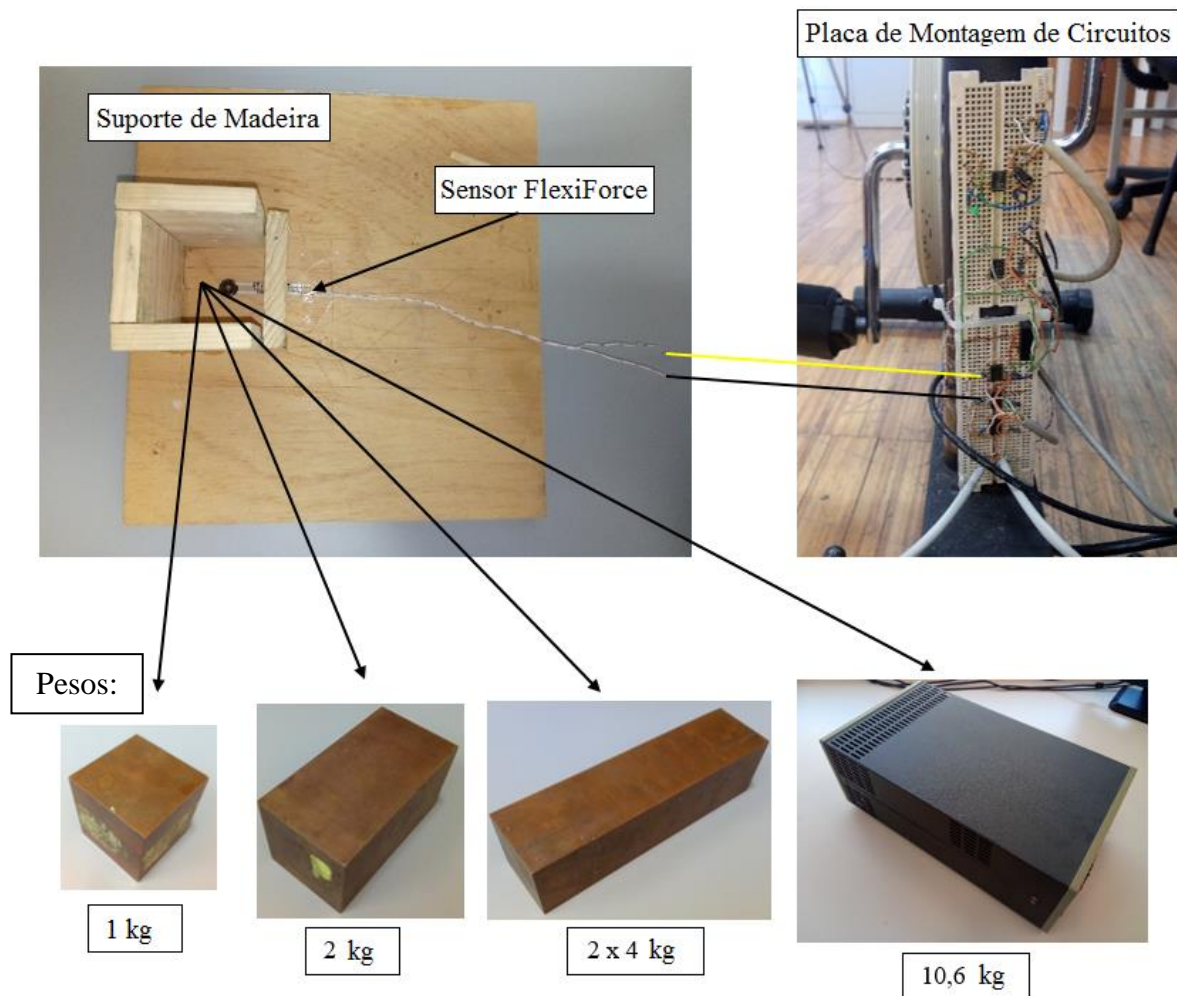


Figura 4.21 - Material utilizado no processo de calibração

Este processo consiste em colocar o sensor no suporte de madeira e liga-lo ao circuito existente na bicicleta. De seguida vão sendo colocados diferentes pesos no suporte e através do Matlab/Simulink vão sendo adquiridos e anotados os valores de tensão de saída a que corresponde cada um dos pesos. No final, já com todos os valores adquiridos, foi feita uma regressão linear (no *software* Microsoft Excel) para assim conseguir construir uma equação do valor da tensão em função do peso. Na calibração dos sensores do pedal esquerdo foram utilizados 5 pesos: 1kg, 2kg, 2 pesos com 4kg cada e 1 com 10,6kg. Estes 5 pesos permitiram recolher 13 amostras (1kg; 2kg; 3kg; 4kg; 5kg; 6kg; 7kg; 8kg; 9kg; 10kg; 11kg; 12,6kg e 13,6kg), não sendo necessárias mais amostras pois para pesos superiores a 13kg há saturação [12].

Deste modo obteve-se a seguinte função e consequente regressão linear (figura 4.22):

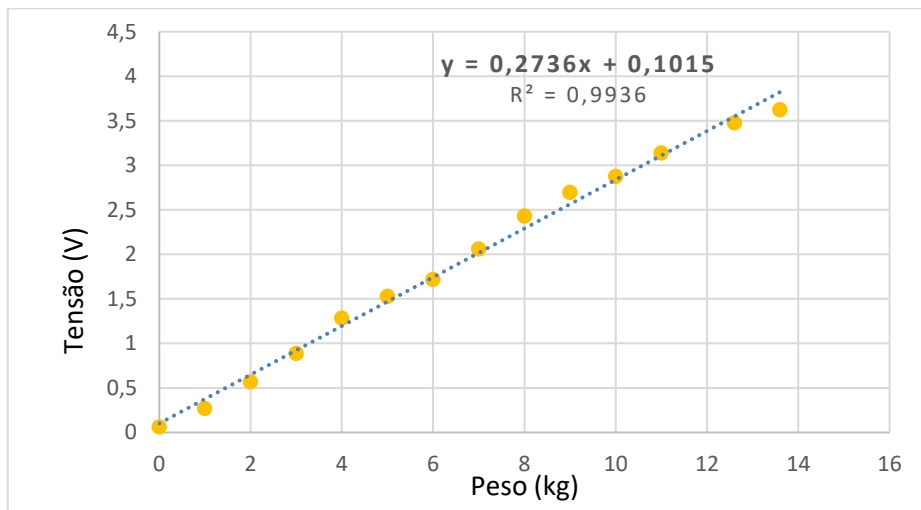


Figura 4.22 - Reta de calibração para os sensores do pedal esquerdo

Modo de funcionamento que permite calibrar sensores de força:

Durante o processo de aquisição, 6 sensores vão adquirindo os valores de força exercida por cada um dos membros inferiores. Estes valores são adquiridos pelo DAQ e posteriormente são enviados para o Matlab. Inicialmente constatou-se que, mesmo quando não se está a aplicar qualquer força sobre cada um dos pedais, os sensores marcam um valor de força maior que 0 (provocado pelo peso do suporte ou simplesmente por ter o pé apoiado no pedal). Embora fossem valores muito pequenos (na ordem dos 5 Newtons) poderiam não transmitir aos utilizadores a confiança necessária no sistema ou adulterar resultados de testes que se pretendessem vir a realizar, pois mesmo não se estando a fazer força dá a falsa sensação de os sensores estarem a ser pressionados.

Outra limitação que se verificou foi aquando da medição de forças mais elevadas. Por exemplo, quando a pessoa em teste colocava todo o seu peso em cima de um dos pedais, o valor medido em teoria deveria ser próximo do seu peso. Ora o que se verificava era um valor muito abaixo e, portanto, bastante díspar do pretendido.

Para corrigir estes aspetos foi desenvolvido o Modo de Calibração (*Calibration*). O utilizador escolhe esse modo, pressiona o botão “START” e o modo é então iniciado. Para dar início ao processo de calibração o utilizador deve carregar no botão “Prepare” para que a bicicleta permita que se manipulem os pedais e seja então feita a calibração. Abaixo segue-se uma imagem da interface com o Modo de Calibração selecionado:

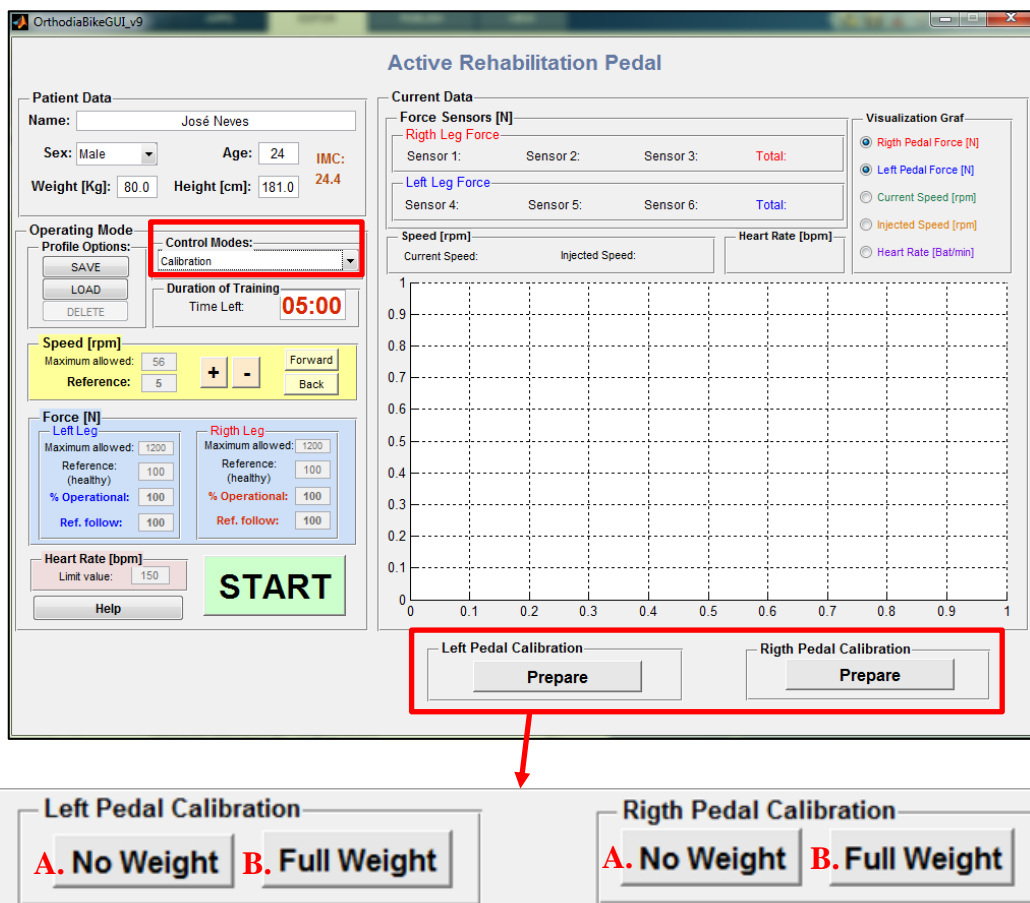


Figura 4.23 - Interface com o Modo de Calibração selecionado e painel de calibração

- A.** A opção “No Weight” permite, quando não é aplicada qualquer força sobre os pedais, que se coloquem todos os sensores com o valor de 0 Newtons, retirando assim o *offset*. Nesta situação, ao ser premido o botão “No Weight” é guardado o valor atual do sensor que vai ser subtraído a ele próprio com a finalidade de dar 0. O valor do sensor é então atualizado e, a cada medição feita a partir daí, o valor medido será sempre apresentado como o valor da medição atual subtraído pelo *offset*. Isto vai permitir que quando não se faz força os sensores marquem 0 N. Este processo é feito para cada um dos sensores (cada botão é responsável pelos três sensores do pedal respetivo).
- B.** A opção “Full Weight” possibilita que quando se aplica todo o peso da pessoa sobre cada um dos pedais (separadamente), estes assumam a força aplicada como sendo igual ao peso do paciente em Newtons. Na figura seguinte é ilustrado este processo:



Figura 4.24 - À esquerda o paciente coloca todo o seu peso sobre o pedal esquerdo e à direita sobre o pedal direito

Quando se está a aplicar todo o peso sobre o pedal respetivo deve-se carregar no botão “Full Weigth” e o valor que está a ser medido pelo sensor é guardado. É então aplicada a fórmula seguinte:

$$\sum \text{Sensores de Força do Pedal} \times \text{Constante}_{(\text{esquerda ou direita})} = \text{Peso} \quad (5)$$

$$\text{Constante}_{(\text{esquerda ou direita})} = \frac{\text{Peso}}{\sum \text{Sensores de Força do Pedal}} \quad (6)$$

A Constante_(esquerda ou direita) é multiplicada ao somatório dos três sensores respetivos a cada pedal.

4.6.1 Implementação

Na figura 4.27 é ilustrado como foi implementado o Modo Calibração em Simulink:

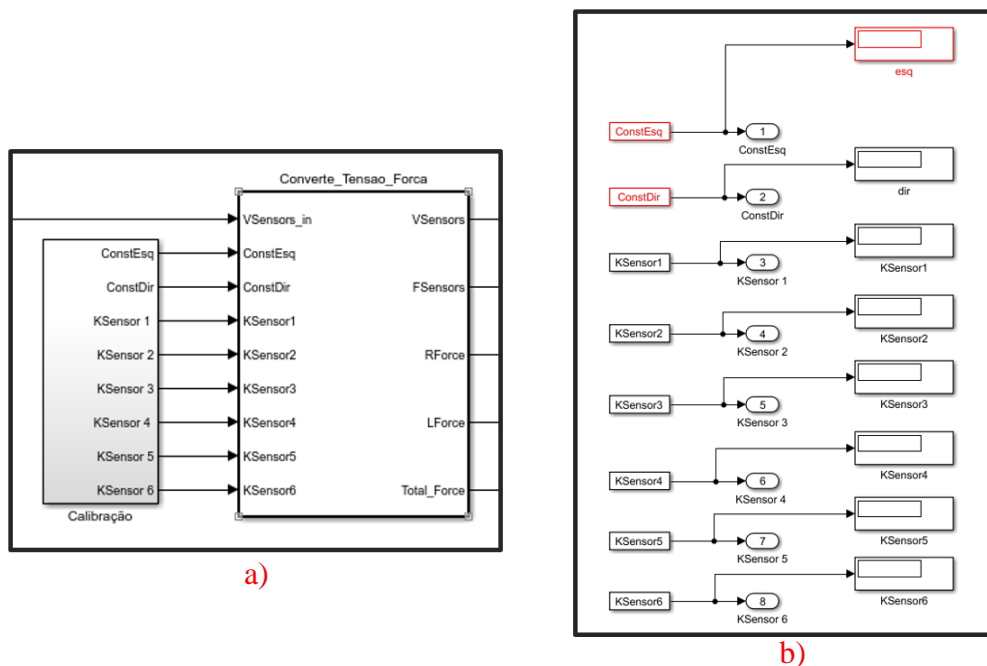


Figura 4.25 - a) Diagrama de blocos em Simulink do Modo de Calibração. b) Interior do bloco Calibração

No interior do bloco de calibração são guardadas as constantes obtidas na interface do terapeuta. Estas constantes vão entrar no bloco em que é feita a conversão do valor dos sensores de tensão para força. A “ConstEsq” e “ConstDir” são multiplicadas ao somatório dos 3 sensores do pedal esquerdo e direito respetivamente e as restantes constantes são subtraídas às regressões lineares de cada um dos 6 sensores dos pedais.

4.7 Modo Configurações

Com vista a preaver que o terapeuta, por lapso, altere parâmetros que possam vir a comprometer o correto funcionamento de todo o sistema, vários campos estão bloqueados durante a manipulação dos diversos modos de funcionamento. Para possibilitar manipular estes campos com segurança, foi desenvolvido o modo Configurações (*Settings*).

Ao seleccionar o modo é exibida a janela de alerta seguinte com o objetivo de chamar à atenção do terapeuta para o facto de estar a utilizar um modo que deve ser usado com precaução, visto que nele terá liberdade para manipular valores críticos [17]:

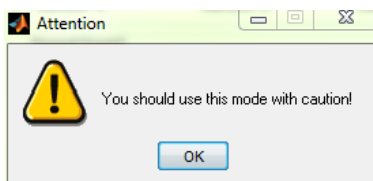


Figura 4.26 - Mensagem de aviso exibida ao seleccionar o modo Configurações/Settings

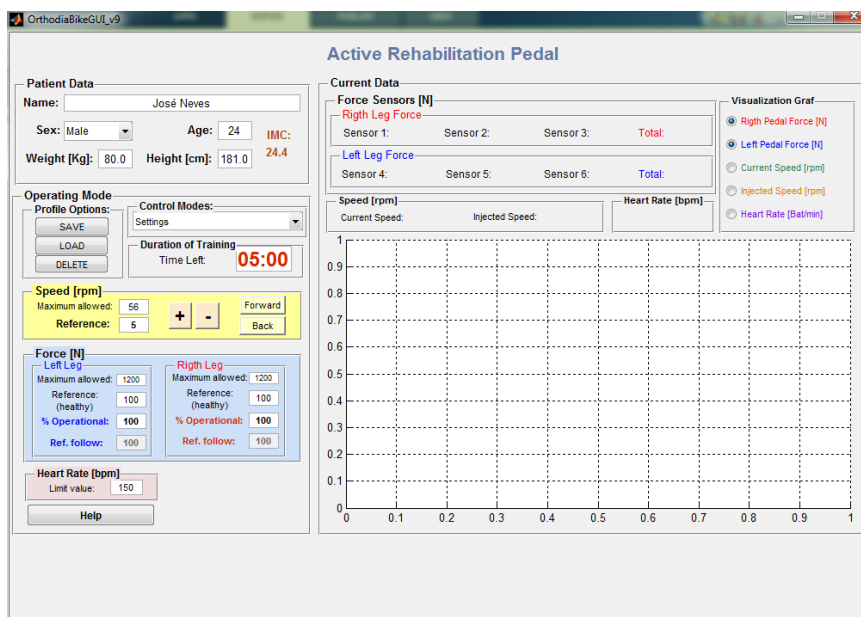


Figura 4.27 - Interface com Modo Configurações (Settings) selecionado

Este modo permite editar todos os campos que estão bloqueados nos outros modos e assim definir novos limites de força, velocidade ou batimentos cardíacos. Permite também manipular a opção “*DELETE*” disponível nas opções de perfil. O botão de “*START*” não está visível precisamente para enfatizar que é um modo de configurações e não de execução.

Capítulo 5

Testes e Resultados

5.1 Testes ao Modo de Calibração

Para comprovar a utilidade da calibração foram feitos alguns testes. O procedimento consistiu numa primeira fase em não aplicar qualquer força sobre os pedais e ver o valor da força medido, o resultado foi o seguinte (figura 5.1):

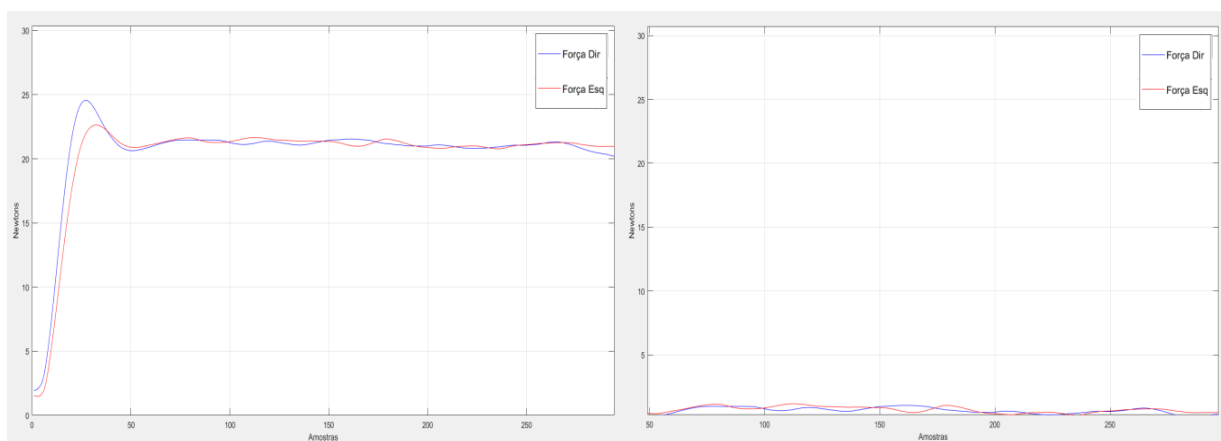


Figura 5.1 - Valor de força medido em cada pedal (à esquerda sem calibração e à direita já com calibração)

De seguida colocou-se todo o peso em cima de cada um dos pedais (a pessoa em teste pesa 80kg):

Pedal Direito (figura 5.2):

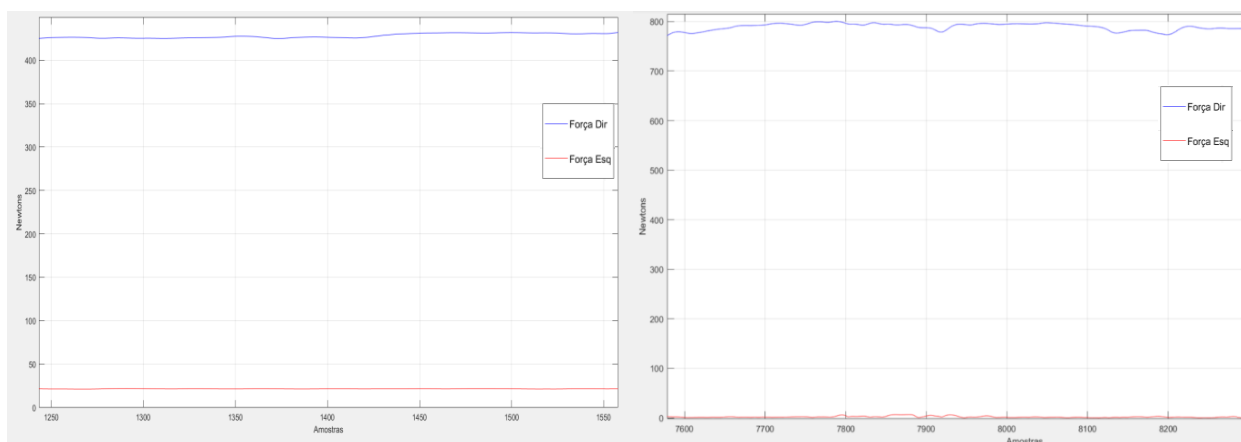


Figura 5.2 - Todo o peso aplicado sobre o pedal direito (à direita sem calibração e à esquerda após a calibração)

Pedal Esquerdo (figura 5.3):

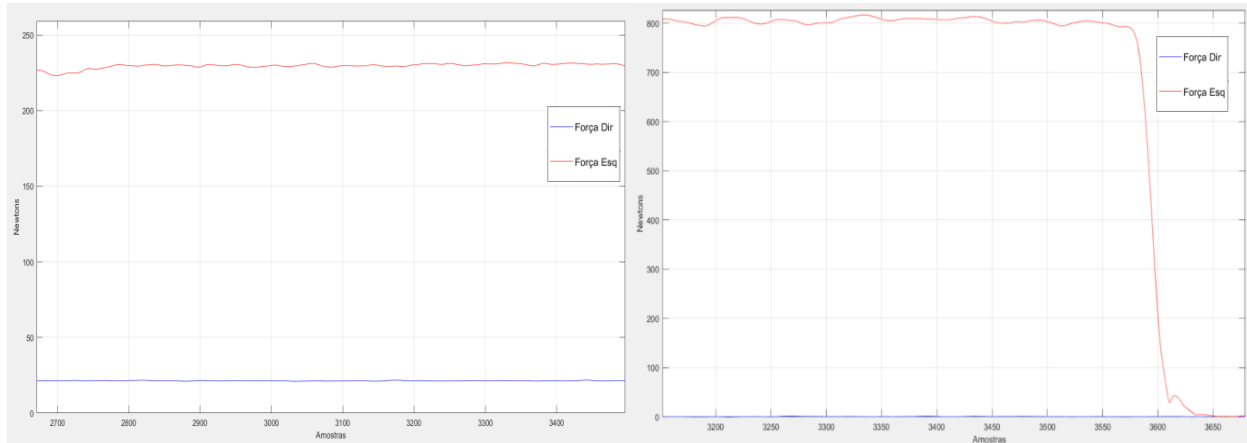


Figura 5.3 - Quando se aplica todo o peso sobre o pedal esquerdo (à direita sem calibração e à esquerda após a calibração)

Por fim colocaram-se ambos os pés nos pedais dividindo o peso entre ambos (figura 5.4):



Figura 5.4 - Peso dividido por ambos os Pedais

Era esperado que a força no pedal esquerdo fosse aproximadamente igual ao valor da força no pedal direito e que o somatório de ambos fosse aproximadamente o peso do paciente. O resultado desta simulação é ilustrado na figura seguinte:

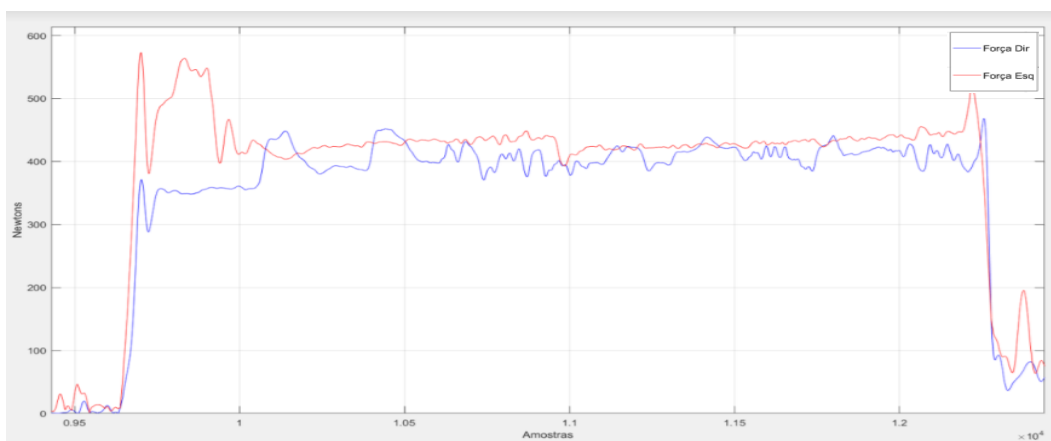


Figura 5.5 - Quando se está a dividir todo o Peso sobre os dois pedais

Como podemos comprovar por análise do gráfico da figura 5.5, tanto o somatório das forças do pedal esquerdo como do pedal direito se aproximam dos 400 N, ora este valor é precisamente metade do peso do paciente. Com isto, pode-se concluir que a calibração também funciona para valores intermédios de força e não apenas para extremos.

5.2 Teste ao Modo Bicicleta Estática Normal

Para testar este modo foi pedido a 3 pessoas (tabela 5.1) para irem variando o painel “Regulação da Força” à medida que iam pedalando. Com este teste pretendeu-se perceber se o utilizador sente o movimento natural ficando com a sensação que durante a pedalada controla o movimento dos pedais e não que é o motor a determinar a velocidade de acordo com a força feita.

Tabela 5.1 – Pessoas sujeitas ao teste no Modo Bicicleta Estática Normal

Nome	Altura (m)	Peso (kg)	Idade	Melhor Pé
Pessoa 1	1,81	80	24	Direito
Pessoa 2	1,8	75	25	Direito
Pessoa 3	1,67	52	27	Direito

Durante a simulação foram adquiridos os dados referentes à velocidade atual, força total em cada um dos pedais e o valor seleccionado no painel “Regulação de Força”. Seguem-se os resultados (figuras 5.6 a 5.8):

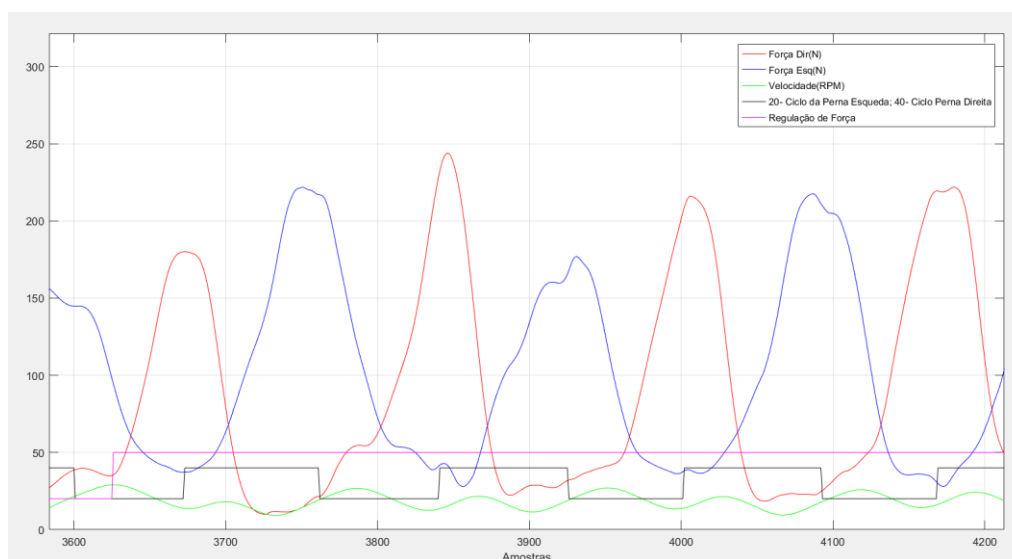


Figura 5.6 - Resultados do teste ao Modo Bicicleta Estática Normal com a Pessoa 1

No momento de simulação ilustrado no gráfico da figura 5.6, a regulação de força (magenta) está com o valor 50, logo a Pessoa 1 tem o nível 3 de Regulação de Força seleccionado.

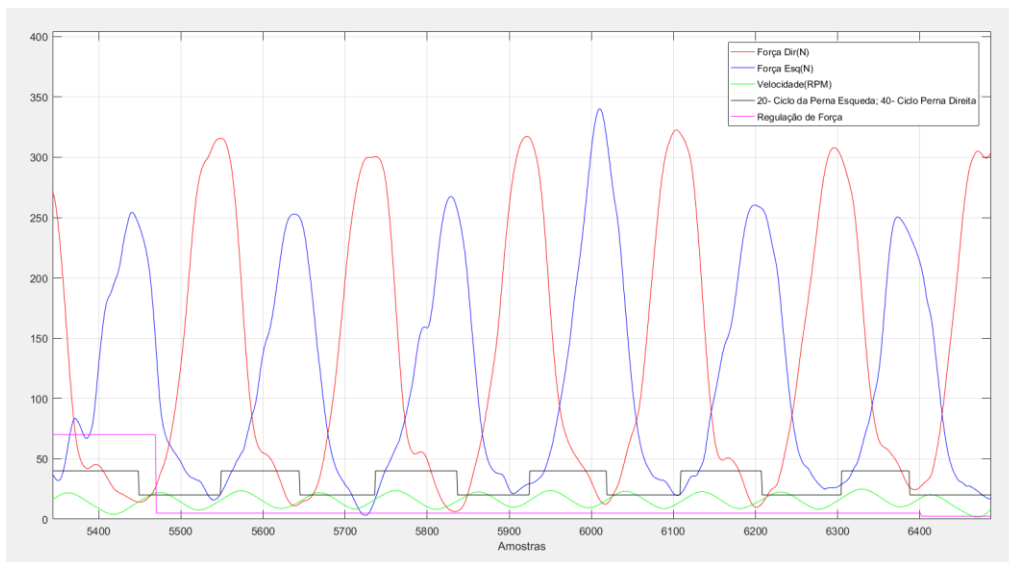


Figura 5.7 - Resultados do teste ao Modo Bicicleta Estática Normal com a Pessoa 2

No momento de simulação ilustrado no gráfico da figura 5.7, a regulação de força (magenta) está com o valor 2.5, logo a Pessoa 2 tem o nível 6 de Regulação de Força selecionado.

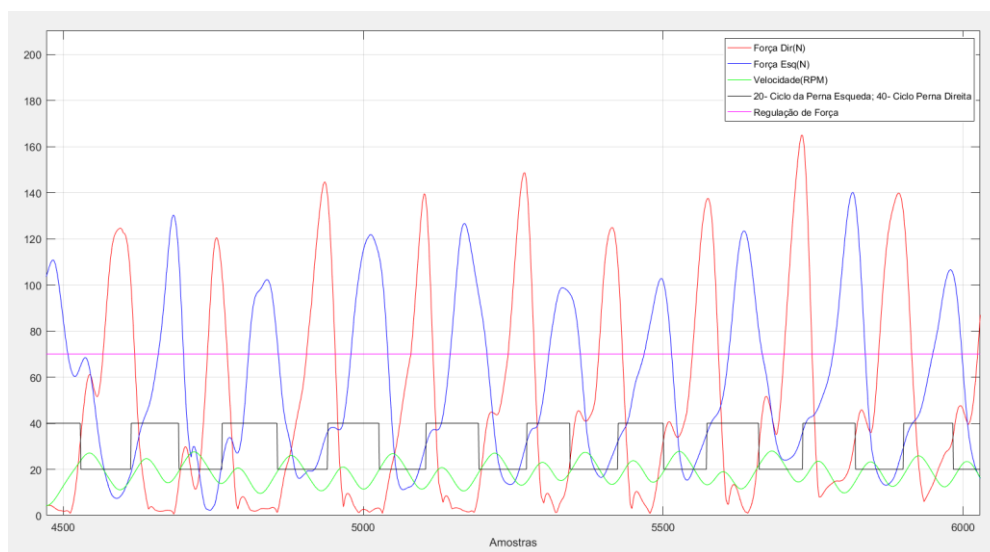


Figura 5.8 - Resultados do teste ao Modo Bicicleta Estática Normal com a Pessoa 2

No momento de simulação ilustrado no gráfico da figura 5.8, a regulação de força (magenta) está com o valor 70, logo a Pessoa 3 tem o nível 2 de Regulação de Força selecionado.

Os resultados foram de encontro ao esperado. A opinião foi unânime relativamente ao fato de durante a pedalada as pessoas em teste terem a sensação de que são as próprias a fazer andar a bicicleta e não que é o motor a determinar a velocidade. Quanto à uniformidade da pedalada, as opiniões foram também bastante satisfatórias, uma vez que tanto a aceleração como a desaceleração (quando a pessoa exerce mais ou menos força) foi feita de uma forma muito suave.

5.3 Testes ao Modo Perna Debilitada

Neste modo foram feitos testes a pacientes com diferentes percentagens de funcionalidade num dos membros inferiores. O ideal seria realizar estes testes com pessoas realmente debilitadas e com o diagnostico real de qual a percentagem de funcionalidade do membro inferior debilitado. Uma vez que não foi possível pediu-se às pessoas em teste para tentarem simular as situações descritas a seguir.

Os testes consistiram em verificar se os pacientes conseguiam realizar o movimento normal da pedalada uniformemente, mesmo com as limitações que tinham. Na tabela 5.2 pode-se consultar as características de cada um dos pacientes sujeitos aos testes.

Tabela 5.2 – Pacientes sujeitos aos testes no Modo Perna Debilitada

Nome	Altura (m)	Peso (kg)	Idade	Membro Inferior		
				Saudável	Debilitado	
					Funcionalidade (%)	
Paciente 1	1,81	80	24	Esquerdo	Direito	0
Paciente 2	1,67	52	27	Esquerdo	Direito	20
Paciente 3	1,77	72	25	Direito	Esquerdo	50
Paciente 3	1,77	72	25	Direito	Esquerdo	70

São apresentados abaixo, nas figuras 5.9 a 5.11, os resultados aos testes efetuados:

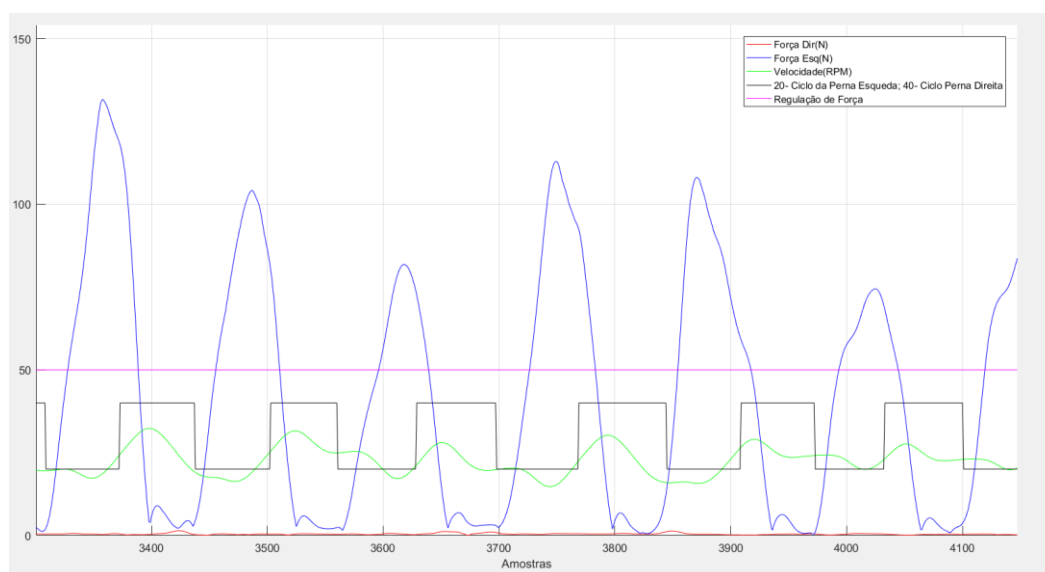


Figura 5.9 – Resultados do teste ao Paciente 1 que tem 0% da perna direita funcional

Pela análise do gráfico da figura 5.9, facilmente se conclui que a velocidade vai variando de forma consistente com o paciente a efetuar o movimento apenas com a perna saudável (esquerda), sendo a que força feita com a perna debilitada (direita) é nula.

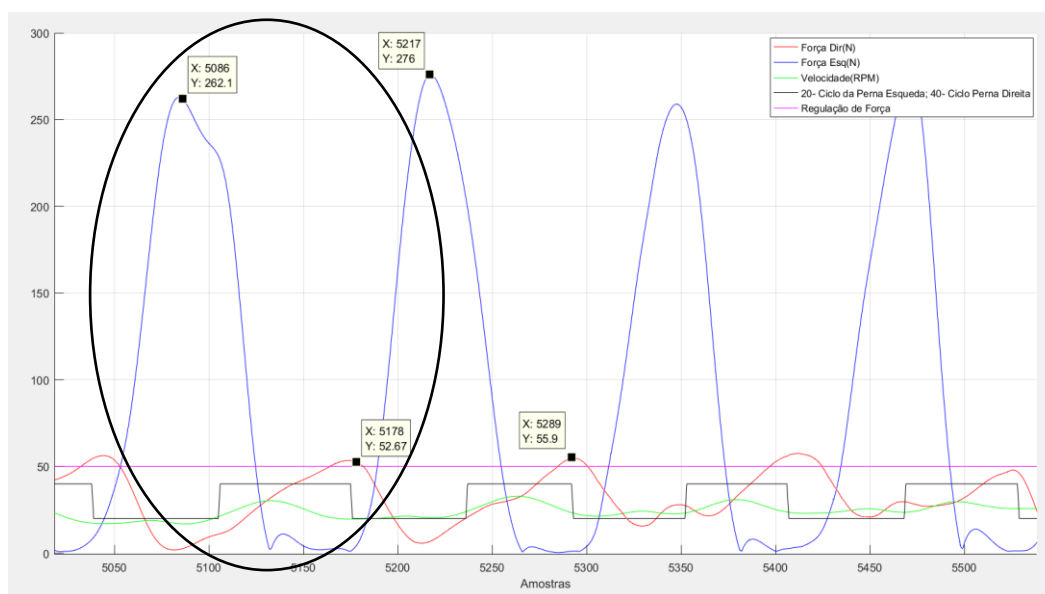


Figura 5.10 – Resultados do teste ao Paciente 2 que tem 20% da perna direita funcional

O paciente 2 (figura 5.10) tem 20% da perna direita funcional, logo o motor terá que compensar com os restantes 80% de força no meio ciclo correspondente à perna direita. Por análise do gráfico da figura 5.10, verifica-se que o paciente faz 262.1N de força no meio ciclo respetivo à perna esquerda (saudável) e 52.67N de força no meio ciclo da perna direita (debilitada). Como 20% de 262.1N são 52.42N conclui-se que se obtém o resultado esperado.

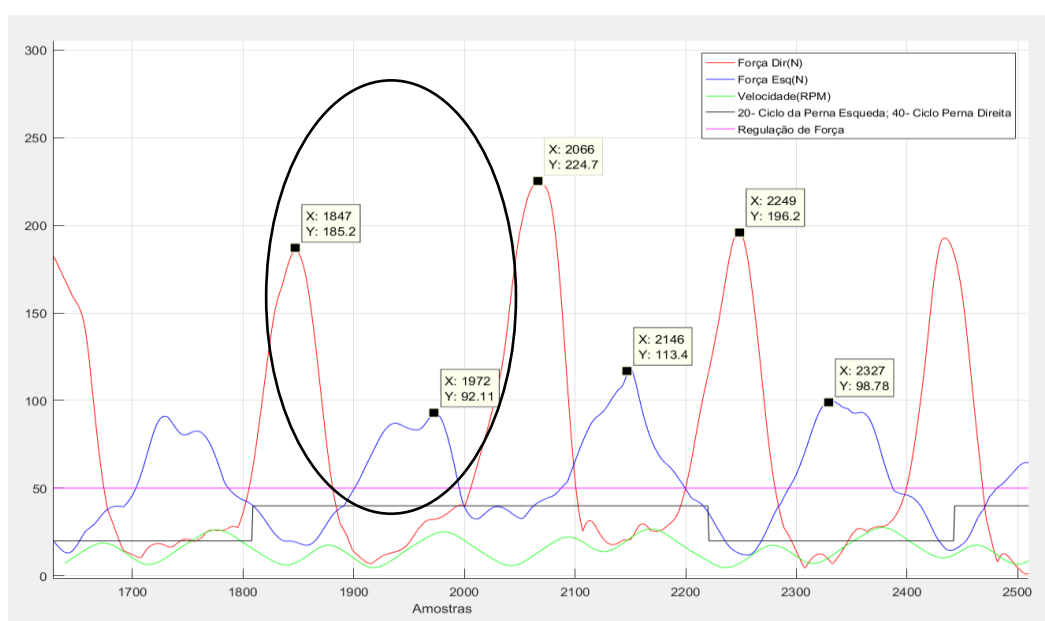


Figura 5.11 – Resultados do teste ao Paciente 3 que tem 50% da perna esquerda funcional

Analisando o gráfico da figura 5.11, vê-se que o paciente fez 185.2N de força com a perna saudável (direita) e 92.11N de força com a perna debilitada (esquerda). Portanto, apenas teve que fazer 50% de força com a perna debilitada para manter a velocidade. Posto isto, conclui-se que o teste teve um resultado esperado.

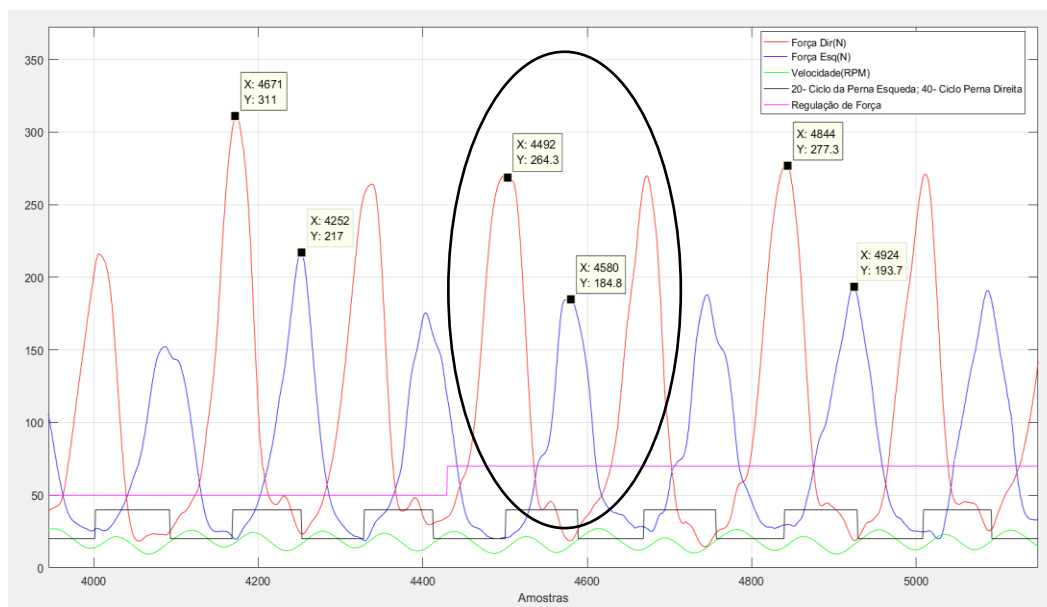


Figura 5.12 – Resultados do teste ao Paciente 3 com 70% da perna esquerda funcional

Ao analisar a Figura 5.12 vemos que o paciente fez 264.3N de força com a perna saudável (direita) e 184.8N com a perna debilitada (esquerda). Como, 70% de 264.3N são 185.01N, constata-se que o paciente teve que fazer apenas 70% de força com a perna debilitada para manter a velocidade alcançada com a perna saudável. O sistema teve, portanto, o comportamento pretendido.

Capítulo 6

Conclusões e Sugestões de Trabalhos Futuros

Neste capítulo são expostas as principais conclusões acerca do trabalho que foi desenvolvido, de acordo com os resultados obtidos aquando da realização de diversos testes experimentais. São também apresentadas diversas sugestões e melhorias a ter em conta em trabalhos futuros.

6.1 Conclusões

Neste projeto colocou-se como principal objetivo a melhoria da interface direcionada ao terapeuta e criar novos modos de funcionamento que pudessem aumentar a qualidade e utilidade deste sistema.

Foram feitas diversas alterações funcionais na interface e desenvolvidos novos modos de funcionamento dos quais se destacam: o modo Bicicleta Estática Normal, que permite ao utilizador usar este equipamento de igual forma às bicicletas estáticas convencionais, conferindo assim uma abrangência maior em termos de potenciais utilizadores; o modo Perna Imóvel, que veio permitir aos pacientes com perda total ou parcial de funcionalidade num dos membros inferiores, conseguirem utilizar uma bicicleta estática apenas com um membro; o modo Calibração que veio resolver o problema da imprecisão e saturação dos sensores de pressão, permitindo assim obter resultados com a precisão que uma reabilitação exige; o modo de Estimulação e Relaxamento Muscular que permite ao utilizador/paciente alongar ou relaxar os músculos, antes ou depois do treino.

Os testes finais vieram uma vez mais cimentar a real utilidade e o bom funcionamento de cada um dos modos desenvolvidos, permitindo que seja possível afirmar que o objetivo foi, sem dúvida, alcançado.

No futuro espera-se que após algumas melhorias de *hardware*, que possam conferir ao equipamento mais autonomia e resistência, este sistema venha a ser testado em hospitais e outros locais de reabilitação e mais tarde venha a comprovar o seu real contributo na reabilitação médica dos pacientes.

6.2 Sugestões de Trabalhos Futuros

Durante o desenvolvimento desta dissertação foram surgindo naturalmente ideias de possíveis melhorias relativamente ao que já está implementado. Pensou-se inclusive em utilizar este mesmo sistema em novas aplicações para assim poder servir outros utilizadores em diferentes contextos. Deste modo, apresentam-se as propostas seguintes não só com a finalidade de uma mera sugestão mas também com a esperança de que possam realmente vir a ser ponderadas e implementadas em trabalhos futuros.

6.2.1 Sugestões de *Hardware*

1. Neste momento a bicicleta tem ligado à roda onde estão acoplados os pedais um sensor de Hall e 2 ímanes que detetam o momento de transição em que se faz menos força num pedal e se começa a fazer mais força no outro. A limitação deste sistema é que sempre que se começa um novo ciclo global, não se sabe a posição exata dos pedais. Posto isto, fica a sugestão para se colocar mais ímanes a fim de saber com maior precisão a posição exata dos pedais, o que poderia ser de grande relevo para as análises feitas pelo terapeuta.
2. Substituir a atual *breadboard* (placa onde são implementados os circuitos) por uma placa eletrónica.
3. Para tornar todos os modos desenvolvidos ainda mais precisos e fiáveis, sugere-se a substituição dos atuais sensores de força por outros mais precisos (por exemplo células de carga utilizada em balanças) de modo a evitar que haja oscilações no seu valor a fim de obter mais precisão.
4. Criar uma ferramenta que permita gerar e armazenar energia elétrica durante a pedalada e que depois possa vir a ser utilizada, por exemplo, para carregar o *smartphone*.

6.2.2 Sugestões de *Software*

1. Esta Bicicleta foi desenvolvida com o propósito de vir a ser utilizada para fins terapêuticos e de reabilitação mas se pensarmos mais em termos de mercado não seria descabido de todo ponderar aplicar o *hardware* que foi desenvolvido ao mundo dos vídeo jogos. Esta sugestão passa por criar um *software* com um ambiente virtual que mostra o percurso de estrada consoante a distância que o ciclista (utilizador) percorre. Consoante a inclinação da

estrada o motor ia criando mais ou menos resistência para assim tornar o ambiente ainda mais real. Isto poderia ser utilizado não só por jogadores mas também por ciclistas para treinar para provas sendo que nesse caso este projeto ganharia ainda maior repercussão.

3. Sugere-se o desenvolvimento de um novo modo (que poderá partir do modo Bicicleta Estática Normal), em que estando o paciente a pedalar livremente, permita medir a diferença de força entre os membros inferiores e assim calcular a diferença de funcionalidade entre ambos. Deste modo poderia ser diagnosticada qual a percentagem exata de funcionalidade de um membro inferior debilitado.
2. Por fim, sugere-se também o desenvolvimento de um modo dirigido a pessoas que sofrem de obesidade. Tendo em conta que é um problema que afeta um número tão grande de pessoas e que tem vindo a aumentar com o estilo de vida cada vez mais sedentário. Isto poderia ser feito com a criação de um modo de funcionamento onde fosse possível criar planos de treino com o objetivo de perder calorias. Consoante o número de calorias perdidas a cada treino, a carga ia sendo ajustada com o propósito de cumprir a meta traçada para um período de tempo específico.

Referências Bibliográficas

- [1] Sociedade Portuguesa de Medicina Física e de Reabilitação, "Especialidade Médica", [Online]. Available: <http://www.spmfr.org/a-fisiatria/especialidade-medica/>. [Acedido em 2017].
- [1] Sociedade Portuguesa de Medicina Física e de Reabilitação, "Especialidade Médica", [Online]. Available: <http://www.spmfr.org/a-fisiatria/especialidade-medica/>. [Acedido em 2017].
- [2] Secção de Medicina Física e de Reabilitação da Union Européenne des Médecins Spécialistes (UEMS) e Académie Européenne de Médecine de Réadaptation, em conjunto com o European Board of Physical and Rehabilitation Medicine, " Livro Branco De Medicina Física e de Reabilitação na Europa", [Online]. Available: http://www.spmfr.org/wp-content/uploads/2016/05/livro_branco_final.pdf. [Acedido em 2017].
- [3] Ma, Rui, and Fei Hu. "An Intelligent Thermal Sensing System for Automatic, Quantitative Assessment of Motion Training in Lower-Limb Rehabilitation." IEEE Transactions on Systems, Man, and Cybernetics: Systems (2016).
- [4] Hwang, Beomsoo, and Doyoung Jeon. "A wheelchair integrated lower limb exercise/rehabilitation system: Design and experimental results on the knee joint." System Integration (SII), 2012 IEEE/SICE International Symposium on. IEEE, 2012.
- [5] Shihui Shen, Chang Gao, Yong Zhao, Haojian Lu, Yajing Shen, Chunbao Wang, Tongyang Sun, Quanquan Liu, Qing Shi, Jianjun Long, Yulong Wang, Zhengzhi Wu, Jian Qin, Weiguang Li, Massimiliano Zecca and Atsuo Takanishi, "Development of Lower Limb Rehabilitation Evaluation System Based on Virtual Reality Technology", InReal-time Computing and Robotics (RCAR), IEEE International Conference, Angkor Wat, Cambodia, June 2016.

- [6] Maxon Motor, "EC 90 flat Ø90 mm, brushless, 90 Watt, with Hall sensors", [Online]. Available: <https://www.maxonmotor.com/maxon/view/product/323772>. [Acedido em 2017].
- [7] Maxon Motor, "ESCON 50/5, 4-Q Servocontroller for DC/EC motors, 5/15 A, 10 - 50 VDC", [Online]. Available: <https://www.maxonmotor.com/maxon/view/product/control/4-Q-Servokontroller/409510>. [Acedido em 2017].
- [8] Maxon Motor, "Planetary Gearhead GP 52 C Ø52 mm, 4 - 30 Nm, Ceramic Version", [Online]. Available: <https://www.maxonmotor.com/maxon/view/product/gear/planetary/gp52/223080>. [Acedido em 2017].
- [9] Tekscan, "Force Sensors", [Online]. Available: <https://www.tekscan.com/product-group/embedded-sensing/force-sensors>. [Acedido em 2017].
- [10] Tekscan, "Tekscan Flexiforce A201 Sensor", [Online]. Available: <https://www.else.sk/media/download/datasheet/Tekscan-Flexiforce-Sensor-A201.pdf>. [Acedido em 2017].
- [11] Richard E. Klabunde, "Cardiac Cycle", [Online]. Available: <http://www.cvphysiology.com/Heart%20Disease/HD002>. [Acedido em 2017].
- [12] Amaro, João André Pereira. "Controlador e Interface Android para Equipamento de Reabilitação", MS Thesis, 2016.
- [13] National Instruments, "USB 6008 - Multifunction I/O Device", [Online]. Available: <http://www.ni.com/products/pdf/usb-datasheet.pdf>. [Acedido em 2017].
- [14] Garcia, Filipa Margarida Azevedo. "Active pedal exerciser for leg rehabilitation", MS thesis, 2015.
- [15] Fox News Point, "Top 10 Most Spoken Language In The World 2017", [Online]. <http://www.foxnewspoint.com/top-10-most-spoken-language-in-the-world-2017/A201.pdf>. [Acedido em Julho de 2017].
- [16] MathWorks, "Save", [Online]. <https://www.mathworks.com/help/matlab/ref/save.html> [Acedido em Abril de 2017].

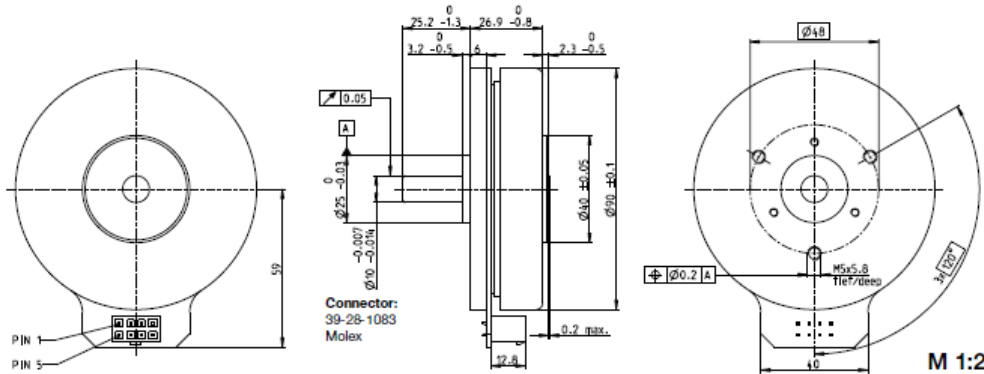
- [17] MathWorks, “msgbox”, [Online]. <https://www.mathworks.com/help/matlab/ref/msgbox.html> [Acedido em Maio de 2017].
- [18] Image Analyst, “uigetfile, load m-file variables into workspace ”, [Online]. <https://www.mathworks.com/matlabcentral/answers/317297-uigetfile-load-m-file-variables-into-workspace> [Acedido em Junho de 2017].
- [19] MathWorks, “delete”, [Online]. <https://www.mathworks.com/help/matlab/ref/delete.html> [Acedido em Abril de 2017].
- [20] MathWorks, “imread”, [Online]. <https://www.mathworks.com/help/matlab/ref/imread.html> [Acedido em Abril de 2017].
- [21] Guilherme Garcia Holderbaum, Antônio Carlos Stringhini Guimarães, Ricardo Demétrio de Souza Petersen, “Methodological proposal for teaching the pedaling technique of cycling using a system of visual augmented feedback: a previous study”, [Online]. <http://www.efdeportes.com/efd173/proposal-for-teaching-the-pedaling-technique.htm> [Acedido em Abril de 2017].
- [22] MathWorks, “set_param”, [Online]. https://www.mathworks.com/help/simulink/slref/set_param.html?searchHighlight=set_param&s_tid=doc_srchtile [Acedido em Abril de 2017].
- [23] MathWorks, “Sample and Hold”, [Online]. <https://www.mathworks.com/help/dsp/ref/sampleandhold.html> [Acedido em Abril de 2017].
- [24] Maxon Motor Control, "EC 90 flat Ø90 mm, brushless, 90 Watt - Datasheet", [Online]. Available: https://www.maxonmotor.com/medias/sys_master/root/8825435389982/17-EN-271.pdf. [Acedido em 2017].
- [25] Maxon Motor Control, "Escon Overview ", [Online]. Available: <https://www.maxonmotor.com/maxon/view/product/323772>. [Acedido em 2017].

Anexo A

Datasheets

Motor EC 90 flat [24]

EC 90 flat $\varnothing 90$ mm, brushless, 90 Watt



maxon flat motor

- Stock program
- Standard program
- Special program (on request)

Part Numbers

with Hall sensors

323772 **429271** **244879**

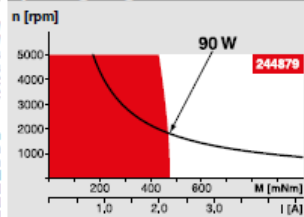
Motor Data

Values at nominal voltage		V	24	36	48
1	Nominal voltage	V	24	36	48
2	No load speed	rpm	3190	3120	2080
3	No load current	mA	544	348	135
4	Nominal speed	rpm	2590	2510	1610
5	Nominal torque (max. continuous torque)	mNm	444	560	533
6	Nominal current (max. continuous current)	A	6.06	4.76	2.27
7	Stall torque	mNm	4940	7480	4570
8	Stall current	A	70	69	21.1
9	Max. efficiency	%	84	87	85
Characteristics					
10	Terminal resistance phase to phase	Ω	0.343	0.522	2.28
11	Terminal inductance phase to phase	mH	0.264	0.625	2.5
12	Torque constant	mNm/A	70.5	109	217
13	Speed constant	rpm/V	135	88	44
14	Speed/torque gradient	rpm/mNm	0.659	0.423	0.462
15	Mechanical time constant	ms	21.1	13.6	14.8
16	Rotor inertia	gcm ²	3060	3060	3060

Specifications

Thermal data		
17	Thermal resistance housing-ambient	1.91 K/W
18	Thermal resistance winding-housing	2.6 K/W
19	Thermal time constant winding	46 s
20	Thermal time constant motor	283 s
21	Ambient temperature	-40...+100°C
22	Max. winding temperature	+125°C
Mechanical data (preloaded ball bearings)		
23	Max. speed	5000 rpm
24	Axial play at axial load < 15 N	0 mm
	> 15 N	0.14 mm
25	Radial play	preloaded
26	Max. axial load (dynamic)	12 N
27	Max. force for press fits (static) (static, shaft supported)	183 N
28	Max. radial load, 5 mm from flange	8000 N
		68 N

Operating Range



Comments

- Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.
- Short term operation**
The motor may be briefly overloaded (recurring).
- Assigned power rating**

Other specifications

- 29 Number of pole pairs 12
- 30 Number of phases 3
- 31 Weight of motor 600 g

Values listed in the table are nominal.

Connection

- Pin 1 Hall sensor 1
- Pin 2 Hall sensor 2
- Pin 3 V_{sup} 4.5...18 VDC
- Pin 4 Motor winding 3
- Pin 5 Hall sensor 3
- Pin 6 GND
- Pin 7 Motor winding 1
- Pin 8 Motor winding 2

Wiring diagram for Hall sensors see p. 43

Cable

- Connection cable Universal, L = 500 mm **339380**
- Connection cable to EPOS2, L = 500 mm **354045**

maxon Modular System

Planetary Gearhead
 $\varnothing 62$ mm
 4 - 30 Nm
 Page 351



Overview on page 28-36

Encoder MILE
 512 - 6400 CPT,
 2 channels
 Page 390

Recommended Electronics:

- Notes Page 32
- ESCON Mod. 50/4 EC-S 427
- ESCON Mod. 50/5 427
- ESCON 50/5 428
- ESCON 70/10 428
- DEC Module 50/5 430
- EPOS2 24/5, 50/5, 70/10 435
- EPOS2 P 24/5 438
- EPOS4 Module/CB 50/5 442
- EPOS4 Module 50/8 443
- EPOS4 Comp. 50/8 CAN 443
- MAXPOS 50/5 447

Controlador Maxon ESCON [25]

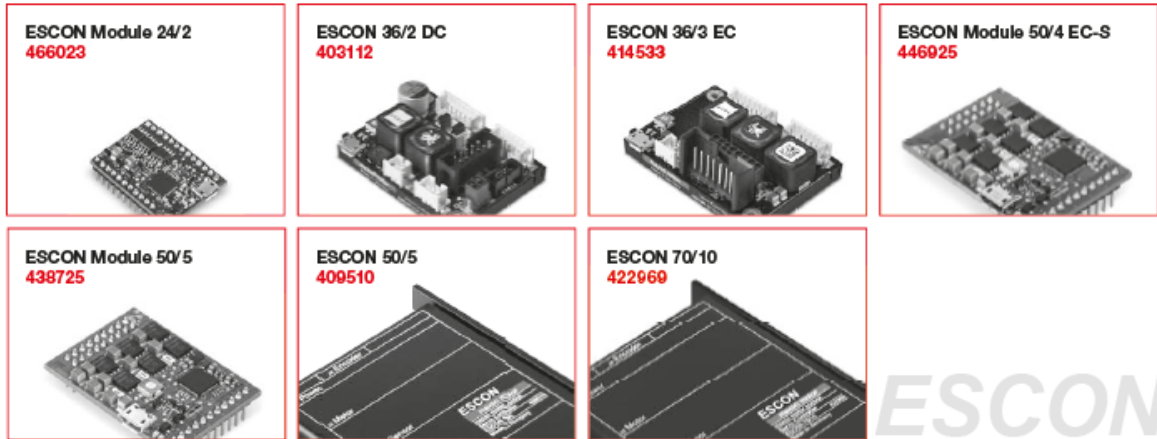
maxon motor control

ESCON Overview

The ESCON servo controllers are small-sized, powerful 4-quadrant PWM servo controller for the highly efficient control of permanent magnet-activated DC motors.

The featured operating modes – speed control (closed loop), speed control (open loop), and current control – meet the highest requirements. The ESCON servo controllers are designed being commanded by an analog set value and

features extensive analog and digital I/O functionality and are being configured via USB interface using the graphical user interface "ESCON Studio" for Windows PCs.



Depending on the ESCON variant, the following motor types can be operated

- **DC motor:** Permanent-magnet DC motor
- **EC motor:** Brushless, electronically commutated permanent-magnet DC motor (BLDC) with and without Hall sensors.

Various operating modes allow an adaptable use in a wide range of drive systems

- **Current controller:** The current controller compares the actual motor current (torque) with the applied set value. In case of deviation, the motor current is dynamically readjusted.
- **Speed controller (closed loop):** The closed loop speed controller compares the actual speed signal with the applied set value. In case of deviation, the speed is dynamically readjusted.
- **Speed controller (open loop):** The open loop speed controller feeds the motor with a voltage proportional to the applied speed set value. Changes in load are compensated using the IxR methodology.

Speed measurement by

- **Digital incremental encoder:** The encoders deliver simple square signals for further

processing. Their impulses are counted to determine the speed. Channels A and B are phase-shifted signals, which are being compared to determine the direction of rotation.

- **DC tachometer:** The DC tachometer delivers a speed-proportional analog voltage.
- **Available Hall sensors:** The Hall sensors deliver six different combinations of switching impulses per electrical turn which are counted to determine speed. They also deliver phase-shifted signals that are being compared to determine the direction of rotation.
- **Sensorless EC:** The speed is determined by the progression of the induced voltage. The electronics evaluates the zero crossing of the induced voltage (EMF).

To the numerous inputs and outputs, various functionalities can be assigned to.

Set value (speed or current), current limitation, as well as offset can be assigned as follows.

- **Analog value:** The value is defined by an analog voltage set via external or internal potentiometer.
- **PWM value:** The value is defined by fixed frequency and amplitude. The desired change is achieved by variation of the duty cycle of 10...90%.

- **RC Servo Value:** The value is set with a signal pulse with a duration of 1.0...2.0 ms.

- **Fixed value:** The value is defined by a fixed preset value.
- **2 fixed values:** Value 1 is defined by a fixed preset value 1. Value 2 is defined by a fixed preset value 2. A digital input is used to switch between the two preset values.

Various functionalities are available to enable the power stage.

- **Enable:** Enables or disables the power stage.
- **Enable & Direction:** Enables or disables the power stage and determines the motor shaft's direction of rotation.
- **Enable CW:** Enables or disables the power stage in direction of rotation-dependent sense. The rotor can only turn clockwise (CW).
- **Enable CCW:** Enables or disables the power stage in direction of rotation-dependent sense. The rotor can only turn counterclockwise (CCW).
- **Enable CW & CCW:** Enables or disables the power stage in direction of rotation-dependent sense. The signals are interlocked against each other.

The ramp function permits controlled acceleration/deceleration of the motor shaft in both, open loop and closed loop speed controller mode.

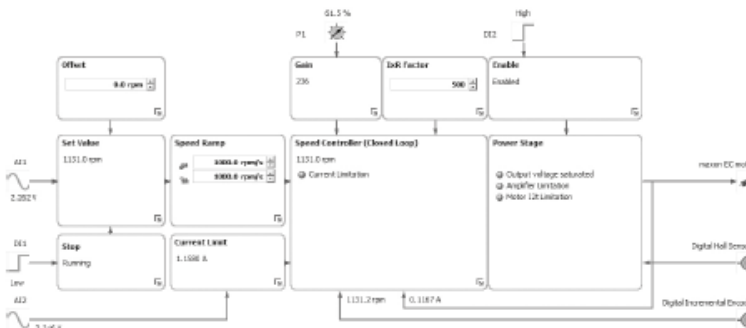
- **Analog ramp:** The ramp is defined by a variable analog value.
- **Fixed ramp:** The ramp is defined by a fixed preset value.

Stop: The motor shaft decelerates with preset speed ramp until complete standstill.

Ready: The Ready signal can be used to transmit the operational status (respectively fault) to a superior control.

Speed and Current Comparator: The digital output is set depending on the actual value.

- **Limit:** The digital output is set as soon as the preset value is reached. It remains set as long as the value is exceeded.



ESCON Studio (Controller Monitor)

- **Range:** The digital output is set as soon as the preset value range is reached. It remains set as long as the value remains in range.
- **Deviation:** The digital output is set as soon as the preset value deviation (based on the set value) is in range.

With the integrated **potentiometers** the additional following functions can be adjusted

- **Current Gain:** Adjustment of the current controller gain.
- **Speed Gain:** Adjustment of the speed controller gain.
- **IR Factor:** The voltage drop caused by terminal resistance will be compensated in the range of [0...1000...2000].

Analog outputs allow monitoring of

- **Actual current:** Actually measured motor winding current.
- **Actual current averaged:** Actually measured motor winding current filtered by first order digital low-pass filter with a cut-off frequency of 5 Hz.
- **Actual speed:** Actually measured motor speed.
- **Actual speed averaged:** Actually measured motor speed filtered by 1st order digital low-pass filter with a cut-off frequency of 5 Hz.
- **Demand Current:** Demanded motor winding current.
- **Demand Speed:** Demanded motor speed.
- **Temperature Power Stage:** Actually measured power stage temperature.
- **Fixed value:** The output voltage is said fixed to the preset value.

Easy startup

Startup and parameterization are performed using the intuitive graphical user interface "ESCON Studio" with the help of simple to use, menu-guided wizards. The following wizards are available: Startup, Regulation Tuning, Firmware Update, Controller Monitor, Parameters, Data Recording, and Diagnostics.

Protective equipment

The servo controller has protective circuits against overcurrent, excess temperature, under- and overvoltage, against voltage transients, and against short-circuits in the motor cable. Furthermore it is equipped with protected digital inputs and outputs and an adjustable current limitation for protecting the motor and the load. The motor current and the actual speed of the motor shaft can be monitored by means of the analog output voltage.

Comprehensive documentation

Using the "Feature Comparison Chart", the suitable ESCON servo controller can easily be determined. The "Hardware Reference" comprises the specifications of the hardware in detail. The documents "Firmware Version" and "Release Notes" describe changes and improvements of firmware and software. In addition, the graphical user interface "ESCON Studio" features a comprehensive online help.



Software

Installation Program: ESCON Setup

Graphical User Interface: ESCON Studio

- ✓ Startup Wizard
- ✓ Regulation Tuning
- ✓ Diagnostic
- ✓ Firmware Update
- ✓ Controller Monitor
- ✓ Parameters
- ✓ Data Recording
- ✓ Online Help

Language: German, English, French, Italian, Spanish, Japanese, Chinese

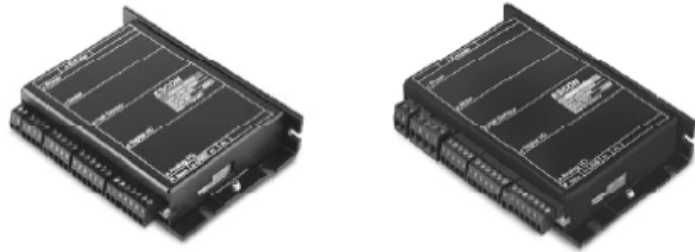
Operating System: Windows 10, Windows 8, Windows 7, Windows XP SP3

Communication interface: USB 2.0/3.0 (full speed)

Accessories ESCON*	M 24/2	36/2 DC	36/3 EC	M 50/4 EC-S	M 50/5	50/5	70/10
404404 ESCON 36/2 DC Connector Set	—	✓	—	—	—	—	—
425255 ESCON 36/3 EC Connector Set	—	—	✓	—	—	—	—
403962 DC Motor Cable	—	✓	—	—	—	—	—
403964 I/O Cable 7 core (analog I/O's)	—	✓	✓	—	—	—	—
403965 I/O Cable 6 core (digital I/O's)	—	✓	✓	—	—	—	—
275934 Encoder Cable	—	✓	—	—	—	✓	✓
403957 Power Cable	—	✓	✓	—	—	—	—
403968 USB Type A - micro B Cable	✓	✓	✓	✓	✓	✓	✓
418719 Adapter BLACK FPC11poles	—	—	✓	—	—	—	—
418723 Adapter BLUE FPC8poles	—	—	✓	—	—	—	—
418721 Adapter GREEN FPC8poles	—	—	✓	—	—	—	—
486400 ESCON Module 24/2 Motherboard	✓	—	—	—	—	—	—
438779 ESCON Module Motherboard	—	—	—	—	✓	—	—
450237 ESCON Module Motherboard Sensorless	—	—	—	✓	—	—	—
409286 ESCON USB Stick	✓	✓	✓	✓	✓	✓	✓

*not included in delivery

ESCON Feature Comparison Chart



	ESCON 50/5	ESCON 70/10
DC motors up to (continuous / maximum)	250 W / 750 W	700 W / 2100 W
EC motors up to (continuous / maximum)	250 W / 750 W	700 W / 2100 W
Sensors		
	Digital Incremental Encoder (2 channel with or without Line Driver)	Digital Incremental Encoder (2 channel with or without Line Driver)
	DC Tacho	DC Tacho
	Without sensor (DC motors)	Without sensor (DC motors)
	Digital Hall Sensors (EC motors)	Digital Hall Sensors (EC motors)
Operating mode		
	Current controller (torque control), Speed controller (closed and open loop)	Current controller (torque control), Speed controller (closed and open loop)
Electrical data		
Nominal operating voltage V_{CC}	10 - 50 VDC	10 - 70 VDC
Max. output voltage	$0.99 \times V_{CC}$	$0.95 \times V_{CC}$
Max. output current	15 A (<20 s)	30 A (<20 s)
Continuous output current	5 A	10 A
Pulse width modulation frequency	53.6 kHz	53.6 kHz
Sampling rate PI current controller	53.6 kHz	53.6 kHz
Sampling rate PI speed controller	5.36 kHz	5.36 kHz
Max. efficiency	95%	98%
Max. speed (DC)	limited by max. speed (motor) and max. output voltage (controller)	limited by max. speed (motor) and max. output voltage (controller)
Max. speed (EC; 1 pole pair)	150 000 rpm	150 000 rpm
Built-in motor choke	3 x 30 μ H / 5 A	3 x 15 μ H / 10 A
Inputs/Outputs		
Hall sensor signals	H1, H2, H3	H1, H2, H3
Encoder signals	A, A \backslash , B, B \backslash	A, A \backslash , B, B \backslash
Max. encoder input frequency differential (single-ended)	1 MHz (100 kHz)	1 MHz (100 kHz)
Potentiometers	2	2
Digital inputs	2	2
Digital inputs/outputs	2	2
Analog inputs	2	2
Resolution, Range, Circuit	12-bit, -10...+10 V, differential	12-bit, -10...+10 V, differential
Analog outputs	2	2
Resolution, Range	12-bit, -4...+4 V	12-bit, -4...+4 V
Auxiliary voltage output	+5 VDC (IL \leq 10 mA)	+5 VDC (IL \leq 10 mA)
Hall sensor supply voltage	+5 VDC (IL \leq 30 mA)	+5 VDC (IL \leq 30 mA)
Encoder supply voltage	+5 VDC (IL \leq 70 mA)	+5 VDC (IL \leq 70 mA)
Status Indicators	Operation: green LED / Error: red LED	Operation: green LED / Error: red LED
Environmental conditions		
Temperature - Operation	-30...+45°C	-30...+45°C
Temperature - Extended range	+45...+85°C; Derating: -0.111 A/°C	+45...+82°C; Derating: -0.270 A/°C
Temperature - Storage	-40...+85°C	-40...+85°C
Humidity (condensation not permitted)	5...90%	5...90%
Mechanical data		
Weight	Approx. 204 g	Approx. 259 g
Dimensions (L x W x H)	115 x 75.5 x 24 mm	125 x 78.5 x 27 mm
Mounting holes	for screws M4	for screws M4
Part numbers		
	409510 ESCON 50/5	422969 ESCON 70/10
	Order accessories separately, from page 437	Order accessories separately, from page 437



FlexiForce[®]

Standard Model A201



The FlexiForce A201 is our standard sensor and meets the requirements of most customers. The A201 is a thin and flexible piezoresistive force sensor that is available off-the-shelf in a variety of lengths for easy proof of concept. These ultra-thin sensors are ideal for non-intrusive force and pressure measurement in a variety of applications. This sensor is designed to use with your own electronics or multimeter.

BENEFITS

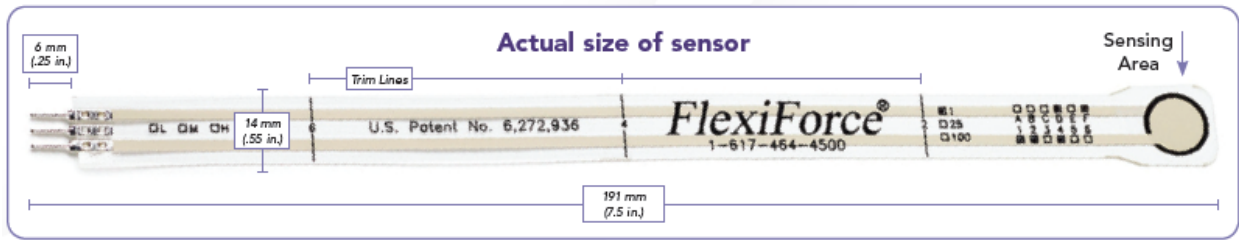
- Thin and flexible
- Easy to use
- Convenient and affordable

PHYSICAL PROPERTIES

Thickness	0.203 mm (0.008 in.)
Length	191 mm (7.5 in.)* (optional trimmed lengths: 152 mm (6 in.), 102 mm (4 in.), 51 mm (2 in.))
Width	14 mm (0.55 in.)
Sensing Area	9.53 mm (0.375 in.) diameter
Connector	3-pin Male Square Pin (center pin is inactive)
Substrate	Polyester (ex: Mylar)
Pin Spacing	2.54 mm (0.1 in.)

✓ ROHS COMPLIANT

* Length does not include pins, please add approximately 6mm (0.25 in.) for pin length for a total length of approximately 197 mm (7.75 in).

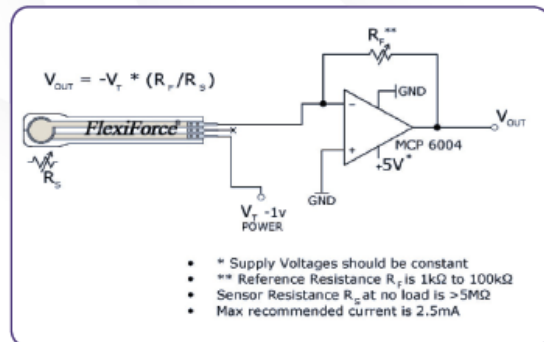


STANDARD FORCE RANGES (as tested with circuit shown below)

- 4.4 N (0 - 1 lb)
- 111 N (0 - 25 lb)
- 445 N (0 - 100 lb)

In order to measure forces above 100 lb (up to 1000 lb), apply a lower drive voltage (-0.5 V, -0.10 V, etc.) and reduce the resistance of the feedback resistor (1kΩ min.) Conversely, the sensitivity can be increased for measurement of lower forces by increasing the drive voltage or resistance of the feedback resistor.

Recommended Circuit



	Typical Performance	Evaluation Conditions
Linearity (Error)	< ±3%	Line drawn from 0 to 50% load
Repeatability	< ±2.5% of full scale	Conditioned sensor, 80% of full force applied
Hysteresis	< 4.5 % of full scale	Conditioned sensor, 80% of full force applied
Drift	< 5% per logarithmic time scale	Constant load of 111 N (25 lb)
Response Time	< 5μsec	Impact load, output recorded on oscilloscope
Operating Temperature	-40°C - 60°C (-40°F - 140°F)	Time required for the sensor to respond to an input force

- Force reading change per degree of temperature change = 0.36%/°C (±0.2%/°F)



ISO 9001 & 13485



PURCHASE TODAY ONLINE AT
WWW.TEKSCAN.COM/STORE

+1.617.464.4283

1.800.248.3669

info@tekscan.com

www.tekscan.com/flexiforce

Anexo B

Código Matlab Desenvolvido

Opções do Perfil do Paciente

Opção SAVE:

```
% --- Executes on button press in GravarPerfil.
function GravarPerfil_Callback(hObject, eventdata, handles)
[filename, pathname] = uiputfile('*.mat','Save Workspace As: '); %janela browse
newfilename = fullfile(pathname, filename); %guardar a localização e o nome dados pelo user
command = sprintf('save('%s')', newfilename); %comando para guardar na localização e com o nome definidos
evalin('base', command) %guarda no local e c/nome pretendido todas as variaveis do workspace atual

h = msgbox('Workspace saved!','Success');
```

Opção LOAD:

```
function CarregarPerfil_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.mat','Select mat file: ');
newfilename = fullfile(pathname, filename);
command = sprintf('load('%s')', newfilename); %comando para carregar na localização e com o nome definidos
evalin('base', command);

h = msgbox('File loaded!','Success');
```

Opção DELETE:

```
function EliminarPerfil_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.mat','Select the item you want to delete: ');
newfilename = fullfile(pathname, filename);

choice = questdlg('Are you sure you want to delete this file?', ...
    'Confirmation', ...
    'Yes','No','No');
% Handle response
switch choice
case 'Yes'
    command = sprintf('delete('%s')', newfilename); %comando para eliminar na localização e com o nome definidos
    msgbox('File deleted!','Success');
    evalin('base', command);
case 'No'
    msgbox('Dont worry, file not deleted!');
end
```

Instruções ao Escolher Modo de Funcionamento

Modo Bicicleta Estática Normal:

```

%%Mensagem e Imagem de posição inicial correta%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
img=imread('BEC.jpg');
I2=imresize(img, [449 484]); %[449 484] tamanho original, posso colocar outros
h=uicontrol('style','pushbutton',...
            'units','pixels',...
            'position',[400 10 485 450],... %[posição x,posição y,comprimento,altura]
            'cdata',I2);

msg = msgbox('Before pressing START put the pedals in position as shows in figure.','Attention');
set(msg, 'position', [368 570 280 60]); %makes box bigger, [posição x,posição y,comprimento,altura]
uiwait(msg);
set(h,'visible', 'off')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Modo Perna Debitada:

```

%%Mensagem e Imagem de posição inicial correta%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
img=imread('NoSensitivityInOneLeg223.jpg');
I2=imresize(img, [449 484]); %[449 484] tamanho original, posso colocar outros
h=uicontrol('style','pushbutton',...
            'units','pixels',...
            'position',[400 10 485 450],... %[posição x,posição y,comprimento,altura]
            'cdata',I2);

msg = msgbox('Before pressing START put the pedals in position as shows in figure (pedal on top that corresponds to the healthy leg)','Attention');
set(msg, 'position', [368 570 280 60]); %makes box bigger, [posição x,posição y,comprimento,altura]
uiwait(msg);
set(h,'visible', 'off')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Modo Velocidade Constante

Botão “+”:

```

% --- Executes on button press in more_speed.
function more_speed_Callback(hObject, eventdata, handles)

    incremento=2;
    handles.vel_ref=handles.vel_ref+incremento; %pegando no valor da vel ref definido inicialmente incrementa-se ao carregar no botao

    assignin('base', 'speed_ref', handles.vel_ref*handles.rel_MPedaleira);
    set(handles.vel_ref_edit,'String',num2str(handles.vel_ref,'%2.0u')) %% coloca a Vel ref com o valor pretendido

    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

if handles.vel_ref>=handles.vel_max
%Termina simulacao
    set(handles.tempo_edit,'Enable','off')
    set(handles.start,'String','STOP')
    set(handles.start,'BackgroundColor',[1.0,0.4,0.4])
    set_param(handles.ModelName, 'SimulationCommand', 'start');

    set_param([handles.ModelName, '/Enable'], 'sw', '0'); %Faz o Disable do ESCON
    set_param([handles.ModelName, '/Stop'], 'sw', '1'); %Faz o Stop do Motor
    stop(handles.update_simdata_timer); %parar a contagem do tempo

    h=errordlg('Maximum speed is 55 rot/min!');%mensagem de erro
    uiwait(h);
    return;
end
handles.output = hObject;
guidata(hObject, handles);

```

Botão “-“:

```
% --- Executes on button press in less_speed.
function less_speed_Callback(hObject, eventdata, handles)
% hObject    handle to less_speed (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes on button press in less_speed.

decremento=2; %valor de decremento
handles.vel_ref=handles.vel_ref-decremento; %pegando no valor da vel ref definido inicialmente decrementa ao carregar
assignin('base', 'speed_ref', handles.vel_ref*handles.rel_MPedaleira);
set(handles.vel_ref_edit,'String',num2str(handles.vel_ref,'%2.0u')) %coloca a Vel ref com o valor pretendido

set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

if handles.vel_ref<=4 %como se vai decrementar 5, quando estamos com 9 se tirarmos 5 atingimos o limite
%Termina simulacao
set(handles.tempo_edit,'Enable','off')
set(handles.start,'String','STOP')
set(handles.start,'BackgroundColor',[1.0,0.4,0.4])
set_param(handles.ModelName, 'SimulationCommand', 'start');

set_param([handles.ModelName, '/Enable'], 'sw', '0'); %Faz o Disable do ESCON
set_param([handles.ModelName, '/Stop'], 'sw', '1'); %Faz o Stop do Motor
stop(handles.update_simdata_timer); %parar a contagem do tempo

h=errordlg('Minimum speed is 4 rot/min!'); %mensagem de erro
uiwait(h);
return;
end
handles.output = hObject;
guidata(hObject, handles);
```

Botão “frente”:

```
% --- Executes on button press in rot_frente.
function rot_frente_Callback(hObject, eventdata, handles)
% hObject    handle to rot_frente (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param([handles.ModelName, '/Direction'], 'sw', '0'); %Roda para a frente
```

Botão “trás”:

```
% --- Executes on button press in rot_tras.
function rot_tras_Callback(hObject, eventdata, handles)
% hObject    handle to rot_tras (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set_param([handles.ModelName, '/Direction'], 'sw', '1'); %Roda para trás
```

Painel de Regulação de Força

Botões “1” até “7” respetivamente:

```
% --- Executes on button press in RegForcel.
function RegForcel_Callback(hObject, eventdata, handles)
% hObject    handle to RegForcel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.RegForce=90;
assignin('base', 'RegForca', handles.RegForce); %RegForca-->variavel simulink da regulacao da forca
set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da simulink
```

```

% --- Executes on button press in RegForce2.
function RegForce2_Callback(hObject, eventdata, handles)
% hObject     handle to RegForce2 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.RegForce=70;
assignin('base', 'RegForca', handles.RegForce); %RegForca-->variavel simulink da regulacao da forca
set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da simulink

% --- Executes on button press in RegForce3.
function RegForce3_Callback(hObject, eventdata, handles)
% hObject     handle to RegForce3 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.RegForce=50;
assignin('base', 'RegForca', handles.RegForce); %RegForca-->variavel simulink da regulacao da forca
set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da simulink

% --- Executes on button press in RegForce4.
function RegForce4_Callback(hObject, eventdata, handles)
% hObject     handle to RegForce4 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.RegForce=20;
assignin('base', 'RegForca', handles.RegForce); %RegForca-->variavel simulink da regulacao da forca
set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da simulink

% --- Executes on button press in RegForce5.
function RegForce5_Callback(hObject, eventdata, handles)
% hObject     handle to RegForce5 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.RegForce=5;
assignin('base', 'RegForca', handles.RegForce); %RegForca-->variavel simulink da regulacao da forca
set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da simulink

% --- Executes on button press in RegForce6.
function RegForce6_Callback(hObject, eventdata, handles)
% hObject     handle to RegForce6 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.RegForce=2.5;
assignin('base', 'RegForca', handles.RegForce); %RegForca-->variavel simulink da regulacao da forca
set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da simulink

% --- Executes on button press in RegForce7.
function RegForce7_Callback(hObject, eventdata, handles)
% hObject     handle to RegForce7 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
handles.RegForce=1;
assignin('base', 'RegForca', handles.RegForce); %RegForca-->variavel simulink da regulacao da forca
set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da simulink

```

Modo Perna Debilitada – Painel “Sick Leg Functionality (0-100%)”

Botão “-”:

```

% --- Executes on button press in Less_Functional.
function Less_Functional_Callback(hObject, eventdata, handles)
% hObject    handle to Less_Functional (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    decremento=5;
    handles.Functional=handles.Functional-decremento; %pegando no valor da % Functional definido inicialmente incrementa 5

    assignin('base', 'Functional', handles.Functional/100); %/100 para ser com valores entre 0 e 1
    set(handles.Functional_edit,'String',num2str(handles.Functional,'%2.0u')) %% coloca a % Functional com o valor pretendido

    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

if handles.Functional<0
    handles.Functional=0;
    assignin('base', 'Functional', handles.Functional/100); %/100 para ser com valores entre 0 e 1
    set(handles.Functional_edit,'String',num2str(handles.Functional,'%2.0u')) %% coloca a % Functional com o valor pretendido
    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

    h=errordlg('Minimum is 0%!');%mensagem de erro
    uiwait(h);
end
handles.output = hObject;
guidata(hObject, handles);

```

Botão “-”:

```

% --- Executes on button press in LessLess_Functional.
function LessLess_Functional_Callback(hObject, eventdata, handles)
% hObject    handle to LessLess_Functional (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    decremento=10;
    handles.Functional=handles.Functional-decremento; %pegando no valor da % Functional definido inicialmente incrementa 5

    assignin('base', 'Functional', handles.Functional/100); %/100 para ser com valores entre 0 e 1
    set(handles.Functional_edit,'String',num2str(handles.Functional,'%2.0u')) %% coloca a % Functional com o valor pretendido

    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

if handles.Functional<0
    handles.Functional=0;
    assignin('base', 'Functional', handles.Functional/100); %/100 para ser com valores entre 0 e 1
    set(handles.Functional_edit,'String',num2str(handles.Functional,'%2.0u')) %% coloca a % Functional com o valor pretendido
    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

    h=errordlg('Minimum is 0%!');%mensagem de erro
    uiwait(h);
end
handles.output = hObject;
guidata(hObject, handles);

```

Botão “+”:

```

% --- Executes on button press in More_functional.
function More_functional_Callback(hObject, eventdata, handles)
% hObject    handle to More_functional (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    incremento=5;
    handles.Functional=handles.Functional+incremento; %pegando no valor da % Functional definido inicialmente incrementa 5

    assignin('base', 'Functional', handles.Functional/100); %/100 para ser com valores entre 0 e 1
    set(handles.Functional_edit,'String',num2str(handles.Functional,'%2.0u')) %% coloca a % Functional com o valor pretendido

    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

if handles.Functional>100
    handles.Functional=100;
    assignin('base', 'Functional', handles.Functional/100); %/100 para ser com valores entre 0 e 1
    set(handles.Functional_edit,'String',num2str(handles.Functional,'%2.0u')) %% coloca a % Functional com o valor pretendido
    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

    h=errordlg('Maximum is 100%!');%mensagem de erro
    uiwait(h);
end
handles.output = hObject;
guidata(hObject, handles);

```

Botão “+”:

```
% --- Executes on button press in MoreMore_Funtional.
function MoreMore_Funtional_Callback(hObject, eventdata, handles)
% hObject    handle to MoreMore_Funtional (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    incremento=10;
    handles.Functional=handles.Functional+incremento; %pegando no valor da % Functional definido inicialmente incrementa 5

    assignin('base', 'Functional', handles.Functional/100); %/100 para ser com valores entre 0 e 1
    set(handles.Functional_edit,'String',num2str(handles.Functional,'%2.0u')) %% coloca a % Functional com o valor pretendido
    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

if handles.Functional>100
    handles.Functional=100;
    assignin('base', 'Functional', handles.Functional/100); %/100 para ser com valores entre 0 e 1
    set(handles.Functional_edit,'String',num2str(handles.Functional,'%2.0u')) %% coloca a % Functional com o valor pretendido
    set_param(handles.ModelName, 'SimulationCommand', 'update'); %Atualiza o valor da velocidade

    h=errordlg('Maximum is 100%!');%mensagem de erro
    uiwait(h);
end
handles.output = hObject;
guidata(hObject, handles);
```

Modo Configurações

Janela com Mensagem de Alerta:

```
msg = msgbox('You should use this mode with caution!','Attention','warn');
uiwait(msg);
```

Testes e Resultados

Para o Modo Bicicleta Estática Normal:

```
plot(Forca_dir,'b')
hold on
plot(Forca_esq,'r')
plot(Velocidade,'g')
plot(StepSide*20,'k')
plot(RegForce,'m')

grid on;
xlabel('Amostras')
legend('Força Dir(N)', 'Força Esq(N)', 'Velocidade(RPM)', '20- Ciclo da Perna Esquada; 40- Ciclo Perna Direita', 'Regulação de Força')
```

Para o Modo Perna Imóvel:

```
plot(Forca_dir,'b')
hold on
plot(Forca_esq,'r')
plot(Velocidade,'g')
plot(StepSide*20,'k')
plot(RegForce,'m')

grid on;
xlabel('Amostras')
legend('Força Dir(N)', 'Força Esq(N)', 'Velocidade(RPM)', '20- Ciclo da Perna Boa; 40- Ciclo Perna Debilitada', 'Regulação de Força')
```

Este *scripts* poderão ser também usados pelos terapeutas para analisar os dados adquiridos de um determinado paciente e assim ir verificando a sua evolução no processo de reabilitação.

Anexo C

Simulink – Esquemático

