



Universidade de Aveiro

Departamento de Física

Sistema de Assistência na Reabilitação de Traumatizados Medulares

Sónia Maria Vaz Semedo

Orientador: Prof. Dr. Fernão Vistulo de Abreu

Co-orientador: Prof. Dr. Vítor Manuel Ferreira dos Santos

Julho de 2008

Agradecimentos

Agradeço a todos os que contribuíram directa e indirectamente para o desenvolvimento deste trabalho destacando os seguintes: o professor Fernão Abreu, o professor Vítor Santos, ao colega Eduardo Durana, aos colegas do Laboratório de Automação e Robótica, aos colegas do Labmeeting.

Agradeço ainda aos meus familiares e amigos que sempre me apoiaram, a universidade e a todos os professores que constituem o meu currículo escolar.

Queria ainda agradecer ao engenheiro Arlindo da Veiga pelas orientações na programação, ao Sr. Ivo Mateus, ao Sr. Mário Rocha e ao Sr. Júlio Gonçalves pela disponibilidade na confecção das peças para os equipamentos.

Índice

1	<i>Introdução</i>	2
2	<i>Análise de Componentes Principais</i>	4
2.1	Conceitos matemáticos	4
2.2	O que é o PCA?	5
2.2.1	<i>Metodologia do PCA</i>	6
2.2.2	<i>Re-escalonamento dos dados</i>	8
2.2.3	<i>Determinação de números de pc's</i>	10
2.3	Aplicação do método	11
	<i>Interpretação dos resultados</i>	14
2.3.1	14
3	<i>Sensor de curvatura</i>	19
3.1	Hardware do sistema	19
3.1.1	<i>Descrição do modo de funcionamento do Hardware</i>	19
3.1.2	<i>Apresentação do equipamento</i>	21
3.2	Conversão analógica digital	21
3.3	Comunicação por porta série	22
3.4	Protocolo de comunicação	24
3.5	Software para o utilizador	25
4	<i>Assistente de exercícios</i>	27
4.1	Processamento de imagem	27
4.1.1	<i>Noções básicas de processamento de imagem</i>	27
4.1.2	<i>Aplicação do processamento de imagem ao trabalho</i>	29
4.2	Controlo do motor	31
4.2.1	<i>Modulação do PWM</i>	31
4.2.2	<i>Hardware do controlo</i>	32
4.2.3	<i>Protocolo de comunicação</i>	35
	Exercício proposto	35
4.3	35
	<i>Resultados e Perspectivas futuras</i>	36
5	36
	<i>Apêndice A</i>	37
	<i>Apêndice B</i>	38
	<i>Apêndice C</i>	42
	<i>Apêndice D</i>	47
	<i>Referências</i>	50

1 Introdução

Todos os anos surgem milhares de situações clínicas que resultam num défice motor, como por exemplo, lesões vertebromedulares, acidentes vasculares cerebrais e traumatismos crânio – encefálicos. Muitas vezes, a terapia intensiva ajuda a recuperar e leva a uma vida digna, menos dependente de terceiros: é uma realidade clínica que existem muitos pacientes que quando sujeitos a trabalhos de reabilitação intensivos têm uma grande probabilidade de recuperação. Também se sabe que a oferta de cuidados de fisioterapia e fisioterapia é pequena face à procura [22] [21] levando pacientes portugueses a procurarem programas de reabilitação em países que oferecem cuidados intensivos e a preços económicos. Por outro lado os custos envolvidos neste tipo de cuidados são, em geral, muito elevados (em Portugal superiores a 250€/dia). Por isso, é importante tornar os cuidados de reabilitação acessíveis a um maior número de pacientes explorando novas estratégias de reabilitação. Uma via possível consiste em identificar exercícios de reabilitação que, uma vez automatizados, não requeiram a presença contínua de um especialista. Esta solução poderá ser particularmente atractiva, se os equipamentos utilizados forem transportáveis e de baixo custo, para que os exercícios possam ser feitos tanto numa clínica como em casa.

No âmbito deste projecto, desenvolveram-se dois equipamentos que seguem esta filosofia. Neste trabalho, comecei por readaptar um equipamento anteriormente desenvolvido no laboratório, envolvendo fibras ópticas para a análise da curvatura. O objectivo inicial foi o de ganhar competências em instrumentação e software, e simultaneamente aumentar a funcionalidade do equipamento. Desenvolvi ainda um segundo equipamento a que chamei Assistente de Exercícios.

Estes equipamentos podem ser úteis num programa da reabilitação, pois dispensam a presença permanente de terceiros. O Assistente de Exercícios é controlado por computador e usa processamento de imagem para assistir o exercício do paciente. Os dados são recolhidos durante o exercício e podem ser enviados posteriormente para o clínico que acompanha o paciente. O sensor de curvatura pode ser útil para estudar o progresso de um paciente com problemas de flexão/extensão nos membros superiores e/ou inferiores. Os dados são também guardados, podendo ser analisados posteriormente.

Estes equipamentos oferecem vantagens tanto para o paciente como para o clínico, pois os exercícios são sistemáticos, e as análises quantitativas, tornando possível a detecção de progressos, mesmo que pequenos. Além disso cada equipamento e exercício pode ser adaptado às necessidades de cada paciente, e dada a sua portabilidade, o mesmo tipo de exercício pode ser usufruído por muitos pacientes com apenas um clínico a motorizá-los.

Uma vez que a análise de dados é uma tarefa essencial para a interpretação de uma série temporal, neste trabalho procurou-se também desenvolver um método quantitativo de análise de dados. O método desenvolvido usa a Análise de Componentes Principais (PCA – Principal Components Analysis) e foi aplicado em dados adquiridos no controlo da postura [5].

A Análise das Componentes Principais tem muitas áreas de aplicação e por ser um método capaz de identificar padrões correlacionados foi utilizado nesse projecto para o estudo de padrões de movimentos quando um paciente aponta fixamente um laser. A

análise dos resultados deste método pode servir para diagnóstico, tanto a nível da caracterização da deficiência motora como para o estudo do progresso de reabilitação. No futuro, penso que será possível adaptar este tipo de análise aos dados recolhidos pelo Assistente de Exercícios e pelo Sensor de Curvatura. No entanto, e porque o desenvolvimento dos instrumentos foi realizado em simultâneo ao do desenvolvimento da aplicação do método PCA, a análise das componentes principais ainda não foi aplicada aos resultados que poderiam ser obtidas com os instrumentos desenvolvidos.

No próximo capítulo vou descrever o método da Análise das Componentes Principais (PCA). Depois apresentarei o equipamento de fibra óptica, e de seguida o equipamento Assistente de Exercícios. Acabarei este trabalho por discutir os resultados obtidos e as perspectivas futuras. Durante todo o documento utilizarei a notação PCA para análise de componentes principais, pois esta classe de método designa-se em inglês por *principal components analysis*, sendo a sigla amplamente conhecida.

2 *Análise de Componentes Principais*

Neste capítulo pretende-se investigar a possibilidade de extrair informação relevante usando o método da análise das componentes principais. Para esse efeito utilizaram-se séries temporais fornecidas pela colega Ana Fonseca, e recolhidas no âmbito do seu projecto de licenciatura. A natureza dos exercícios realizados será discutida em maior detalhe na secção **Error! Reference source not found.**

Vou começar por abordar alguns conceitos matemáticos necessários para uma boa compreensão do método. Posteriormente descreverei o método do PCA, e discutirei as vantagens e conclusões que se podem extrair. Por fim discutirei os resultados obtidos nos exercícios de postura corporal e controlo motor.

2.1 *Conceitos matemáticos*

Para uma melhor compreensão do método do PCA, existem alguns conceitos básicos de estatística e de álgebra que se devem rever.

A estatística baseia-se na ideia da existência de um conjunto de dados e na análise das relações entre pontos individuais deste conjunto. Várias medidas estatísticas podem ser construídas a partir de um conjunto de dados. Porém neste trabalho vou apresentar apenas as necessárias para a compreensão do método do PCA.

Suponhamos que se tem um conjunto de dados $\{X_i\}$, correspondendo aos oito primeiros números primos:

$$X = [2 \ 3 \ 5 \ 7 \ 11 \ 13 \ 17 \ 19].$$

Para designar um elemento específico do conjunto usarei o índice subscrito para indicar a sua posição no conjunto. Por exemplo, X_5 representa o quinto elemento do conjunto. Também utilizarei a notação n para representar o número total de elementos do conjunto.

Pode-se calcular o valor médio ou a **média** do conjunto X usando a expressão:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}.$$

Porém a média não é suficiente para caracterizar o conjunto dos dados, pois podem-se ter dois conjuntos com a mesma média que sejam muito diferentes. Para descrever essa diferença pode calcular-se o **desvio padrão**:

$$s = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)}}.$$

Uma outra medida que pode descrever essa diferença é a **variância**:

$$s^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n-1)}.$$

Para o conjunto $\{X_i\}$ obtêm-se os seguintes resultados:

\bar{X}	S	S^2
9.62	6.39	40.83

O exemplo acima referido tem apenas uma variável, contudo muitos conjuntos de dados têm mais do que uma variável e a análise estatística destes conjuntos pode servir para estabelecer relações entre as diferentes variáveis. Como a variância e o desvio padrão operam só a uma dimensão, para estudar as correlações entre variáveis é importante definir a **covariância**:

$$s_{xy} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)},$$

Aqui, X e Y representam duas variáveis e s_{xy} é a covariância entre elas. Se a covariância fôr positiva isso significa que ambas as variáveis têm comportamento semelhante (por exemplo, quando uma é grande a outra tende a ser grande também); se for negativa significa que quando uma é grande a outra tende a ser pequena. Quando a covariância for desprezável significa que as variáveis são aproximadamente independentes.

Se o conjunto de dados tiver mais do que duas dimensões, então podemos definir mais do que uma covariância. Para um conjunto a $n -$ dimensões podemos calcular $\frac{n!}{(n-2)! \cdot 2}$ covariâncias, que podem ser dispostas numa matriz, a **matriz de covariância**.

Por exemplo, para um caso em que o conjunto de dados tem três dimensões tem-se a seguinte matriz:

$$S = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix}.$$

Os valores ao longo da diagonal são as próprias variâncias de cada variável. Esta matriz é simétrica, isto é, $s_{ij}=s_{ji}$.

Tendo a matriz de covariância podem-se calcular os seus valores próprios e os vectores próprios. Para uma matriz S, $N \times N$, podem-se calcular os valores próprios utilizando a expressão algébrica conhecida como equação característica:

$$\det(A - \lambda I_n) = 0,$$

onde I_n é a matriz identidade de dimensão n . Os vectores próprios são encontrados pela equação:

$$[A - \lambda_i I_n] X_i = 0$$

2.2 O que é o PCA?

Encontrar uma definição consensual do que é o PCA é uma tarefa árdua, pois cada autor define de maneira a ir ao encontro do objectivo da aplicação do método para o seu caso específico. Contudo em todas as possíveis definições existe uma base em comum. Sendo assim diria que o PCA é um método estatístico que permite reorganizar a informação contida num conjunto de dados, reduzindo a sua dimensionalidade enquanto retém a variabilidade dos mesmos.

Este método envolve um procedimento matemático que transforma variáveis possivelmente correlacionadas num pequeno número de variáveis não correlacionadas conhecidas como “*principal components*” (PC’s) que conseguem explicar a maior parte da variabilidade contida nos dados e permitindo identificar padrões e exprimir os dados de modo a salientar as suas semelhanças e diferenças sem grande perda de informação.

Pode ser utilizado em dados que contêm poucas variáveis, contudo é mais útil quando se têm muitas variáveis. Isso sucede de forma natural numa série temporal, onde cada instante numa sequência pode corresponder a uma variável.

Por ser um método que reduz a dimensionalidade dos dados sem grande perda de informação é utilizado em áreas muito diversas, tais como: nas neurociências, no reconhecimento de faces e movimentos, na compressão de imagens, em biologia, economia, física, e psicologia...

2.2.1 Metodologia do PCA

Supondo que se tem um conjunto de dados x_1, x_2, \dots, x_M onde x_i são variáveis com dimensões de vectores $N \times 1$, correspondendo a N observações para cada variável, aplica-se o PCA seguindo os passos seguintes.

Passo 1: calcular a média

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i.$$

Passo 2: subtrair a média

$$\tilde{x}_i = x_i - \bar{x}.$$

Passo 3: formar a matriz A ($N \times M$) e calcular a covariância

$$A = [\tilde{x}_1 \ \tilde{x}_2 \ \dots \ \tilde{x}_M]$$

$$S = \frac{1}{M} \sum_{n=1}^M \tilde{x}_n \tilde{x}_n^T = AA^T.$$

Passo 4: calcular os valores próprios (λ_i) e vectores próprios ($eigV_i$) da matriz de covariância.

Passo 5: escolher os vectores próprios com maior valor próprio e formar uma matriz de características, FV .

Passo 6: obter os resultados com as novas variáveis,

$$A_{new} = FV \times A.$$

Para melhor compreensão do método e da interpretação dos resultados consideremos o exemplo de um estudo realizado com 25 pessoas, sobre a relação entre o comprimento e largura da medida da cabeça do primeiro filho adulto. Os valores medidos foram:

Primeiro filho adulto	
Comprimento da cabeça	Largura da cabeça
x1	x2
191	155
195	149
181	148
183	153
176	144
208	157
189	150
197	159
188	152
192	150
179	158

183	147
174	150
190	159
188	151
163	137
195	155
186	153
181	145
175	140
192	154
174	143
176	139
197	167
190	163

Tabela 1 – valores das medidas do comprimento e da largura da cabeça de 25 pessoas

Fazendo um gráfico da largura em função do comprimento da cabeça (Figura 2.1), pode não ser totalmente clara a relação existente entre as duas variáveis. De maneira a se poderem identificar possíveis correlações entre as variáveis podemos então aplicar o método do PCA.

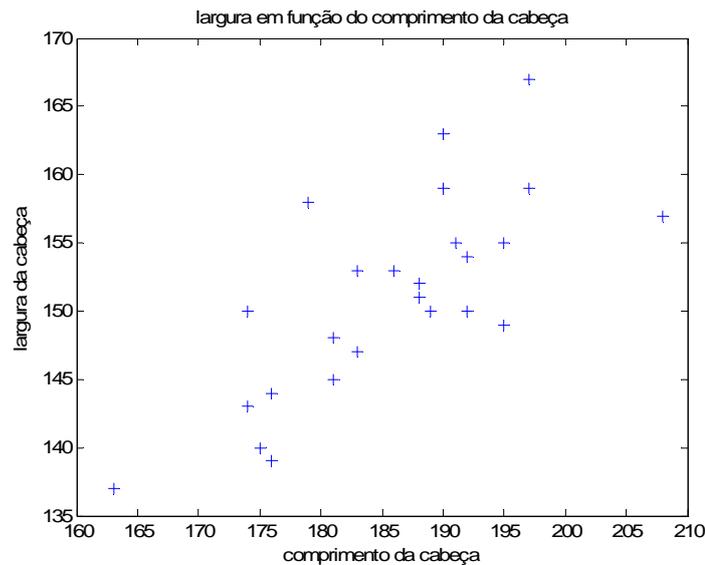


Figura 2.1 – Representação dos dados originais.

Neste caso, calcula-se a média obtendo $\bar{X} = [185.72 \ 151.12]$. Após subtrair a média a cada medição, calcula-se a matriz covariância, S,

$$S = \begin{bmatrix} S_1^2 & S_{12} \\ S_{21} & S_2^2 \end{bmatrix} = \begin{bmatrix} 95.29 & 52.87 \\ 52.87 & 54.36 \end{bmatrix},$$

onde os valores ao longo da diagonal são as variâncias de cada variável.

De seguida extraem-se os valores próprios, L, e determinam-se os respectivos vectores próprios, U:

$$L = \begin{bmatrix} 131.52 & 0 \\ 0 & 18.13 \end{bmatrix} \text{ e } U = \begin{bmatrix} .824 & -.565 \\ .565 & .824 \end{bmatrix}.$$

Os coeficientes dos vectores próprios são ortonormais e representam as direcções dos novos eixos de coordenadas em relação aos antigos. Estes vectores transformam as variáveis originais x_1 , x_2 em novas variáveis não correlacionadas z_1 e z_2 :

$$z = U' [X - \bar{X}].$$

Como foi acima referido estas novas variáveis são conhecidas como *componentes principais* (principal components, pc) e a i -nésima componente principal é:

$$z_i = u_i' [X - \bar{X}].$$

A variância destas novas variáveis é o valor próprio associado ao vector próprio u_i . Neste exemplo, têm-se duas pc's, À primeira componente, PC1, está associada cerca de 87.8% da variabilidade dos dados e um valor próprio de 131.52 e à segunda (PC2) cerca de 12.1% e um valor próprio de 18.13.

Para a redução da dimensão são escolhidas as pc's com maior valor próprio associado, pois estas são necessárias para explicar a maior variabilidade dos dados. Neste caso, pode-se reduzir a dimensão utilizando apenas o pc1 que contém a maioria da variabilidade dos dados. Geometricamente, o procedimento descreve o eixo principal de rotação dos eixos das coordenadas originais x_1 e x_2 em torno das suas médias. A direcção do pc1 está representada pela linha d1 e esta linha captura a maioria da variabilidade dos dados enquanto que a linha d2 representa a direcção do pc2.

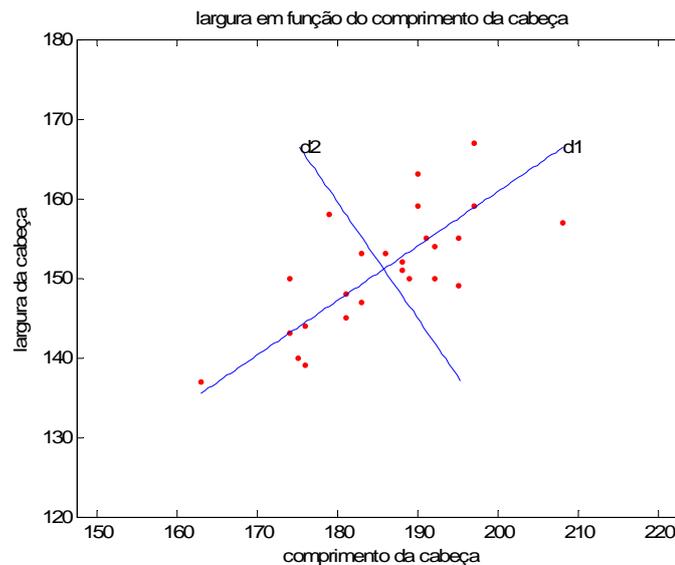


Figura 2.2 – Representação das direcções principais

Daqui se compreende que se podiam representar as duas variáveis originais razoavelmente utilizando apenas o pc1.

O mesmo procedimento utilizado para este exemplo a duas dimensões será utilizado para um caso p – dimensões, onde a dimensão da matriz covariância passará a ser $p \times p$ e ter-se-ão p pc's.

2.2.2 Re-escalamento dos dados

O método do PCA é muito sensível ao re-escalamento dos dados, pois dependendo do re-escalamento ter-se-ão vectores próprios diferentes e diferentes conclusões poderão ser tiradas. Por exemplo, medindo variáveis diferentes em diferentes

unidades, pode afectar os resultados. Por isso utilizar um reescalonamento dos dados pode ser crucial para obter resultados com significado.

Existem dois tipos de re-escalonamento dos dados: podem-se reescalonar os dados originais ou então os vectores próprios.

No re-escalonamento dos vectores próprios pode-se aplicar um dos três métodos possíveis:

- **Re-escalonamento à unidade** – consiste em não reescalonar os vectores próprios. Neste método os vectores são ortogonais, ortonormais e tem comprimento igual a unidade, conhecidos por alguns autores por **U-vectors**. Usando estes vectores para encontrar os *principal components* produzir-se-ão pc's não correlacionados e cuja variância é igual aos respectivos valores próprios.
- **Re-escalonamento à raiz dos valores próprios** – consiste em reescalonar os vectores próprios multiplicando-os pela raiz quadrada dos respectivos valores próprios. Os vectores que resultam deste reescalonamento são ortogonais e denominam-se por **V-vectors**. Os pc's resultantes deste método têm a vantagem de estarem nas mesmas unidades das variáveis originais, sendo a variância igual ao quadrado dos valores próprios.
- **Re-escalonamento ao inverso da raiz** – os vectores são multiplicados pelo inverso da raiz dos respectivos valores próprios, os vectores originados são ortogonais, conhecidos como **W-vectors**. Este método produz pc's ortogonais com variância igual a unidade.

Estes três tipos de re-escalonamento de vectores diferem uns dos outros apenas por um factor de escala, daí que a escolha é uma questão de gosto do utilizador, sendo que os U-vectors são úteis de um ponto de vista de diagnóstico, os V-vectors têm a vantagem de tanto estes vectores com os pc's que deles originam, terem as mesmas unidades das variáveis originais e os W-vectors produzirem pc's com variância igual a um.

Usando o exemplo do caso do primeiro filho adulto tem-se:

$$U = \begin{bmatrix} .824 & -.565 \\ .565 & .824 \end{bmatrix}$$

$$V = \begin{bmatrix} 9.442 & -2.405 \\ 6.479 & 3.507 \end{bmatrix}$$

$$W = \begin{bmatrix} .071 & -.132 \\ .049 & .193 \end{bmatrix}$$

Utilizando este tipo de re-escalonamento a direcção que contém maior variabilidade continua a mesma, a única diferença entre as escolhas é a variância dos pc's utilizados. Na escolha do U-vectors a variância dos pc's é igual ao valor próprio (131.52 e 18.13), se a escolha for V-vectors a variância será o quadrado do valor próprio (17297.51 e 328.69) e na escolha do W-vectors a variância será igual a unidade (1 e 1).

O re-escalonamento dos dados originais determina o tipo de matriz de dispersão que será utilizada no cálculo dos vectores próprios e pode-se utilizar um dos três métodos abaixo descritos:

- Os dados originais não são escalonados e a matriz de dispersão utilizada é a matriz produto. Neste caso os dados mantêm a média e a variância. Cada vector da matriz será a variável x_i .

- Os dados originais são normalizados de maneira a terem média zero e a matriz de dispersão será a matriz covariância. Os vectores utilizados para o cálculo da matriz covariância são $x_i - \bar{x}_i$.
- Os dados são normalizados de maneira a terem media zero e variância igual a unidade e a matriz de dispersão utilizada será a matriz correlação. Os vectores utilizados para o cálculo dessa matriz são $\frac{(x_i - \bar{x}_i)}{s_i}$.

Existem vantagens na escolha do tipo de reescalonamento, pois utilizar a matriz covariância para determinar os vectores próprios é muito vantajoso numericamente quando as variáveis têm muitos dígitos, enquanto que utilizar a matriz produto é muito útil no campo da química.

Existem duas razões para escolher a matriz correlação, se as variáveis têm unidades diferentes ou então se a variância for muito diferente em termos de grandeza. Utilizando esta normalização as variáveis passarão a ter o mesmo peso em termos de variabilidade pois cada variável passará a ter variância um.

Para melhor compreensão do reescalonamento dos dados está descrito no **Apêndice D** o exemplo de um estudo sobre características de vinhos. Este exemplo elucidará as vantagens e as conclusões que surgem dependendo do reescalonamento utilizado.

2.2.3 Determinação de números de pc's

O método do PCA como se sabe é um método que tem como um dos objectivos reduzir a dimensão, mas com pouca perda de informação. No entanto, quantos mais pc's se utilizarem mais informação se terá e por isso deve existir um compromisso entre a redução dimensional e a informação contida, surgindo por isso a questão de saber “ quantos PC's se devem usar? ”.

Existem alguns métodos estatísticos que auxiliam na determinação do número de pc's a utilizar na representação das variáveis. Porém neste trabalho abordar-se-á apenas o método do *scree plot*.

Este método consiste na visualização gráfica dos valores próprios. Regra geral o critério utilizado para escolher o número de pc's é o estudo da quebra que surge no gráfico onde são escolhidos todos os valores acima da quebra e o primeiro abaixo. Por exemplo, de acordo com o scree plot típico apresentado na Figura 2.3 seriam escolhidos seis pc's para representar as variáveis originais.

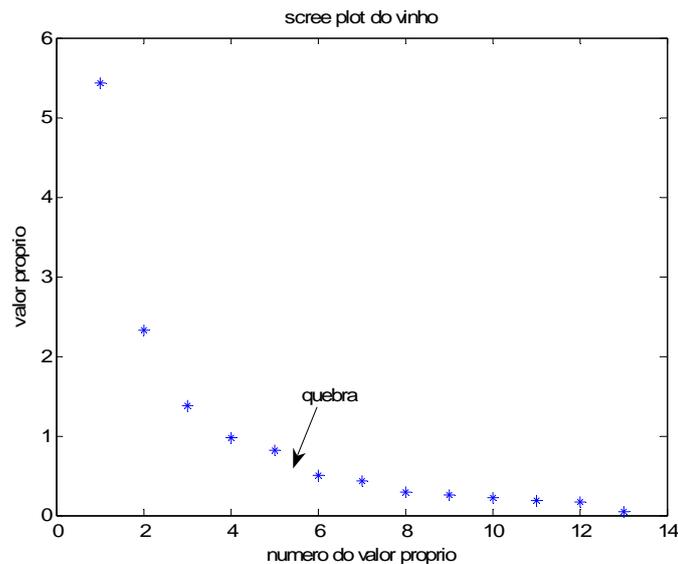


Figura 2.3 – Scree plot típico

Os valores próprios utilizados no scree plot ilustrado em cima foram extraídos da matriz de correlação onde cada variável tem uma variância igual à unidade. Neste caso pode-se utilizar o *Cattell's scree plot* onde são escolhidos apenas os valores próprios com valor superior a um, pois apenas estes contêm a variabilidade correspondendo a mais do que uma variável.

2.3 Aplicação do método

Para a aplicação do método do PCA no presente projecto utilizei dados fornecidos pela colega Ana Rita Fonseca. Ela filmou a trajectória de um laser segurado fixamente por um paciente e registou vídeos com 10800 frames. Posteriormente trataram-se essas imagens determinando a posição do “centro de massa” da luz laser para cada frame do vídeo.

O objectivo principal da aplicação do método foi o de tentar identificar padrões de movimentos, pois um sujeito ao apontar fixamente para um ponto oscila em torno de uma posição de equilíbrio, mas involuntariamente faz movimentos bruscos que o afastam do ponto de equilíbrio, ainda que posteriormente retorne aproximadamente à posição inicial. Pretendem-se então estudar esses movimentos e interpretar os resultados.

A partir dos dados das coordenadas espaciais, (x,y), calculei a média móvel para suavizar a trajectória do centro de massa do laser:

$$xm_i = \frac{\sum_{n=i}^{i+d-1} x_n}{d}.$$

Aqui d representa a largura do intervalo ou o número de elementos utilizados no cálculo da média móvel.

Todas as operações foram efectuadas ao conjunto de dados da média móvel $\{xm_i, ym_i\}$. As operações são análogas para ambas as coordenadas, representarei então apenas as operações efectuadas a xm_i .

Determinei as componentes da velocidade e calculei a norma:

$$v_{xi} = xm_{i+1} - xm_i$$

$$\|v_i\| = \sqrt{v_{xi}^2 + v_{yi}^2}$$

Na Figura 2.4 pode-se ver o histograma da norma do vector velocidade, para uma série onde o espaçamento temporal entre pontos da série corresponde a dois frames, i.e., a intervalos de tempo $\Delta t=0.01s$, pois a feita noventa aquisições por segundo.

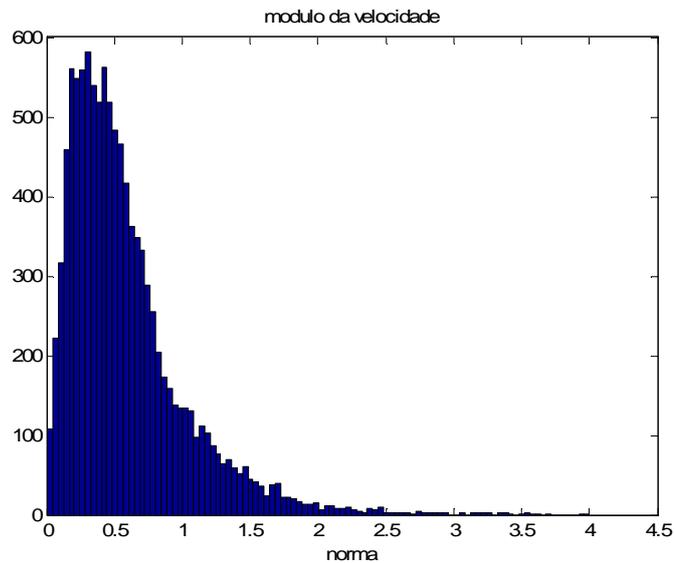


Figura 2.4 – Histograma da norma da velocidade

Calculei o ângulo, θ , entre dois vectores velocidade consecutivos:

$$\theta_i = \sin^{-1} \left(\frac{(v_{i+1} \times v_i)_z}{\|v_{i+1}\| \cdot \|v_i\|} \right)$$

que é uma medida da aceleração normal. As acelerações relacionam-se com forças, e por isso também com decisões motoras no controlo motor. O produto externo entre dois vectores é um vector perpendicular a ambos. Como os vectores velocidade só têm duas componentes não nulas (pois descrevem o movimento do laser na superfície onde este é projectado) o vector $v_{i+1} \times v_i$ só tem uma componente não nula, que é a usada no cálculo do ângulo. O sinal desta componente é também utilizado para distinguir ângulos positivos de negativos como se pode ver no histograma da Figura 2.5. Para evitar problemas de indeterminação excluem-se todos os vectores cuja norma seja inferior a 0.05. Como se pode ver no histograma na Figura 2.4, existe uma centena de vectores com o valor da norma inferior a esse limite que serão excluídos da análise.

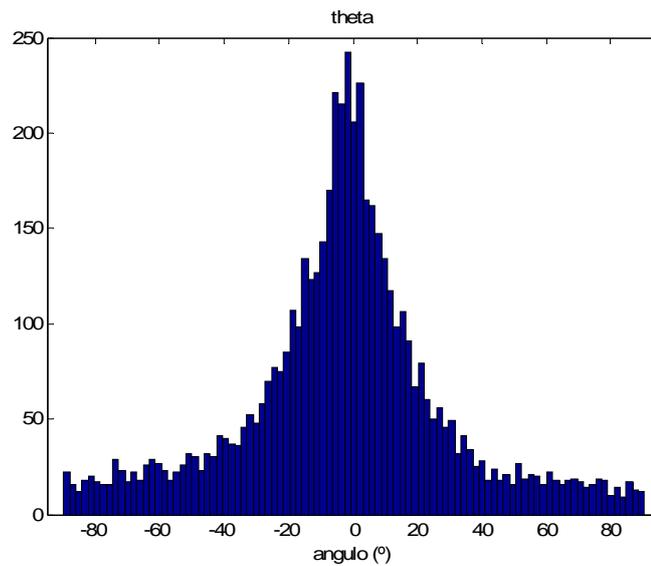


Figura 2.5 – Histograma dos ângulos

Conhecendo a norma da velocidade e o ângulo entre dois vectores velocidade sucessivos, defini critérios para seleccionar séries às quais apliquei o método PCA. Como o objectivo foi o de identificar padrões de movimentos, fragmentou-se a trajectória da posição do laser em pequenas sub-séries. Para isso tive que encontrar um compromisso no número de pontos a considerar em cada sub-série, *npts*: se a dimensionalidade da sub-série fosse muito elevada poderíamos obter uma enorme diversidade movimentos podendo ser difícil identificar padrões; para séries demasiado curtas, os resultados correm o risco de se tornar triviais. Assim, usamos *npts*=7, o que corresponde a determinar *npts-2*=5 ângulos na sub-série.

Um outro parâmetro que foi necessário definir foi o intervalo de tempo (que designei por *step*) usado na discretização da velocidade. Intervalos muito pequenos são mais susceptíveis de serem contaminados com ruído proveniente de uma certa falta de precisão na determinação da posição do laser, após a segmentação da imagem. Este efeito pode ainda ser reduzido após a aplicação da média móvel acima definida. Assim defini assim *d*=9, *npts*=7 e *step*=2. Todos medidos em números de frames.

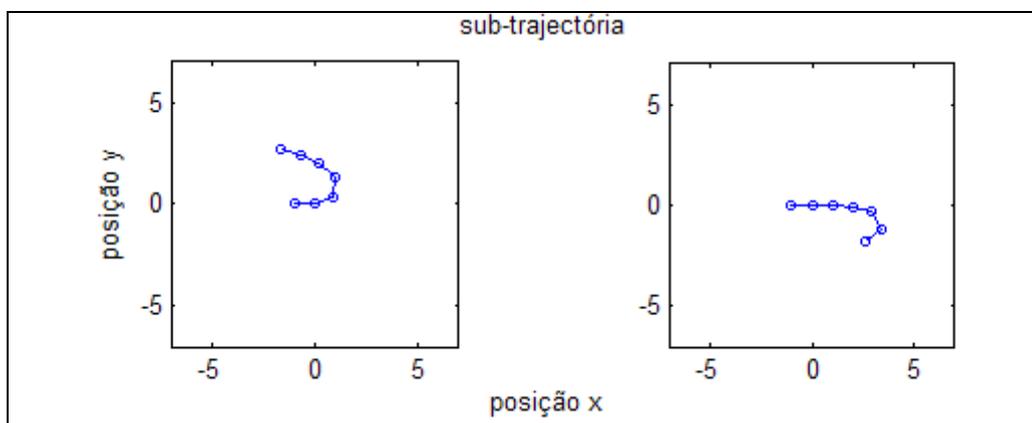


Figura 2.6 – Exemplo de sub-trajectórias (*d*=9, *npts*=7, *step*=2)

Estabelecido o compromisso, foi ainda importante seleccionar um subconjunto de sub-trajectórias a utilizar, pois é importante analisar sub-trajectórias que possam ter uma situação física (ou fisiológica) semelhante. Decidi utilizar como critério de escolha a ocorrência de um valor mínimo para a norma da velocidade, pois os movimentos bruscos que uma pessoa faz para retomar a posição de equilíbrio têm uma velocidade elevada e correspondem a situações semelhantes física ou fisiologicamente.

Para definir qual deveria ser o limiar da velocidade estudei o histograma da velocidade, e posteriormente escolhi um valor mínimo a partir do qual uma sub-trajectória é iniciada. Assim, quando num instante a norma da velocidade é superior a este valor, é construída uma sub-série com os ângulos dos sucessivos vectores velocidade.

Organizei estes valores numa matriz $[M \times N]$ onde M depende do número de sub-séries que obedecem à condição e N é igual ao número de ângulos, a ser utilizada para a aplicação do método PCA. Aplicando o método PCA a essa matriz, obtive a matriz de correlação $[N \times N]$, e depois calculei os seus valores e vectores próprios.

2.3.1 Interpretação dos resultados

Analisando os resultados da aplicação do método do PCA com todos os critérios definidos acima, pude constatar que dos dados que tinha à minha disposição apareceram dois casos distintos. Os coeficientes do vector próprio que armazena a maioria da variabilidade dos dados para um caso podiam ser todos positivos (**caso soma**) ou então havia coeficientes com sinais diferentes (**caso diferença**), pode-se constatar isso nas tabelas seguintes que representam os vectores e valores próprios para dois casos distintos.

	PC1	PC2	PC3	PC4	PC5
	0,3920	-0,4781	0,6061	0,4210	-0,2704
	0,4977	-0,4168	-0,0391	-0,5329	0,5413
	0,5080	-0,0596	-0,6126	-0,0445	-0,6009
	0,4773	0,4509	-0,1458	0,5759	0,4647
	0,3358	0,6252	0,4843	-0,4529	-0,2383
Valor próprio	2,5990	1,1842	0,6964	0,3455	0,1748
% Variância	51,9806	23,6843	13,9290	6,9101	3,4960

Tabela 2 – resultados em que todos os ângulos têm um peso positivo para o PC1 (caso soma)

	PC1	PC2	PC3	PC4	PC5
	-0,5084	0,1587	-0,6228	0,4672	0,3319
	-0,6528	-0,1121	-0,1079	-0,3529	-0,6520
	-0,4444	-0,5021	0,3772	-0,2328	0,5949
	0,1040	-0,7255	0,0190	0,6053	-0,3100
	0,3272	-0,4286	-0,6766	-0,4865	0,1214
Valor próprio	1,8970	1,5183	0,9041	0,4238	0,2568
% Variância	37,9400	30,3660	18,0816	8,4754	5,1358

Tabela 3 – resultados em que o peso dos ângulos no PC1 é a diferença entre os três primeiros ângulos e os dois últimos (caso diferença).

Identifiquei estes casos como **caso soma** e **caso diferença**, para facilitar a descrição das conclusões que tirei destes resultados.

Para ambos os casos utilizei o método do Cattell's scree plot para determinar o número de pc's a utilizar para a representação dos dados originais, sendo preciso apenas dois pc's para representar a maioria da variabilidade dos dados como se pode ver na Figura

2.7 e na Figura 2.8. Para o caso soma com dois pc's representa-se 75.66% da variabilidade e para o caso diferença 74.50%.

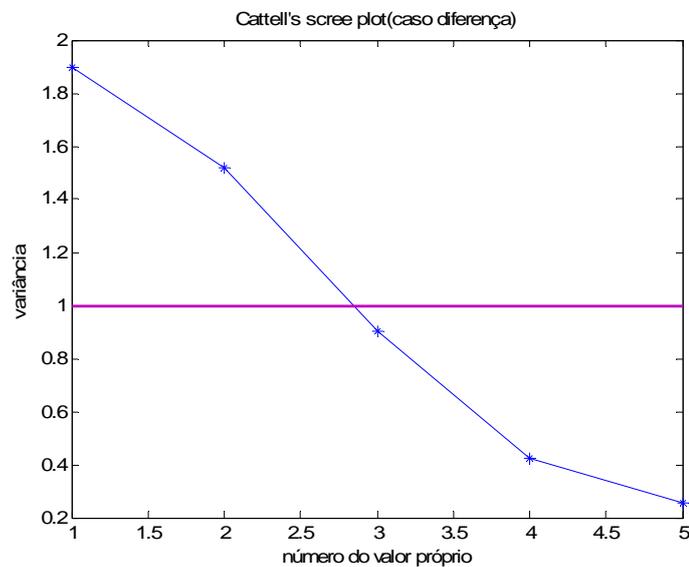


Figura 2.7- Scree plot para o caso diferença. Foram escolhidos dois vectores próprios para determinar as novas variáveis.

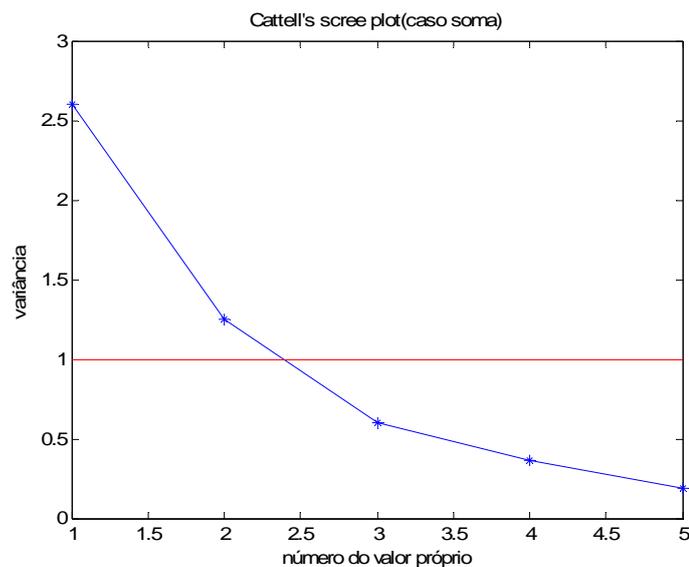


Figura 2.8-- Scree plot para o caso soma. Foram escolhidos dois vectores próprios para determinar as novas variáveis.

Para poder identificar possíveis padrões nos meus dados estudei as novas variáveis separadamente, isto é, estudei as variáveis apenas com o peso do vector próprio com maior valor próprio associado.

Para o caso soma verifiquei que existe uma certa persistência nas trajectórias definidas pelos *z-scores*. Quando o *z-score* for positivo significa que a trajectória seja positiva ($y > 0$) e quando for negativo a trajectória tende a ser negativa ($y < 0$), como se pode verificar na Figura 2.9 e na Figura 2.10, respectivamente.

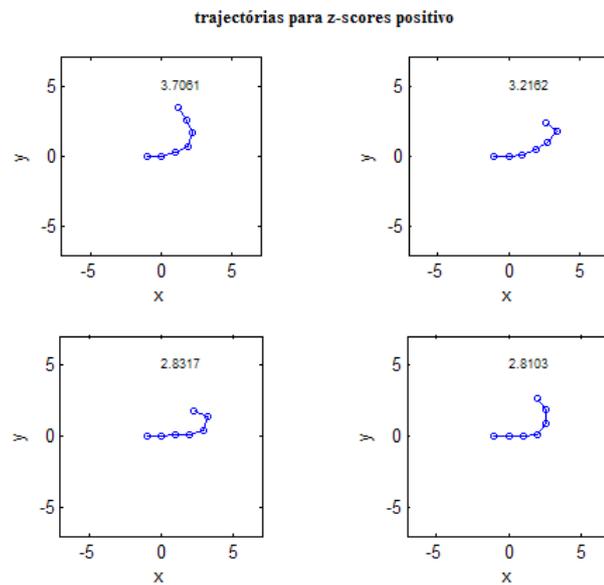


Figura 2.9- Trajectória com z1-scores positivos para o caso soma.

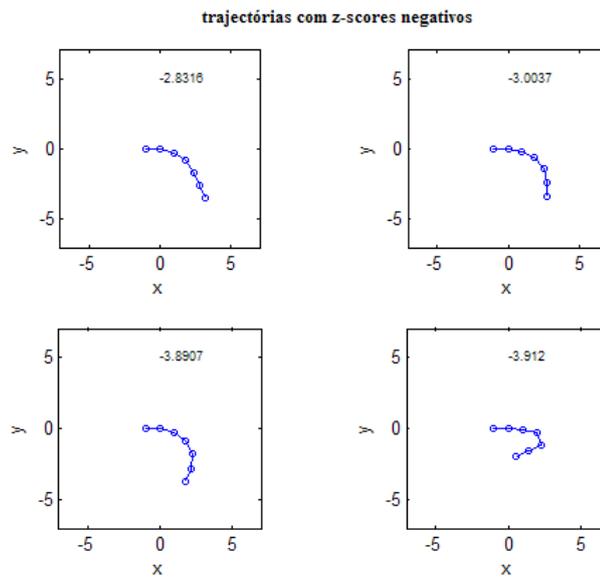


Figura 2.10- Trajectória com z1-scores negativos para o caso soma.

Para o caso diferença quando o z-scores for positivo a trajectória tende a ser negativa ($y < 0$) e quando for negativa a trajectória tende a ser positiva ($y > 0$) como se pode constatar na Figura 2.11 e na Figura 2.12.

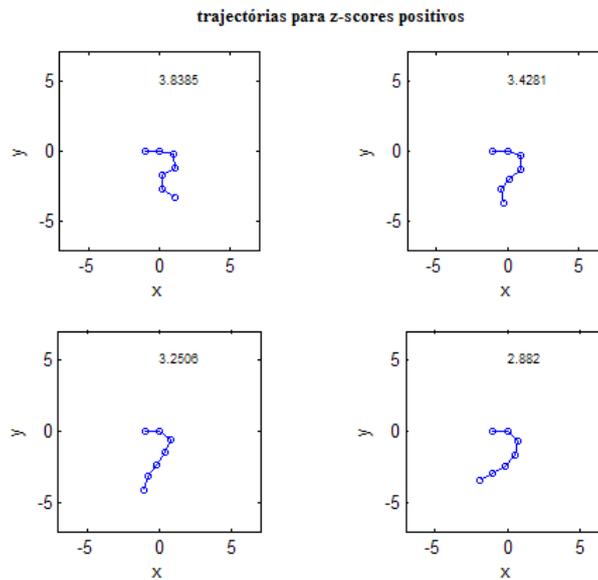


Figura 2.11- Trajetória com z1-scores positivo para o caso diferença

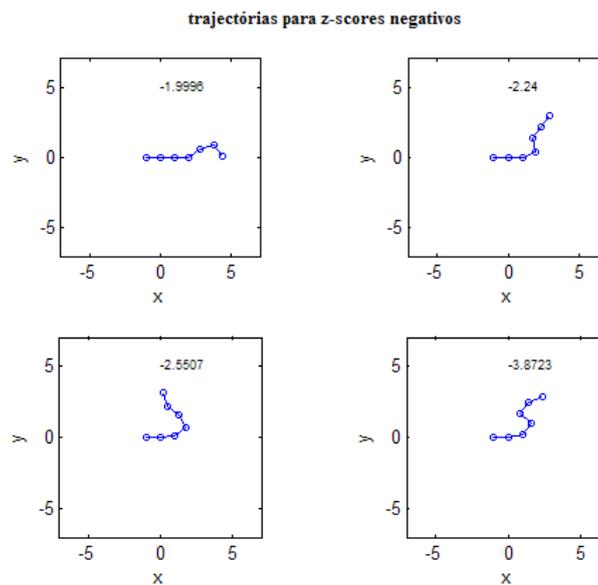


Figura 2.12- Trajetória com z1-scores negativo para o caso diferença

Comparando as figuras para ambos os casos pode notar que a diferença entre eles vai além do sentido da trajetória, pois também varia o tipo de trajetória efectuada para cada z-scores. Para o caso soma para valores do pc1 elevados em valor absoluto os ângulos das trajetórias tendem a ser todos positivos ou todos negativos não ocorrendo uma mudança da direcção da trajetória (movimentos brownianos), enquanto que para o caso diferença em algum momento ocorre uma mudança na direcção da trajetória devido a mudança do sinal dos ângulos (movimentos gaussianos), como se pode ver na Figura 2.13.

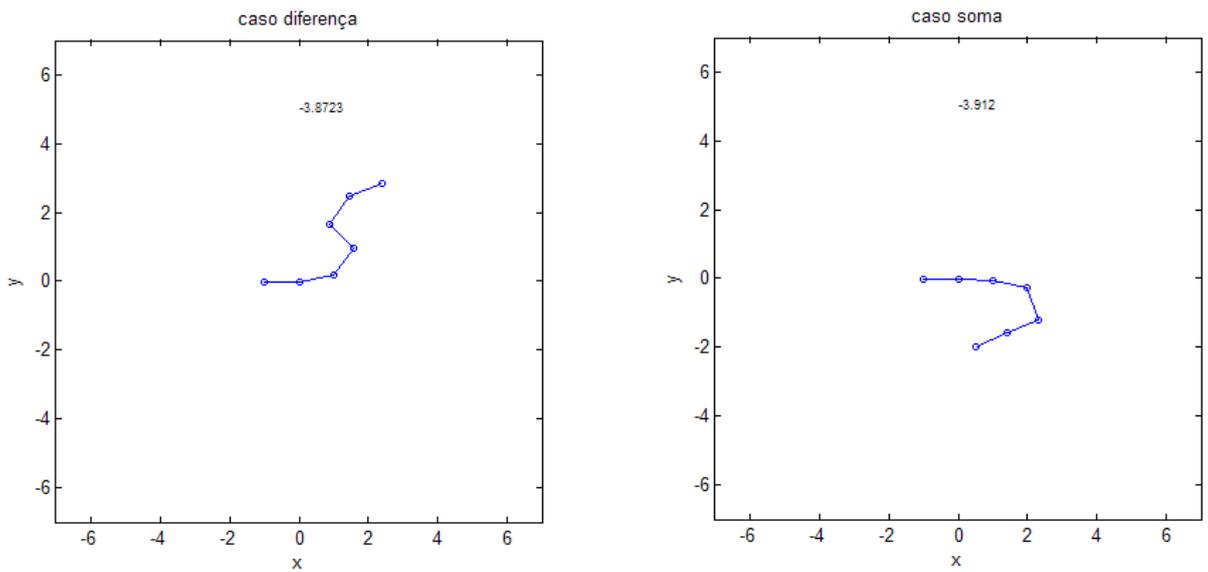


Figura 2.13- Diferença entre os dois tipos de trajetórias identificados na aplicação do método PCA.

Utilizando os dois pc's na representação das trajetórias o resultado é uma relação entre o peso do pc1 e o peso do pc2, continuando o pc1 a ser responsável pelo sentido e tipo da trajetória e pc2 funciona como um ruído associado ao movimento. Se o ruído for muito grande pode ocorrer mudanças na trajetória (curvatura, ponto de viragem).

Conclui-se então que para o caso soma o sujeito faz mais movimentos típicos de um movimento browniano e para o caso diferença o sujeito faz movimentos típicos de um movimento gaussiano.

3 Sensor de curvatura

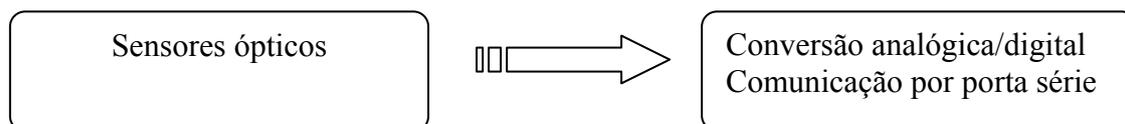
Neste capítulo apresentarei os aperfeiçoamentos que desenvolvi num equipamento de fibra óptica desenvolvido pela colega Mónica Cameirão no âmbito do seu projecto de mestrado [6]. Neste trabalho, foram medidas as perdas da fibra ao longo da curvatura causada pela flexão/extensão dum dedo, e registados esses valores através de software desenvolvido em Labview. Ao readaptar o equipamento fiz melhorias de modo a que a comunicação com o computador fosse mais flexível e que o equipamento registasse valores provenientes de dois sensores de fibra óptica em simultâneo.

3.1 Hardware do sistema

O funcionamento base deste equipamento é medir e registar as perdas que uma fibra óptica sofre quando é curvada. Para alcançar esse objectivo defini os seguintes passos:

- Detectar e medir as perdas da fibra;
- Fazer a conversão analógico/digital;
- Registar o valor convertido.

Estando definidos os passos que me levaram ao desenvolvimento do equipamento, dividi-o em dois blocos, o bloco dos sensores ópticos e o bloco da conversão analógica/digital e comunicação por porta série.



O bloco dos sensores ópticos é composto por dois sensores de fibra óptica e um circuito integrado amplificador. Cada sensor óptico é um Kit comercial (Experimenter's Kit da Industrial Fiber Optic, Inc) constituído por um LED de infravermelho (IF-E91A) e por um fototransistor (IF-D92), emissor e receptor respectivamente, com invólucros plásticos que permitem adaptar uma fibra óptica de 1000 μm (IF-C-E1000). O circuito integrado (LM324) é composto por quatro amplificadores operacionais internos independentes.

O bloco da conversão é composto por um micro-controlador PIC, um MAX232 e um DB9. O PIC utilizado é PIC18F2550, tem 28 pinos com várias funcionalidades entre elas a conversão analógica/digital e a comunicação por porta série.

3.1.1 Descrição do modo de funcionamento do Hardware

A corrente que atravessa o LED do emissor óptico, faz com que este emita luz, o sinal é transportado ao longo da fibra óptica até ao fototransistor, que transforma o sinal óptico em sinal eléctrico. Este sinal eléctrico passa por um estágio de amplificação e depois convertido no PIC para um sinal digital que será posteriormente enviado através da comunicação por porta série para o computador aonde será guardado.

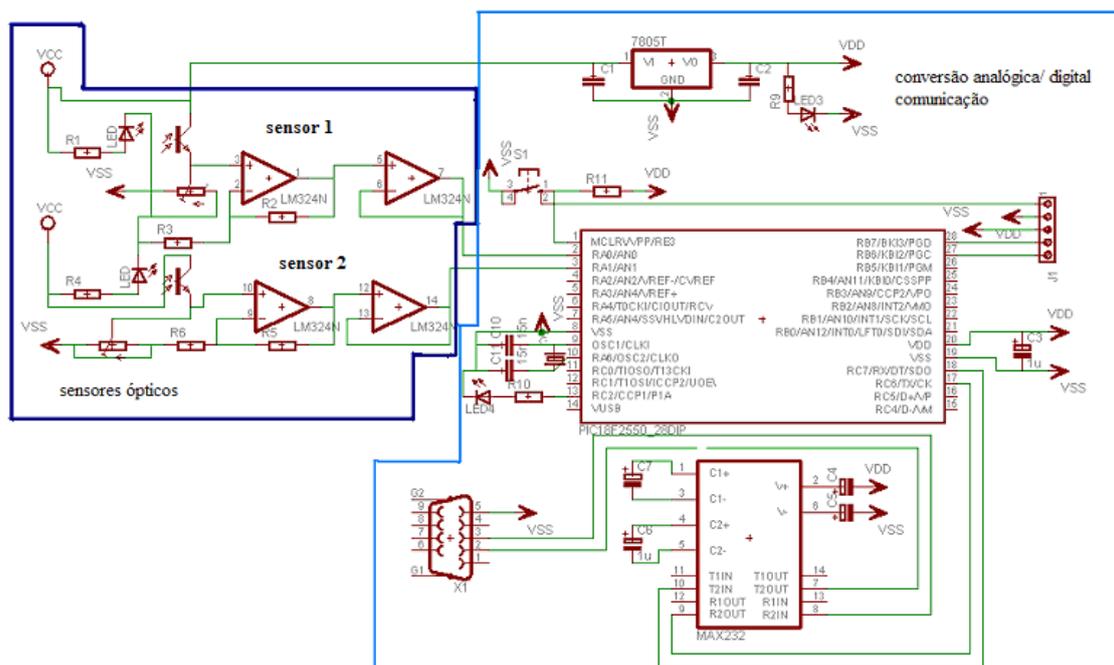


Figura 3.1 – Esquemático do circuito do sistema de curvatura com os dois blocos identificados

Durante a implementação do circuito tomei algumas precauções para garantir o bom funcionamento do sistema, tais como: a corrente que atravessa o LED, as possíveis flutuações do sistema e o valor máximo da tensão a entrada do PIC.

VCC= 9V
VSS= 0V
VDD= 5V
R1=R4=2.2KΩ
R2=R5=10 KΩ
R3=R6=1 KΩ
Potenciômetros=10 KΩ

Tabela 4 – tabela de correspondência dos símbolos aos valores para o funcionamento do equipamento

Sabendo que o sistema funciona com uma alimentação VCC= 9V e que o LED tem uma tensão de polarização directa de 1.2V, pode-se calcular o valor da corrente que atravessa o LED recorrendo a lei de Ohm ($V=RI$), onde V corresponde à tensão nos terminais da resistência R. No caso do sensor 1 a resistência seria R1. Foi escolhida propositadamente uma resistência grande de modo a ter uma corrente baixa, pois quando a corrente é muito elevada a intensidade da luz também o é, e se a intensidade da luz transmitida for elevada não se verificam alterações no sinal recebido devido a saturação do fototransistor. Sendo assim a corrente que atravessa o LED neste equipamento é de $I = (9 - 1.2) / 2200 = 3.5 \text{ mA}$. Para o sensor 2 o raciocínio é análogo. Utilizando uma resistência grande resolve-se o problema da detecção das variações da intensidade de luz recebida pelo receptor quando a fibra sofre curvaturas.

O sinal à saída do receptor da ordem dos milivolts sendo por isso necessário amplificá-lo. O circuito de amplificação pode chegar a ter um ganho de 11 vezes, pelo que

usa também um potenciômetro para regular o sinal. Como o PIC trabalha a cinco volts, regulei o potenciômetro para quando a fibra estivesse estendida correspondesse a tensão máxima suportada pelo micro-controlador, evitando tensões elevadas para o PIC.

Resolvidas as questões técnicas a montagem do circuito na placa experimental ficou com o aspecto da Figura 3.2.

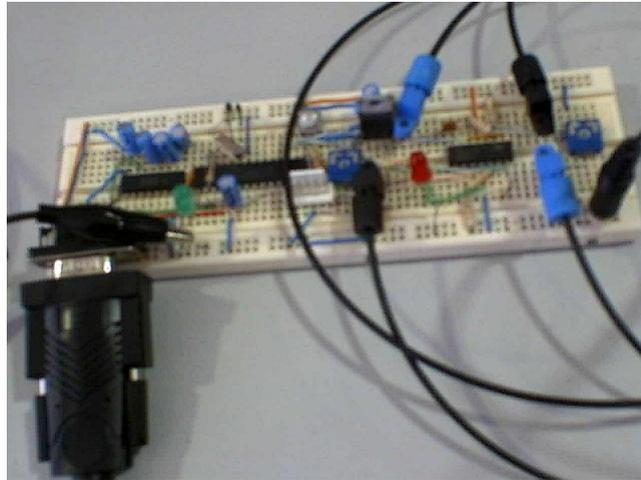


Figura 3.2 – Circuito da fibra óptica na placa de montagem

3.1.2 Apresentação do equipamento

Testado o circuito satisfatoriamente, fiz a respectiva placa de circuito impresso que ficou acondicionada em caixa própria, como se pode ver na Figura 3.3.

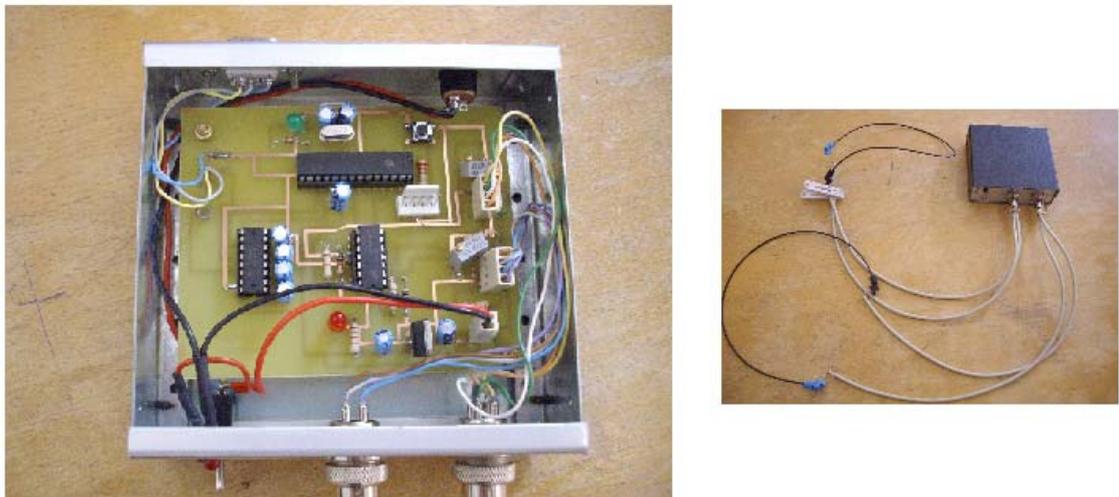


Figura 3.3- Circuito do sensor de curvatura em caixa própria.

3.2 Conversão analógica digital

O sinal óptico emitido pelo LED é convertido num sinal eléctrico no receptor sendo este convertido para digital pelo ADC incorporado no PIC.

Para que essa conversão seja feita é preciso programar o PIC. O software utilizado para esse fim foi o MPLAB IDE que para além de servir para escrever código de programas a serem carregados para o micro-controlador, compila, simula, faz detecção de

erros ('debugging') e faz a ligação para código de máquina apropriada para transferir para o micro-controlador. A linguagem de programação utilizada foi o C, e o compilador foi o MCC18 (Microchip C Compiler, para controladores da família 18F).

Para programar o PIC de modo a fazer a conversão analógica/digital têm que se definir os cinco registos seguintes: ADRESH, ADRESL, ADCON0, ADCON1 e ADCON2.

Os registos ADRESH (Analog-to-digital result high register) e ADRESL (Analog-to-digital result low register) são responsáveis pela resolução. Neste PIC podem-se fazer conversões até 10 bits de resolução. Porém assumi que não precisava de muita resolução, tendo utilizado 8 bit de resolução (valores entre 0 e 255).

O registo ADCON0 (Analog-to-digital control register 0) controla a operação de conversão. É neste registo que está definido qual é o sinal a ser convertido, isto é, de que canal vem esse sinal. O registo ADCON1 (Analog-to-digital control register 1) configura a função de cada pino do PORTA, configurei dois analógicos (AN0 e AN1) e os restantes como digitais. O registo ADCON2 (Analog-to-digital control register 2) configura o tempo de aquisição e a frequência de conversão. Usei uma frequência de conversão de 12MHz, dando tempo de segurança ao conversor para funcionar correctamente.

Uma possível conversão pode ser vista na figura em baixo, onde fiz um 'debugging' para verificar se a conversão funcionava. A entrada do pino AN0 tinha um valor de tensão de 3V, fazendo a conversão este valor vale 153 (99 na representação hexadecimal).

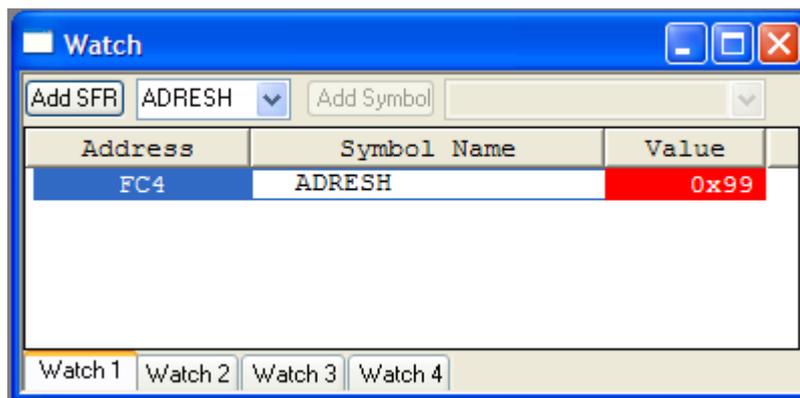


Figura 3.4 – Valor do sinal convertido na representação hexadecimal

3.3 Comunicação por porta série

A comunicação entre o equipamento e um computador é feito usando a porta série. Sendo assim, torna-se necessário ter uma noção geral de como funciona a comunicação de dados entre equipamentos que implementam o padrão EIA232 também conhecido como RS-232. O uso deste padrão foi determinado para aproveitar o módulo EUSART do PIC que possibilita comunicação via porta série. A padronização inicial apenas especificou as características eléctricas e o pinout das portas. No entanto, a forma dos conectores, a codificação, o formato dos dados e a comunicação assíncrona integrou, por força de uso, o padrão RS-232. Os PC normalmente são equipados com porta série com conectores DB9 macho e usam a codificação ASCII.



Figura 3.5 – Configurações dos pinos do DB9

Na conexão entre o PC e o PIC, apenas preciso de três sinais: sinal de terra (GND- pino 5), transmissão de dados (TX- pino 3) e recepção de dados (RX- pino 2).

É enviado um bit de cada vez. O formato típico de uma comunicação assíncrona é o seguinte: um bit inicial (start bit), os bits de dados (7 ou 8 bits), bit de paridade e bit final.

Utilizei um circuito integrado conhecido por MAX232 que é composto por dois receptores e dois transmissores. Cada receptor converte o sinal EIA-232 para os 5V dos TTL/CMOS e cada transmissor converte o sinal dos TTL/CMOS para sinal EIA-232, possibilitando a comunicação computador – PIC.

Para configurar a transmissão e recepção usei o módulo USART (Universal Synchronous Asynchronous Receiver Transmitter) do PIC em modo assíncrono, que usa o formato non-return-to-zero com os bits a comunicar enquadrados entre um start bit e um stop bit. O USART é controlado através de três registros: o TXSTA (Transmit Status and Control), o RCSTA (Receive Status and Control) e o BAUDCON (Baud Rate Control). No programa trabalhei tanto com a transmissão como a recepção de 8 bits, a baixa velocidade e um baud rate de 9600 bits/ segundo.

Para que a transmissão e recepção sejam realizadas é necessário que a informação chegue ao PIC através do pino RX e saia através do pino TX, como esquematizado na Figura 3.1 e descrito nos diagramas de bloco.

O diagrama de bloco da transmissão USART representado na Figura 3.6 descreve o módulo da transmissão. Para iniciar a transmissão tem que se definir o baud rate (SPBRGH:SPBRG) e a polaridade (TXCKP) dos dados. O byte a ser transmitido é colocado no TXREG (Read/Write Transmit Buffer Register). O TSR (Transmit Shift Register) armazena esse byte e envia para o TX pin. O estado do TXREG é indicado pelo TXIF e do TSR pelo TRMT. O TSR é limpo quando o stop bit for transmitido e pode-se então recomeçar a transmissão.

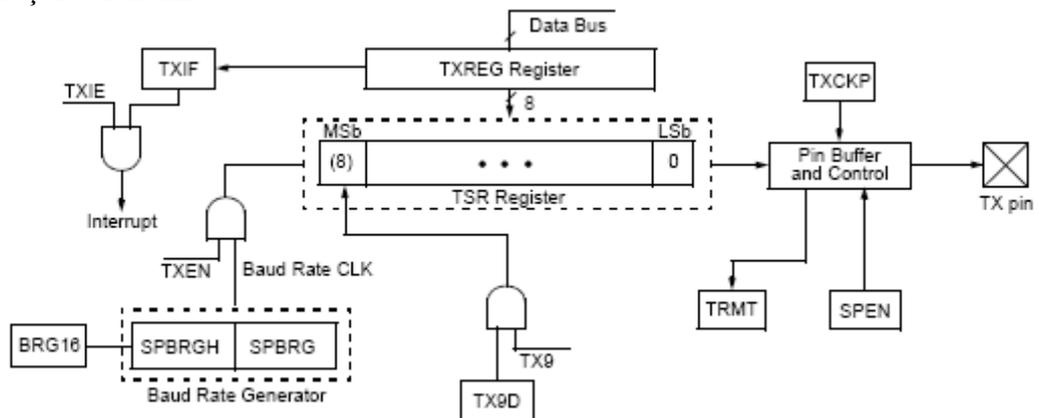


Figura 3.6 – Diagrama de bloco da transmissão USART

Para a recepção USART também é preciso definir o baud rate (SPBRGH:SPBRG) e a polaridade (RXDTP) dos dados. O sinal recebido pelo RX pin é guiado pelo Data Recovery com uma velocidade dezasseis vezes maior que o baud rate e armazenado no RSR (Receive Shift register). O sinal é então enviado para o RCREG e pode-se recomeçar a recepção como se pode ver na Figura 3.7.

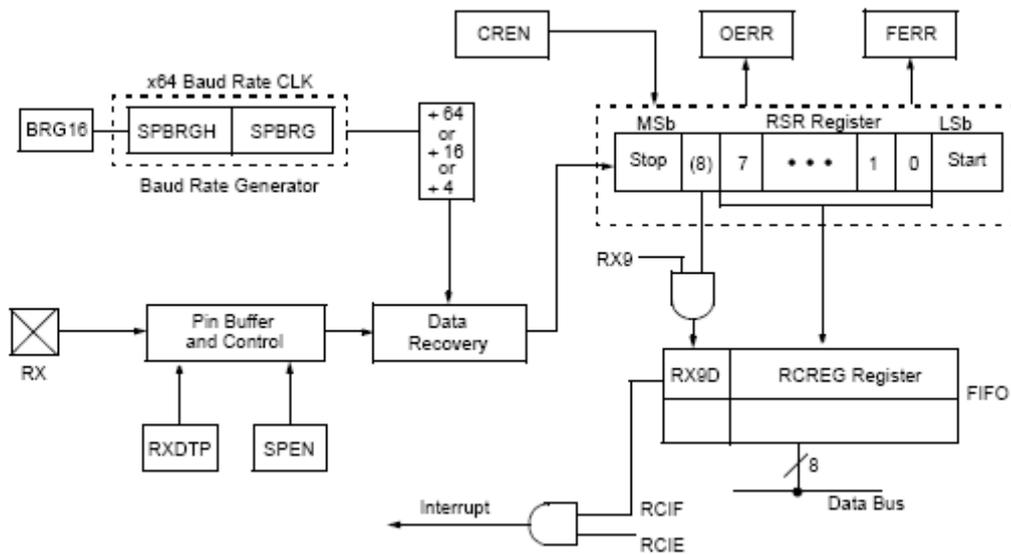


Figura 3.7 – Diagrama de bloco da recepção USART.

Para verificar se a comunicação estava a funcionar utilizei o Hyper Terminal, onde enviava um carácter e recebia esse mesmo carácter de volta.

3.4 Protocolo de comunicação

De forma a determinar quando se inicia o envio de valores para o PC e também controlar a frequência de envio, criei um protocolo simples para a comunicação entre o PIC e o computador. Se o PIC receber o carácter 'R' o valor transmitido será o valor da conversão proveniente do sensor 1, e se for o carácter 'L' o valor transmitido é o valor da conversão do sensor 2. Para esse efeito desenvolvi um programa no Matlab, onde defino qual o valor que quero converter.

Um possível registo gráfico dos valores convertidos pode ser visto na Figura 3.8, onde registei cinquenta conversões simultâneas.

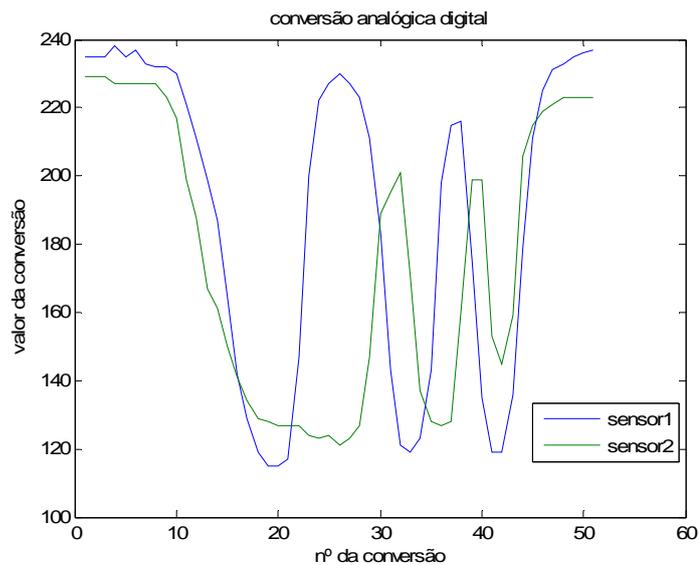


Figura 3.8 – Conversão analógica digital

3.5 Software para o utilizador

Para este equipamento podem ser desenvolvidos vários tipos de software para o utilizador, dependendo da necessidade de cada paciente. Nesse trabalho optei por elaborar uma plataforma que adquire e apresenta as leituras provenientes da mão esquerda e da mão direita. Modificações poderiam ser introduzidas facilmente se se pretendesse medir a flexão/extensão da perna, do braço ou de uma parte do corpo em que se queria estudar a flexão/extensão.

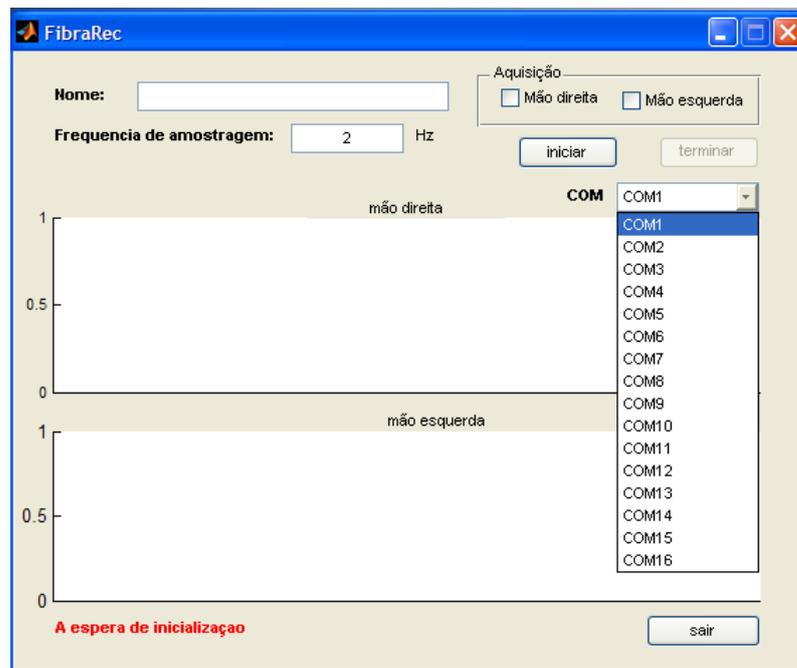


Figura 3.9 – Software para o utilizador

Nesta janela de interface com o utilizador pode-se escolher se a aquisição é da mão direita, da mão esquerda ou de ambas as mãos. Antes de iniciar a aquisição tem que se preencher; o campo “**Nome**” com o nome do sujeito; o campo “**COM**” com a porta a ser usada, que aquisição se quer efectuar e qual a frequência de aquisição dos dados.

Os dados adquiridos são vistos nos gráficos “**mão direita**” e “**mão esquerda**”, dependendo da escolha feita pelo utilizador. Ao mesmo tempo, os dados são guardados num ficheiro.

Para terminar faz-se um clique no botão “**terminar**”, utilizando esta opção o utilizador pode voltar ao exercício sem ter que preencher novamente os campos iniciais. E para sair mesmo do exercício carrega no botão “**sair**” e no botão “**ok**” da caixa de mensagem para confirmar a saída.

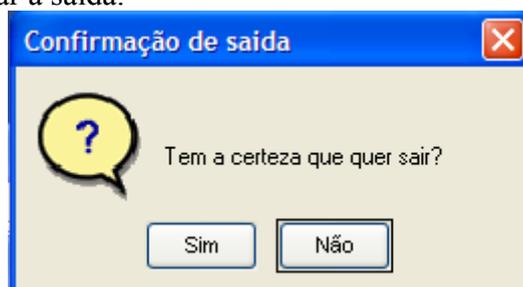


Figura 3.10 – Caixa de mensagem para confirmar a saída

A frequência de amostragem, é usada para calcular o tempo de pausa entre envios de comandos e recepção de valores. Sugiro que seja feita dez amostragens por segundo, pois penso que para sujeitos com deficiências motoras não seja preciso recolher muitos dados para estudar o progresso pois os exercícios são feitos lentamente, permitindo assim não perder grande informação entre amostragens.

Os dados adquiridos são guardados num ficheiro tipo “.txt” com o nome utilizado no campo “Nome” seguido da data em que foi feito exercício. Por exemplo se no campo “Nome” estivesse “Sonia” o ficheiro gerado seria “ Sonia_09-Feb-2008 19-33-55.txt”. O formato do ficheiro é o seguinte:

Freq: <valor de frequência de amostragem>

R L

<valor sensor 1> <valor sensor 2>

<valor sensor 1> <valor sensor 2>

4 Assistente de exercícios

Os conhecimentos que adquiri enquanto aperfeiçoava o equipamento de fibra óptica tinham também como objectivos a sua aplicação no desenvolvimento de um hardware para o controlo de um motor DC. Este motor será utilizado para desenvolver um equipamento capaz de auxiliar um paciente a executar repetidamente exercícios que desenvolvam a sua capacidade de manter o equilíbrio de um membro, quando colocado numa posição instável. O controlo do motor estará dependente da posição do membro, que será avaliada recorrendo a uma webcam e a tratamento de imagem.

Neste capítulo faço uma breve apresentação sobre técnicas de processamento de imagem de modo a que estas possam ser implementadas e dar resposta, em tempo real, as necessidades de um paciente, Posteriormente descrevo o controlo do motor.

4.1 Processamento de imagem

O objectivo do processamento de imagem neste trabalho é o de identificar um objecto na imagem e seguir a sua trajectória. O processamento tem que demorar fracções de segundos para poder dar uma resposta em tempo real as necessidade de um paciente, tem que ser robusto e não ser vulnerável a condições externas de iluminação e a possíveis parecenças entre a cor do marcador e o ambiente em que este está envolvido.

Para o processamento das imagens foram utilizados o software Matlab e a Toolbox de imagem.

4.1.1 Noções básicas de processamento de imagem

Uma **imagem monocromática** é uma função da intensidade da luz, $f(x,y)$, onde x e y são as coordenadas espaciais e o valor de f em (x,y) é proporcional ao brilho da imagem nesse ponto. Se tivermos uma imagem multicolor f é um vector que indica a correspondente banda de cor.

Uma **imagem digital** é uma imagem $f(x,y)$ que foi tornada discreta não só em coordenadas espaciais mas também em intensidade de brilho. É representado por uma matriz de inteiros a duas dimensões, ou por uma série de matrizes, uma para cada banda de cor. Cada elemento da matriz é chamado de **pixel**. O valor do brilho é conhecido como **nível de cinzento**.

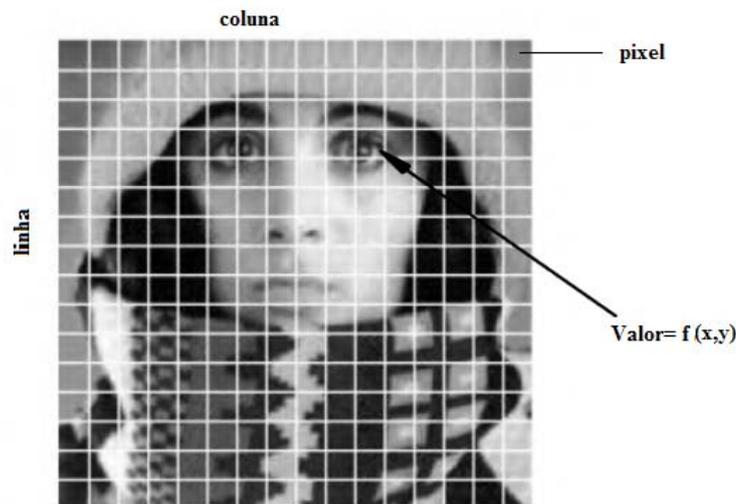


Figura 4.1 – Uma imagem digital. A imagem monocromática foi dividida por N linhas e M colunas, cada intersecção é um pixel e representa o nível de cinzento da imagem nesse ponto.

O nível de cinzento é muitas vezes representado com uma resolução de 8 ou 16 bits ou então por valores decimais compreendidos entre 0 e 1.

Uma **imagem binária** é uma imagem em que os valores associados a cada pixel só podem ser 0 ou 1.



Figura 4.2 – Imagem binarizada

As imagens a cores são definidas a partir de uma série de 3 matrizes com valores compreendidos entre 0 e 1. Existem vários modelos de imagens a cores, entre os quais destaco o **modelo RGB**, o **modelo YUV** e o **modelo HSV** pelas suas potencialidades para este trabalho.

O modelo RGB (red, green, blue) é o sistema base do olho humano. O modelo HSV (Hue, Saturation, Value) é o mais apropriado para segmentação da cor. O modelo YUV (luminância, blue-yellow, red-green) é utilizado na aquisição de vídeo.

O processamento de imagem é feito utilizando transformações de imagem, que por sua vez são executados utilizando operadores. Um **operador** aplica numa imagem inicial uma transformação e gera uma outra imagem.

Para identificar objectos numa imagem pode-se utilizar a técnica de **deteção de arestas** ou a **segmentação da cor**. Neste presente trabalho abordo apenas a segmentação da cor pois as imagens processadas no âmbito do projecto são imagens a cores.

A segmentação utiliza um filtro ou máscara para binarizar uma imagem, passando a um as zonas em que a cor for detectada e as restantes zonas permanecem a zero, como se pode ver na Figura 4.3.



Figura 4.3 – A imagem original foi segmentada para se detectar os objectos cor de laranja

Uma imagem mesmo depois de ser segmentada pode conter muitos ruídos, devendo-se então aplicar **operações morfológicas** para torná-la mais nítida. Estas operações ajudam a isolar objectos com buracos, a preencher buracos em objectos, e a separar objectos que se tocam, entre outros.

4.1.2 Aplicação do processamento de imagem ao trabalho

Para poder detectar o marcador e seguir a sua trajectória tive que definir os seguintes passos:

- Capturar uma imagem;
- Segmentar a imagem;
- Eliminar ruídos da imagem;
- Identificar o centro de massa do marcador;
- Registrar o valor do centro de massa.

As imagens que utilizei foram captadas por uma câmara, no formato YUV e com uma resolução de 320×240. As imagens no formato YUV têm a informação contida em três matrizes de difícil compreensão. Sendo assim tive que convertê-las para o formato RGB, que embora perceptível para o olho humano, mantém a mesma informação conjunto de três matrizes de cores.

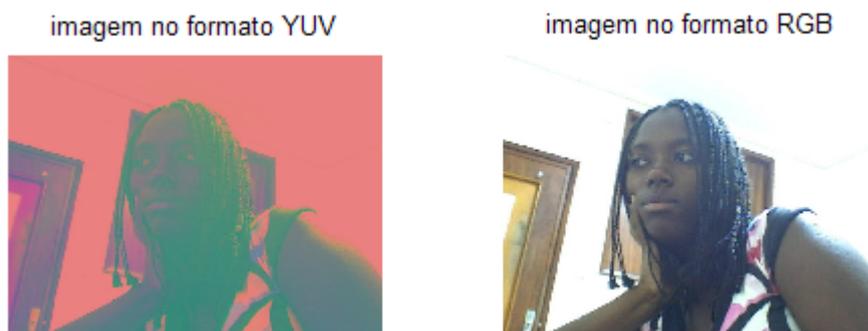


Figura 4.4 – Imagem no formato YUV (esquerda), mesma imagem no formato RGB (direita)

É difícil identificar uma determinada cor quando a informação de cada pixel está contida em três matrizes de cores. Sendo assim tive que converter as imagens do formato RGB para o formato HSV. Embora as imagens continuem a conter a informação em três matrizes agora cada matriz representa uma propriedade distinta. A matriz H representa o tom da cor, a matriz S a saturação da cor e a matriz V a intensidade da cor.

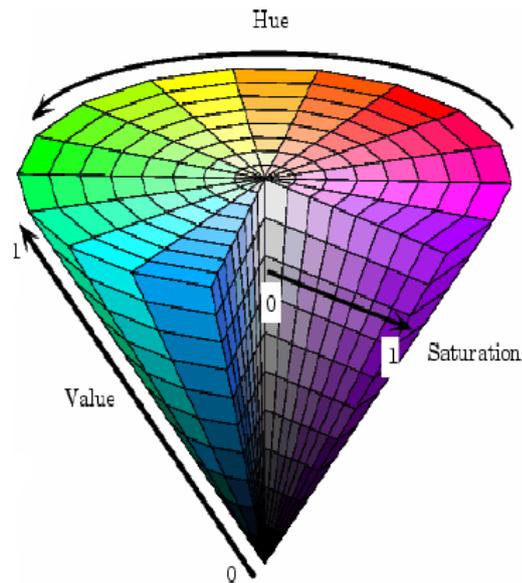


Figura 4.5 – Cone que demonstra com funciona o formato HSV

Tendo três matrizes distintas torna-se possível a manipulação duma imagem. Por exemplo se quiser detectar um objecto laranja preciso apenas de aplicar uma máscara do laranja e a segmentação é feita ficando apenas o objecto laranja identificado na imagem como se pode ver na Figura 4.3.

Utilizando o formato HSV resolvo não só o problema da cor, mas também o da iluminação e da existência de cores próximas. A iluminação quando inadequada condiciona a correcta extracção de informação. Indo ao encontro do objectivo deste equipamento em que não se pode controlar o ambiente em que o mesmo será utilizado, tive que definir uma estratégia para que conseguisse detectar o marcador independentemente da iluminação do local. Para isso tive que manipular a matriz S tornando assim a cor muito saturada, permitindo detectar sempre o marcador.

Como também não posso controlar a cor da roupa que uma pessoa usa, manipulei a matriz V para que apenas uma cor com uma determinada intensidade de brilho fosse detectada.

Juntando as manipulações feitas a cada matriz construí a máscara ou filtro para segmentação das imagens capturadas, como se pode verificar *Apêndice* .

Tendo a imagem segmentada, tive que eliminar o ruído e identificar o centro de massa do sistema emissor de luz composto por LEDs. Contudo um novo problema surge pois as operações morfológicas são muito pesadas não permitindo uma resposta em tempo real. Utilizando operações morfológicas conseguia processar 5 imagens em dez segundos, então abdiquei destas operações em prol de instruções do Matlab como o `meshgrid` que me permitiram alcançar o resultado pretendido e processar 5 imagens por segundo.

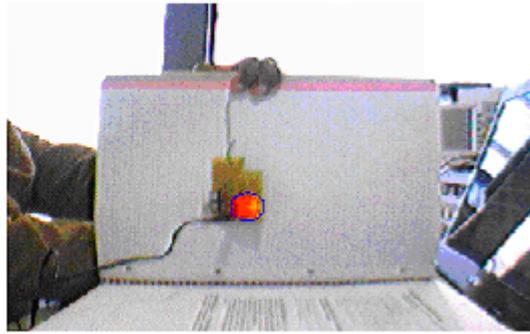


Figura 4.6 – Imagem do marcador (conjunto de nove leds).

4.2 Controlo do motor

Para controlar o motor teve que dar especial atenção a dois pontos cruciais para o seu funcionamento: a sua velocidade do motor e o sentido da rotação.

A velocidade dum motor DC é proporcional a tensão que lhe é aplicada. Porém pode também depender do peso que o motor suporta, o que não é desejável. Para conseguir controlar o motor para que isto não suceda, utiliza-se a técnica de PWM (pulse width modulation).

No circuito implementado, o motor é controlado através de um circuito integrado LM6203. Este circuito por sua vez é controlado por um computador que comunica com o PIC definindo a velocidade e o sentido de rotação.

4.2.1 Modulação do PWM

Para modular um PWM é preciso definir o **período** e o **duty-cycle**. O duty-cycle é a percentagem da razão entre o tempo em que o sinal está em 1 (valor lógico) e o período do ciclo. Por exemplo, para se ter um duty-cycle de 50% o PWM tem que estar metade do período a 1 (estado alto) e metade a 0 (estado baixo).

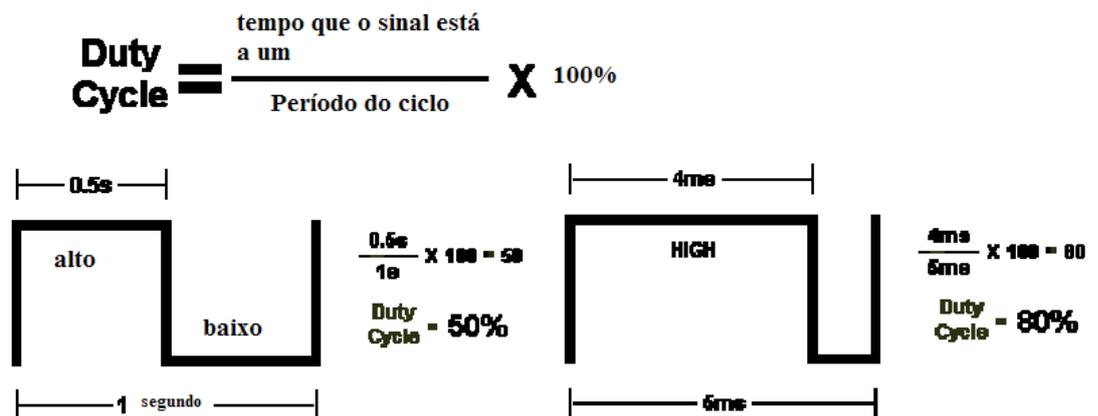


Figura 4.7 – Representação de um ciclo do PWM

O PIC utilizado é equipado com um módulo CCP (Capture/Compare/PWM) que pode ser configurado para produzir um sinal PWM com 10 bits de resolução. O período do PWM e a frequência de oscilação são determinados pelo Timer2 (TMR2). O período pretendido é especificado no registo PR2 ().

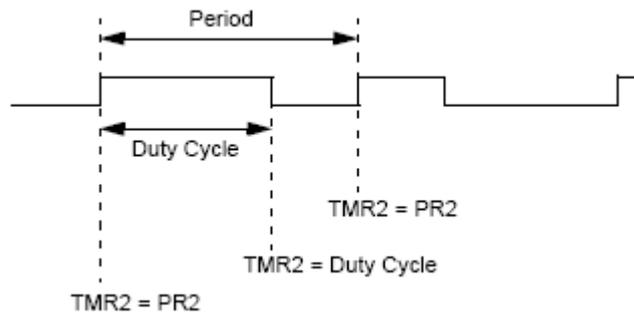


Figura 4.8 – Controlo do PWM

O duty-cycle tem 10 bits de resolução que são especificados escrevendo no registo CCPRxL os 8 bits mais significativos e nos bits quatro e cinco do registo CCPxCON os 2 bits menos significativos. Embora a resolução do duty-cycle seja de 10 bits (0 a 1023), estou limitada pela comunicação por porta série que envia/recebe dados de 8 bits (0 a 255). Assim, converto valores que variam de 0 a 255 para valores que variam de 0 a 1023, ou seja, multiplico por quatro o valor recebido.

Assim sendo tenho um duty-cycle variável dependendo do valor que o PIC receber e uma frequência do PWM de 3KHz.

4.2.2 Hardware do controlo

Como tinha referido o motor é controlado por um circuito integrado que por sua vez é controlado por um PIC dependendo da informação que esse receber do computador, o que me levou a dividir o circuito em dois blocos: o bloco do controlo do motor e o bloco do PIC (comunicação por porta série, comunicação com o LM6203).

O bloco do PIC tem a mesma composição do bloco apresentado na secção 3.1, mudando apenas as funcionalidades do PIC. Neste bloco específico o PIC é responsável pelos parâmetros que o LM6203 recebe (velocidade, sentido e estado on/off), e da comunicação por porta série.

A velocidade é dada pelo PWM gerado no pino multiplexado RC2, o sentido é definido pelo estado do pino RB3 e o enable pelo pino RA2.

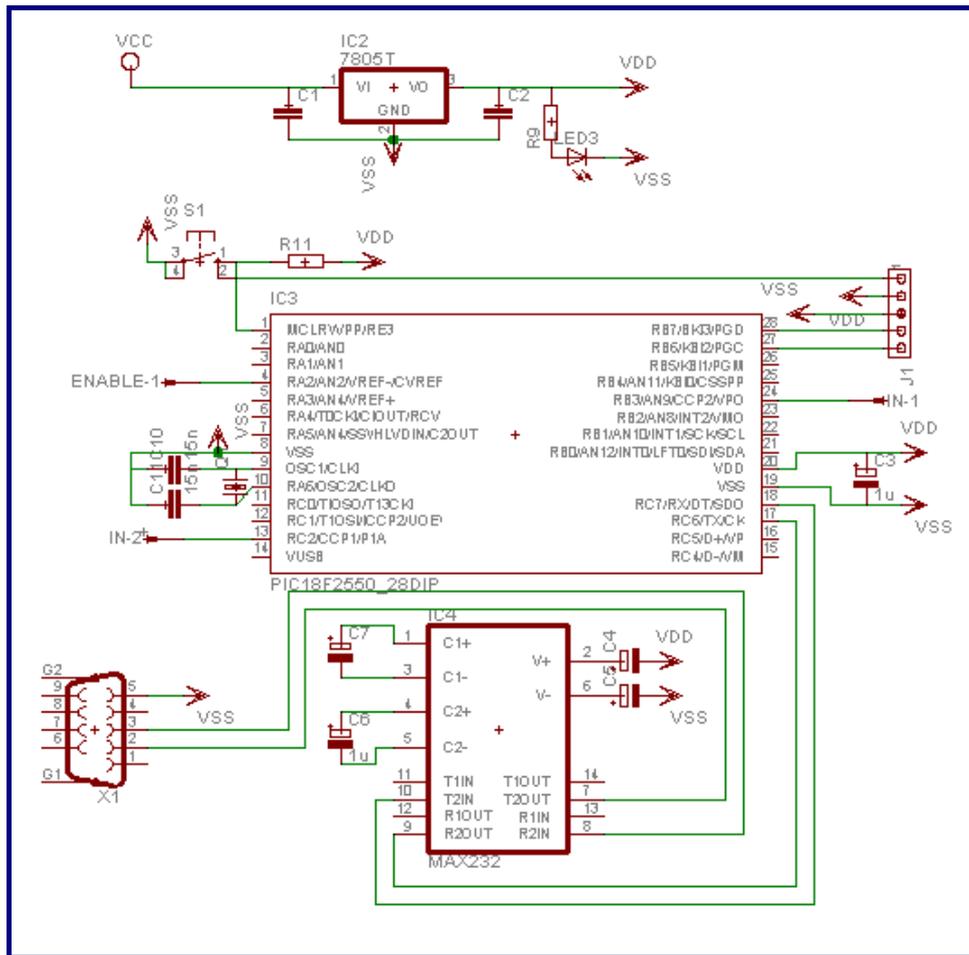


Figura 4.9 – Esquemático do bloco do PIC.

O bloco do controlo do motor é composto por um motor DC de 12 volts da série 942 D da MFA COMO DRILLS, um LM6203 e dois 72LS00 (circuito de portas analógicas).

Um LM6203 é composto por uma ponte-H, onde cada metade é controlada por um input lógico, enquanto um único enable controla ambas as metades da ponte. Quando o valor lógico do enable for zero ou os valores lógicos das entradas forem iguais o motor não trabalha.

O sentido de rotação é definido pelo valor lógico das entradas (IN1, IN2). Por exemplo se a entrada IN1 estiver a um e a entrada IN2 a zero o motor roda no sentido ‘para a esquerda’, caso contrário roda no sentido ‘para a direita’.

Como é o PIC que controla a velocidade, o sentido e o enable, foi preciso arranjar um circuito que permutasse as entradas dependendo do sentido de rotação do motor desejado, e essa é a função das portas NAND no circuito (Figura 4.9).

O motor é um sistema indutivo. Para evitar possíveis picos de corrente da força contra-electromotriz utilizei dois diodos nas saídas (OUT1, OUT2) do LM6203.

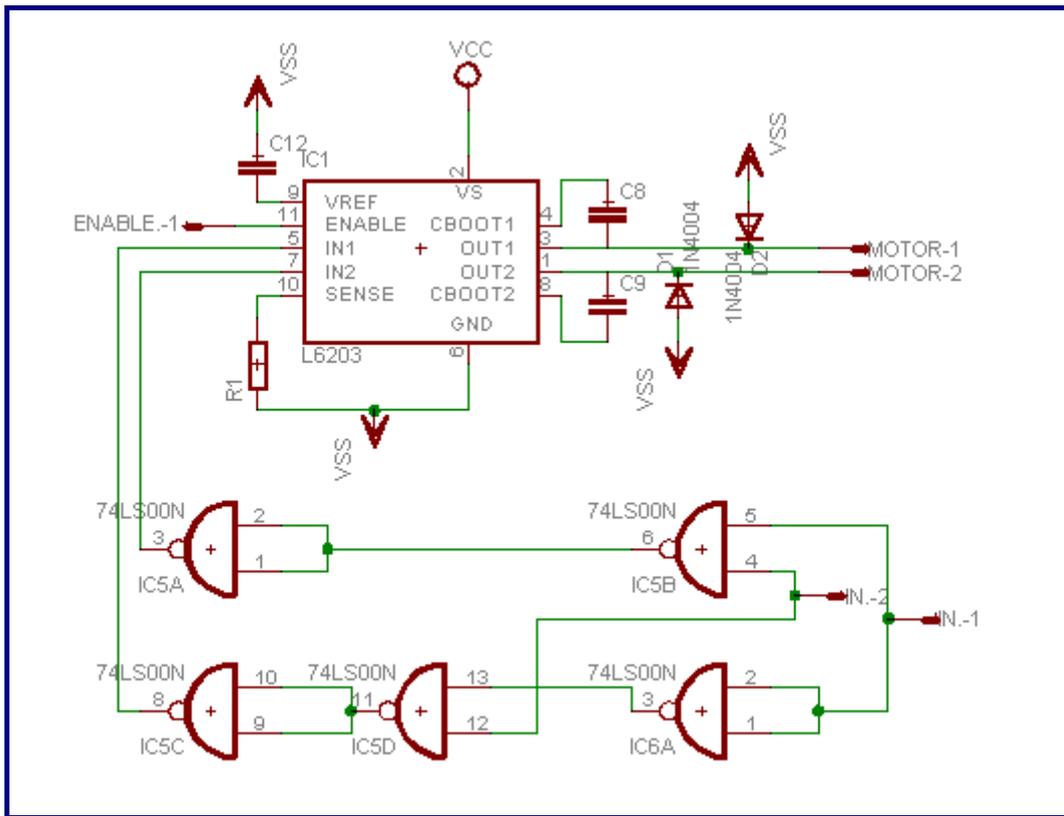


Figura 4.10 – Esquemático do bloco do controlo do motor

Resolvidas todas as questões técnicas e testado o circuito na placa experimental, fiz uma placa de circuito impresso como se pode ver na Figura 4.11. O assistente de exercício é composto por uma câmara, uma placa para controlo do motor e comunicação com um computador, um marcador (Figura 4.6) e uma câmara.



Figura 4.11- Equipamento do Assistente de exercício

4.2.3 Protocolo de comunicação

As características da comunicação entre o computador e o PIC são as mesmas das que foram definidas na secção 3.3.

O protocolo que defini para controlar o motor foi o seguinte:

- Para definir a velocidade envio dois caracteres, primeiro um 'V' para notificar o PIC que o próximo carácter a ser enviado vai definir o duty-cycle (char (0-255));
- Para mover o motor para a direita envio o carácter 'R' ou 'r';
- Para mover o motor para a esquerda envio o carácter 'L' ou 'l';
- Para parar o motor envio o carácter 'S' ou 's'.

4.3 Exercício proposto

Para este equipamento imaginei um exercício que seria fácil de ser executado. O exercício consiste em colocar um marcador no joelho de um paciente e seguir a sua trajetória enquanto o paciente balança a perna entre uma posição inicial, ϕ_0 , e uma posição final, ϕ_{\max} .

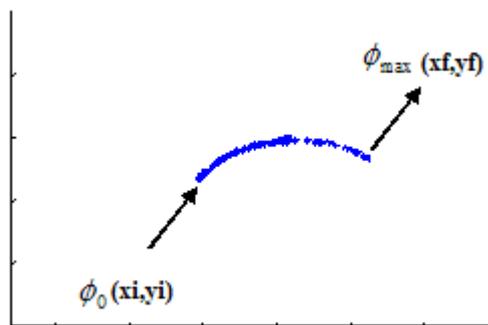


Figura 4.12 – Limites da trajetória do exercício para o paciente

Nesse exercício cada imagem captada passa por dois testes: o teste da existência do objecto e o teste dos limites.

No teste de existência do objecto averigua-se a existência do marcador na imagem a ser processada. Caso não seja encontrado o computador envia uma ordem para o PIC para accionar o motor e para repor a perna da pessoa na posição inicial.

No teste dos limites averigua-se a trajetória do marcador, se o “centro de massa” do marcador estiver fora dos limites predefinidos então o motor é accionado para colocar o marcador dentro dos limites da trajetória. Este teste tem a particularidade de estabelecer o sentido de rotação do motor: quando a posição do “centro de massa” for inferior à posição inicial o carácter a ser enviado é um 'R' e quando for superior a posição final será um 'L'.

Se a imagem passar estes dois teste o paciente fica a fazer o exercício até ao momento que quiser parar ou o tempo predefinido terminar. A imagem captura é sujeita continuamente a estes testes durante todo o exercício.

Os resultados do exercício são guardados para uma análise posterior.

5 Resultados e Perspectivas futuras

Neste trabalho foi desenvolvido equipamentos que possam vir a ser úteis na área da medicina de reabilitação. O desenvolvimento destes instrumentos constitui um passo importante que procura promover a investigação do progresso de reabilitação física utilizando técnicas variadas.

Este trabalho tinha como objectivo desenvolver um equipamento e metodologias de análise de dados que pudessem ser utilizados no progresso da reabilitação de pessoas com deficiência motora. Para que isso fosse possível partiu-se duma ideia abstracta que consistia em desenvolver um equipamento que pudesse acompanhar o exercício de uma pessoa mesmo sem a presença permanente dum clínico. Dai se optou por desenvolver um equipamento que usufruísse de informações captadas por uma câmara para ajudar um paciente a realizar o seu exercício de reabilitação.

Foram desenvolvidos dois equipamentos e estudou-se a possibilidade do método do PCA ser uma metodologia de análise clínica. Os objectivos foram alcançados, contudo existe espaço para melhorias tanto nos equipamentos como no método do PCA. O Sensor de Curvatura pode ser aplicado em mais situações clínicas e acoplado a outros equipamentos como o Lokomat para o estudo da flexão/extensão dos membros inferiores, não se restringir a ser utilizado apenas no estudo dos membros superiores. No Assistente de Exercícios pode-se otimizar o processamento de imagem para que este processe mais frames por segundo permitindo desenvolver exercícios mais complexos que vão de encontro as necessidades dum paciente em particular e não o actual exercício que depende apenas das condições limites do exercício. O método de análise dos dados pode ser explorado permitindo descobrir outras variabilidades para além do padrão do movimento, talvez da disfunção motora que o sujeito sofre.

Futuramente pensa-se em trabalhar na segurança destes equipamentos para que possam ser testados num ambiente clínico, desenvolver aplicações para os mesmos tendo em conta as necessidades dum deficiente motor. Para o método do PCA pensa-se em recolher mais dados para verificar se a ideia aqui desenvolvida está correcta e posteriormente arranjar um modelo matemático que possa explicar esses comportamentos nos movimentos.

Apêndice A

Este apêndice contém as funções desenvolvidas para o método PCA

Média móvel

```
function [xm ym]=MediaMovel(x, y, d)
% calcula a media movel com um intervalo d
for j=1: numel(x)-d
    xm(j,1)= sum(x(j:j+d-1))/d;
    ym(j,1)= sum(y(j:j+d-1))/d;
end
xm=xm;
ym=ym;
```

Velocidade e sua norma

```
function [dxm dym vd]=Velocidade(xm, ym, t)
% calcula a variacao dxm e dym e a norma
% t--> numero de pontos a desprezar
for i=1: numel(xm)-t
    dxm(i)=xm(i+t)-xm(i-1+t);
    dym(i)=ym(i+t)-ym(i-1+t);
    v1=[dxm(i) dym(i)];
    vd(i)= norm(v1);
end
```

Ângulo

```
function [theta]=Angulo(DX, DY)
% calcula o angulo entr duas normas consecutivas
for i=1: numel(DX)-1
    v1=[DX(i) DY(i) 0];
    v2=[DX(i+1) DY(i+1) 0];
    if (norm(v1)*norm(v2)~=0)
        d = cross(v1,v2);
        theta(i)= real(asind(d(3)/(norm(v1)*norm(v2))));
    else
        continue
    end
end
```

Método do PCA

```
function [FV, lamb, Sa]=PCA(A)
% aplica o pca, tem uma normalização de
% media zero e variancia um
Sa=corr(A);
[eigV lamb]=eig(Sa);
l1=eig(Sa);
FV=eigV(:,end:-1:1);
```

Apêndice B

Este apêndice apresenta os programas feitos para o Sensor de Curvatura.

Conversão analógico/digital

O programa da conversão, quando o valor da conversão for maior que 128 acende o led, dando o sinal que a conversão está sendo feita. A linguagem utilizada foi o C.

```
#include<p18f2550.h>

void init(void);

void main_(void){
    // declaração de variaveis
    unsigned int perce;

    init();
    PORTCbits.RC2=0;
    // conversão
    for(;;){
        ADCON0=0b00000101;
        ADCON0bits.GC=1;
        while (ADCON0bits.GC==1);
        perce=ADRESH;
        if(ADRESH>128) PORTCbits.RC2=1;
        else PORTCbits.RC2=0;
    }
}

void init(void){
    //INICIALIZAÇÃO DOS PORT
    PORTA=0;
    LATA=0;
    PORTB=0;
    LATB=0;
    PORTC=0;
    LATC=0;

    // INICIALIZAÇÃO DOS PINS
    TRISA=0b00000011; //duas portas de entrada (analógicas)
    TRISB=0b00000000;
    TRISC=0b00000000;

    //definir os registos do conversor analogico digital
    ADCON1=0b00001101; // duas portas analogicas e as outras digitais
    ADCON2=0b00001100; // frequencia de Fosc/4
}
```

Comunicação por porta série

O programa do PIC para a comunicação por porta série. Quando o PIC estiver a transmitir um led fica a acender e a apagar indicando que está a decorrer uma transmissão.

```
#include<p18f2550.h>

void init(void);

void main(void){
    // declaração de variaveis
    unsigned char reciv;
    init();
    PIR1bits.RCIF=0;
    PORTCbits.RC2=0;

    // conversão
    for(;;){
        reciv = 0;
        if (PIR1bits.RCIF==1){
            reciv = RCREG;
            if (reciv == 'L') ADCON0=0b00000001;
            else if (reciv == 'R') ADCON0=0b000000101;
            if (reciv=='L' || reciv=='R') {
                ADCON0bits.GC=1;
                while (ADCON0bits.GC==1);
                TXREG = ADRESH;
                PORTCbits.RC2 = !PORTCbits.RC2;
            }
            PIR1bits.RCIF=0;
        }
    }
}
```

```
void init(void){
  //INICIALIZAÇÃO DOS PORT
  PORTA=0;
  LATA=0;
  PORTB=0;
  LATB=0;
  PORTC=0;
  LATC=0;

  // INICIALIZAÇÃO DOS PINS
  TRISA=0b00000011; //duas portas de entrada
  TRISB=0b00000000; // todas saidas
  TRISC=0b10000000; //uma entrada e o resto saida

  //definir os registos do conversor analogico digital
  ADCON1=0b00001101; // duas portas analogicas e as outras digitais
  ADCON2=0b00001100;

  //definir os registos da porta série com um baud rate 9600
  TXSTA = 0b00100000; // e uma velocidade baixa no modo assincrono
  RCSTA = 0b10010000;
  SPBRG = 77; //define velocidade de comunicação.
           //77 -> 9600 bits/s com Fosc 48 MHz
  PIE1bits.RCIE=1; // activar o interrupt da porta serie
}
```

Captura de dados

Este código permite capturar valores convertidos e guardá-los num ficheiro tipo .txt.

```
% definir a porta serie a utilizar
s=serial('COM3', 'BaudRate',9600,'DataBits',8);
fopen(s) % abrir a porta serie
fidR = fopen('dadosR.txt','w'); %abrir o ficheiro onde se vai guardar
fidL = fopen('dadosL.txt','w'); % os dados
bufferL = zeros(1,1,'uint8');
bufferR = zeros(1,1,'uint8');

% ciclo de recepcao e transmissao de dados
for k=1:50,
    % dedeira direita
    fprintf(s,'R');
    bufferR(k) = fread(s,1,'uint8');
    if isempty(bufferR(k)) break; end;% se estiver vazio sai do ciclo
    fprintf(fidR,'%d\r\n',bufferR(k));
    % dedeira esquerda
    fprintf(s,'L');
    bufferL(k) = fread(s,1,'uint8');
    if isempty(bufferL(k)) break; end;
    fprintf(fidL,'%d\r\n',bufferL(k));
end

% fechar os ficheiros
fclose(fidR);
fclose(fidL);
% ver os dados
figure(1); plot(bufferL); title('dedeira esquerda'); ylim([0 255])
figure(2); plot(bufferR); title('dedeira direita'); ylim([0 255])

% fechar a porta serie
fclose(s)
delete(s)
clear s
```

Apêndice C

Este apêndice contém os programas desenvolvidos para o Assistente de Exercício.

Modulação do PWM e comunicação por porta série

Este programa utiliza a funcionalidade do PIC para gerar PWM

```
#include<p18f2455.h>

unsigned int dutycycle = 50;
unsigned char recib=0, ant_reciv=0;

void init(void);
void rx_handler (void);
#pragma code rx_interrupt = 0x8
void rx_int (void)
{
    _asm goto rx_handler _endasm
}
#pragma code

#pragma interrupt rx_handler
void rx_handler (void)
{

    /* Get the character received from the USART */
    recib = RCREG;
    TXREG=recib;
    if (ant_reciv=='V') {
        T2CONbits.TMR2ON = 0; // Turn off Timer2
        dutycycle = recib;
        dutycycle = dutycycle<<2; //dutycycle = dutycycle * 4
        CCP1IL = (unsigned char)((dutycycle>>2) & 0xFF);
        CCP1CON = (CCP1CON & 0xCF) | ((dutycycle << 4) & 0x30);

        T2CONbits.TMR2ON = 1; // Turn on Timer2
        recib = 0;
    } else switch (recib) {
        case 'S':
        case 's':
            T2CONbits.TMR2ON = 0; // Turn off Timer2
            PORTAbits.RA2=0;
```

```
        break;
    case 'L':
    case 'l':
        PORTAbits.RA2=1;
        PORTBbits.RB3=1;
        T2CONbits.TMR2ON = 1; // Turn on Timer2
        break;
    case 'I':
    case 'i':
        TMR2 = 0;
        dutycycle = dutycycle + 10;
        if (dutycycle>128) dutycycle = 128;
        CCP1L = (unsigned char)((dutycycle>>2) & 0xFF);
        CCP1CON = (CCP1CON & 0xCF) | ((dutycycle << 4) & 0x30);
        break;
    case 'D':
    case 'd':
        TMR2 = 0;
        dutycycle = dutycycle - 10;
        if (dutycycle<10) dutycycle = 10;
        CCP1L = (unsigned char)((dutycycle>>2) & 0xFF);
        CCP1CON = (CCP1CON & 0xCF) | ((dutycycle << 4) & 0x30);
        break;
}
ant_reciv = reciv;

/* Clear the interrupt flag */
PIR1bits.RCIF = 0;
}

void main(void){
    // declaração de variaveis
    unsigned char mensagem;
    init();
    mensagem='U';
    TXREG=mensagem;
    while(1)
        ;
}

// fim da main

void init(void){
    //INICIALIZAÇÃO DOS PORT
    PORTA=0;
    LATA=0;
    PORTB=0;
    LATB=0;
    PORTC=0;
    LATC=0;

    // INICIALIZAÇÃO DOS PINS
    TRISA=0b00000001;
    TRISB=0b00100000;
    TRISC=0b10000000;
```

```
//PWM1
CCP1CON |= 0b00001100;
T2CONbits.TMR2ON = 0; // STOP TIMER2 registers to POR state
PR2 = 0xff; // Set period

T2CON = 0b00000011; // Set all configuration values,
//but don't start timer yet^^ => Prescaler 1/16
TMR2 = 0; // Clear Timer2
PIR1bits.TMR2IF = 0;

CCPR1L = (unsigned char)((dutyycle>>2) & 0xFF);
CCP1CON = (CCP1CON & 0xCF) | ((dutyycle << 4) & 0x30);

//definir os registos da porta série
SPBRG = 77; //define velocidade de comunicação.
TKSTA = 0b00100000;
RCSTA = 0b10010000;

PIE1bits.RCIE=1; // activar o interrupt da porta serie

RCONbits.IPEN = 1;
IPR1bits.RCIP = 1;
INTCONbits.GIEH = 1;
}
```

Processamento de imagem

```

clc
clear all
close all
%%
% captura do video
info=imqhwinfo('winvideo');
vid=videoinput('winvideo',1,'UYVY_320x240');
preview(vid)
pause(5)

% definição do tamanho da imagem
largura=320;
altura=240;

%% dados pro circulo:)
radius=10;
[x y]=meshgrid(1:largura,1:altura);

% aquisição de imagem e processamento
while(1)
    image_yuv=getsnapshot(vid);
    image_rgb=ycbcr2rgb(image_yuv);
    B=rgb2hsv(image_rgb);
    masc=((B(:,:,1)<0.13)|(B(:,:,1)>0.85))...
        &(B(:,:,3)>0.8)&(B(:,:,2)>0.8));
    masc=medfilt2(masc);
%% determinação do centro de massa
    num_pontos = sum(sum(masc>0)); %numeros de ls
    if num_pontos==0,
        fprintf('Nenhum objecto detectado\n');
        continue;
    end
    ImR = image_rgb(:,:,1);
    ImG = image_rgb(:,:,2);
    ImB = image_rgb(:,:,3);

    Xm = sum(sum(x.*masc))/num_pontos;% posição centro de massa X
    Ym = sum(sum(y.*masc))/num_pontos;% posição centro de massa Y
% visualização do centro de massa todos os pontos:(
    figure(2)
    plot(Xm,-Ym,'*'); hold on
    axis([0 330 -250 0])
    drawnow

```

```
%% desenho do circulo(polar)
for teta=0:0.06:2*pi,
    cirX = round(Xm + radius*cos(teta));
    if cirX<1 | cirX>largura, continue, end
    cirY = round(Ym + radius*sin(teta));
    if cirY<1 | cirY>altura, continue, end

    ImR(cirY, cirX) = 0;
    ImG(cirY, cirX) = 0;
    ImB(cirY, cirX) = 255;
end

image_rgb(:, :, 1) = ImR;
image_rgb(:, :, 2) = ImG;
image_rgb(:, :, 3) = ImB;
figure(1)
imshow(image_rgb)
end

figure(1)
imshow(image_rgb)
closepreview(vid)
```

Apêndice D

Foi efectuado um estudo a 59 vinhos diferentes, onde se analisaram 13 características distintas. Estes resultados foram agrupados numa tabela.

Alco	MaAc	Ash	A_AI	Mag	TotP	Flav	NonP	Proa	Coll	Hue	OD	Prol
14,23	1,71	2,43	15,6	127	2,8	3,06	0,28	2,29	5,64	1,04	3,92	1065
13,2	1,78	2,14	11,2	100	2,65	2,76	0,26	1,28	4,38	1,05	3,4	1050
13,16	2,36	2,67	18,6	101	2,8	3,24	0,3	2,81	5,68	1,03	3,17	1185
14,37	1,95	2,5	16,8	113	3,85	3,49	0,24	2,18	7,8	0,86	3,45	1480
13,24	2,59	2,87	21	118	2,8	2,69	0,39	1,82	4,32	1,04	2,93	735
14,2	1,76	2,45	15,2	112	3,27	3,39	0,34	1,97	6,75	1,05	2,85	1450
14,39	1,87	2,45	14,6	96	2,5	2,52	0,3	1,98	5,25	1,02	3,58	1290
14,06	2,15	2,61	17,6	121	2,6	2,51	0,31	1,25	5,05	1,06	3,58	1295
14,83	1,64	2,17	14	97	2,8	2,98	0,29	1,28	5,2	1,08	2,85	1045
13,86	1,35	2,27	16	98	2,98	3,15	0,22	1,85	7,22	1,01	3,55	1045
14,1	2,16	2,3	18	105	2,95	3,32	0,22	2,38	5,75	1,25	3,17	1510
14,12	1,48	2,32	16,8	95	2,2	2,43	0,26	1,57	5	1,17	2,82	1280
13,75	1,73	2,41	16	89	2,6	2,76	0,29	1,81	5,6	1,15	2,9	1320
14,75	1,73	2,39	11,4	91	3,1	3,69	0,43	2,81	5,4	1,24	2,73	1150
14,38	1,87	2,38	12	102	3,3	3,64	0,29	2,96	7,5	1,2	3	1547
13,63	1,81	2,7	17,2	112	2,85	2,91	0,3	1,46	7,3	1,28	2,88	1310
14,3	1,92	2,72	20	120	2,8	3,14	0,33	1,97	6,2	1,07	2,65	1280
13,83	1,57	2,62	20	115	2,95	3,4	0,4	1,72	6,6	1,13	2,57	1130
14,19	1,59	2,48	16,5	108	3,3	3,93	0,32	1,86	8,7	1,23	2,82	1680
12,37	0,94	1,36	10,6	88	1,98	0,57	0,28	0,42	1,95	1,05	1,82	520
12,33	1,1	2,28	16	101	2,05	1,09	0,63	0,41	3,27	1,25	1,67	680
12,64	1,36	2,02	16,8	100	2,02	1,41	0,53	0,62	5,75	0,98	1,59	450
13,67	1,25	1,92	18	94	2,1	1,79	0,32	0,73	3,8	1,23	2,46	630
12,37	1,13	2,16	19	87	3,5	3,1	0,19	1,87	4,45	1,22	2,87	420
12,17	1,45	2,53	19	104	1,89	1,75	0,45	1,03	2,95	1,45	2,23	355
12,37	1,21	2,56	18,1	98	2,42	2,65	0,37	2,08	4,6	1,19	2,3	678
13,11	1,01	1,7	15	78	2,98	3,18	0,26	2,28	5,3	1,12	3,18	502
12,37	1,17	1,92	19,6	78	2,11	2	0,27	1,04	4,68	1,12	3,48	510
13,34	0,94	2,36	17	110	2,53	1,3	0,55	0,42	3,17	1,02	1,93	750
12,21	1,19	1,75	16,8	151	1,85	1,28	0,14	2,5	2,85	1,28	3,07	718
12,29	1,61	2,21	20,4	103	1,1	1,02	0,37	1,46	3,05	0,91	1,82	870
13,86	1,51	2,67	25	86	2,95	2,86	0,21	1,87	3,38	1,36	3,16	410
13,49	1,66	2,24	24	87	1,88	1,84	0,27	1,03	3,74	0,98	2,78	472
12,99	1,677	2,6	30	139	3,3	2,89	0,21	1,96	3,35	1,31	3,5	985
11,96	1,06	2,3	21	101	3,38	2,14	0,13	1,65	3,21	0,99	3,13	886
11,66	1,88	1,92	16	97	1,61	1,57	0,34	1,15	3,8	1,23	2,14	428
13,03	0,9	1,71	16	86	1,95	2,03	0,24	1,46	4,6	1,19	2,48	392
11,84	2,89	2,23	18	112	1,72	1,32	0,43	0,95	2,65	0,96	2,52	500
12,33	0,99	1,95	14,8	136	1,9	1,85	0,35	2,76	3,4	1,06	2,31	750
12,86	1,35	2,32	18	122	1,51	1,25	0,21	0,94	4,1	0,76	1,29	630
12,88	2,99	2,4	20	104	1,3	1,22	0,24	0,83	5,4	0,74	1,42	530
12,81	2,31	2,4	24	98	1,15	1,09	0,27	0,83	5,7	0,66	1,36	560
12,7	3,55	2,36	21,5	106	1,7	1,2	0,17	0,84	5	0,78	1,29	600

Alco	MaAc	Ash	A_AI	Mag	TotP	Flav	NonP	Proa	Coll	Hue	OD	Prol
12,51	1,24	2,25	17,5	85	2	0,58	0,6	1,25	5,45	0,75	1,51	650
12,6	2,46	2,2	18,5	94	1,62	0,66	0,63	0,94	7,1	0,73	1,58	695
12,25	4,72	2,54	21	89	1,38	0,47	0,53	0,8	3,85	0,75	1,27	720
12,53	5,51	2,64	25	96	1,79	0,6	0,63	1,1	5	0,82	1,69	515
13,49	3,59	2,19	19,5	88	1,62	0,48	0,58	0,88	5,7	0,81	1,82	580
12,84	2,96	2,61	24	101	2,32	0,6	0,53	0,81	4,92	0,89	2,15	590
12,93	2,81	2,7	21	96	1,54	0,5	0,53	0,75	4,6	0,77	2,31	600
13,36	2,56	2,35	20	89	1,4	0,5	0,37	0,64	5,6	0,7	2,47	780
13,52	3,17	2,72	23,5	97	1,55	0,52	0,5	0,55	4,35	0,89	2,06	520
13,62	4,95	2,35	20	92	2	0,8	0,47	1,02	4,4	0,91	2,05	550
12,25	3,88	2,2	18,5	112	1,38	0,78	0,29	1,14	8,21	0,65	2	855
13,16	3,57	2,15	21	102	1,5	0,55	0,43	1,3	4	0,6	1,68	830
13,88	5,04	2,23	20	80	0,98	0,34	0,4	0,68	4,9	0,58	1,33	415
12,87	4,61	2,48	21,5	86	1,7	0,65	0,47	0,86	7,65	0,54	1,86	625
13,32	3,24	2,38	21,5	92	1,93	0,76	0,45	1,25	8,42	0,55	1,62	650
13,08	3,9	2,36	21,5	113	1,41	1,39	0,34	1,14	9,4	0,57	1,33	550

Aplicando o método do PCA obtém-se 13 valores próprios associados a 13 vectores próprios.

Subtraindo apenas a média dos dados, trabalha-se com a matriz covariância e extrai-se os seguintes valores próprios:

Valor próprio	Variância (%)
1,2359e+005	99,83
194,35	0,1569
11,422	0,0092
2,3889	0,0019
1,3891	0,0011
0,5147	0,0004
0,2866	0,0002
0,1545	0,00012
0,1365	0,00011
0,0755	6,1e-005
0,0355	2,8e-005
0,0146	1,2e-005
0,0052	4,2e-006

Tabela 5-valor próprio e a percentagem da variabilidade contida em cada vector próprio associado.

Analisando os valores próprios e a variância chega-se a conclusão que com apenas um pc (tabela 6) consegue-se representar 99,83% da variabilidade dos dados. Conclusão essa que diria que toda a variabilidade dos dados esta contida numa variável (Proline). Isso acontece porque o desvio padrão da variável Proline é 351, tendo assim a sua variância um peso considerável face as outras variáveis.

Para resolver este problema da importância de cada variável, faz-se um re-escalamento normalizando os dados originais de modo a terem media zero e variância igual a um. Aplicando esta normalização cada variável passara a ter igual importância em termos de variabilidade.

Sendo assim trabalha-se com a matriz correlação para extrair os valores próprios e seus vectores próprios associados.

Analisando a tabela 7 conclui-se que é preciso utilizar seis pc's para representar 87,78% da variabilidade dos dados.

P.C	1	Desvio padrão
Alcohol	0.0015	0,78959
MalicAcid	-0.0007	1,1611
Ash	0.0003	0,28837
A_Alcalinity	-0.0038	3,6242
Magnesium	0.0138	14,747
Total Phenol	0.0012	0,70616
Flavanoids	0.0021	1,0958
NonPhenols	-0.0001	0,1297
Proanthocyan	0.0011	0,66642
Color Intensity	0.0020	1,6411
Hue	0.0002	0,22841
OD280/OD315	0.0011	0,73267
Proline	0.9999	351,52

Tabela 6- desvio padrão de cada variável e o primeiro PC que contém 99,83% da variabilidade dos dados.

Valor próprio	Variância (%)	Var cumulativa
5,4217	41,705	41,705
2,3281	17,909	59,614
1,3738	10,568	70,182
0,9743	7,4946	77,676
0,80787	6,2144	83,891
0,50556	3,889	87,78
0,43051	3,3116	91,091
0,28115	2,1627	93,254
0,24843	1,911	95,165
0,22723	1,7479	96,913
0,18666	1,4358	98,349
0,16706	1,2851	99,634
0,04761	0,36626	100

Tabela 7 – valores próprios e suas variâncias para dados re-escalonados (media=0 e variância=1).

Referências

- [1] J. Edward Jackson. “A user’s guide to Principal Components”, John USA, 1991
- [2] Young, Ian Theodore; Gerbrands, Jan Jacob; Van Vliet, Lucas Jozef. “Fundamentals of image processing”, 1998
- [3] Petrou, Maria; Bosdogianni, Panagiota; “Image Processing: The fundamentals”
- [4] Hardle, Wolfgang; Simar, Leopold. “Applied Multivariate Statistical Analysis”, 2003.
- [5] Fonseca, Ana Rita. “ Trabalho final”, 2007.
- [6] Cameirão, Mónica da Silva. “ Desenvolvimento de instrumentos e metodologias para terapia e monitorização dos membros superiores ”, 2005.
- [7] Durana, Eduardo. “Development of integrated electrostimulation equipment for medical rehabilitation of spinal cord injurie”, 2007.
- [8] Daffertshofer, Andreas, Lamothe, Claudine J.C., Meijer, Onno G., Beek, Peter J. “PCA in studying coordination and variability: a tutorial”, 2004.
- [9] Smith, Lindsay I. “A tutorial on Principal components Analysis”, 2002.
- [10] Shlens, Jonathon. “A tutorial on Principal components Analysis”, 2005.
- [11] Lee, Chihoon. “Principal Components Analysis”, 2004
- [12] Microchip Technology Inc., PIC18F2455/2550/4455/4550 Data Sheet, 2005.
- [13] Microchip Technology Inc., MPLAB® C18 C COMPILER USER’S GUIDE, 2005.
- [14] Microchip Technology Inc., MPASM™ Assembler, MPLINK™ Object Linker, MPLIB™ Object Librarian User’s Guide, 2005.
- [15] Bickle, Rick. “DC Motor Control Systems for Robot Applications”, 2003.
- [16] MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS Data Sheet, 2002.
- [17] National Semiconductor Inc., LM124/LM224/LM324/LM2902 Low Power Quad Operational Amplifiers Data Sheet, 2004.
- [18]
- [20]
- [21] <http://novomundo3.wordpress.com/2007/12/10/>
- [22] <http://www.pcd.pt/noticias/ver.php?id=6360>