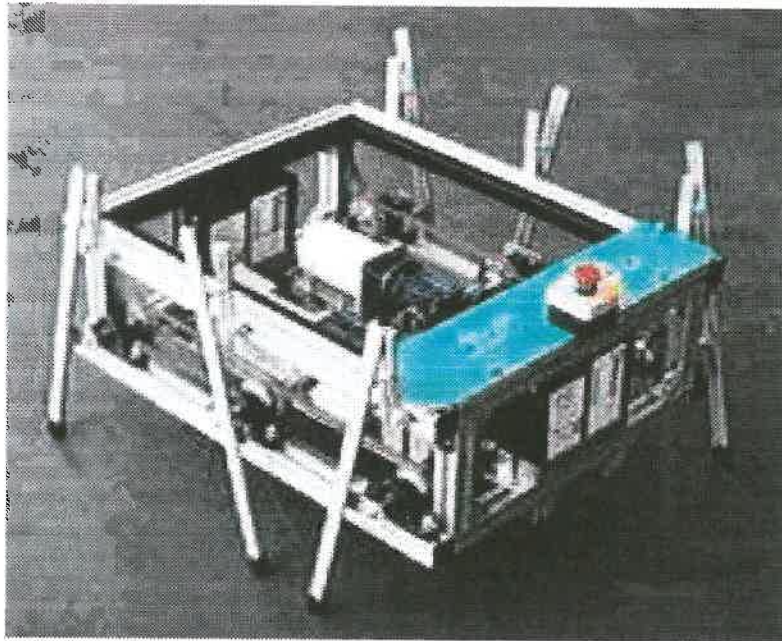




SECÇÃO AUTONOMA DE ENGENHARIA MECÂNICA

Projecto de Automação

Robot com Locomoção Hexápode



Relatório Final

Realizado por: Maria João Abranches
Pedro Viana

Orientadores: Prof. Doutor Vítor Santos
Mestre Filipe Silva



UNIVERSIDADE DE AVEIRO
19 de Setembro de 2000

Agradecimentos

Finalizado este projecto, queremos expressar os nossos agradecimentos a algumas pessoas e instituições que nos apoiaram directa ou indirectamente e que sem elas teria sido mais complicado atingir os objectivos propostos.

- Aos nossos orientadores, Professor Doutor Vitor Santos e Engenheiro Filipe Silva, pelo apoio e sugestões fornecidas ao longo do projecto;
- Ao Engenheiro Abílio Borges por todo o apoio que nos deu, e que mesmo durante o mês de Agosto se mostrou cooperante na conclusão do trabalho;
- A todos os nossos colegas, amigos e professores que nos incentivaram com críticas e sugestões construtivas ao longo deste ano, e em especial ao Rui Santos pela ajuda que sempre nos disponibilizou;
- Aos Srs. Carlos Cordeiro e Alexandre Martins que foram essenciais na compra dos redutores;
- À secção Autónoma de Engenharia Mecânica, que nos acolheu ao longo destes últimos anos;
- A todos os não mencionados neste texto, mas que colaboraram para a concretização deste projecto.

A todos, o nosso muito obrigado.



Índice

	Página
1 - Introdução	3
2 - Objectivos	3
3 - Especificações do robot hexápode	4
4 - Modelação Cinemática	5
4.1 - Definição dos pontos críticos relativos a uma perna	5
4.2 - Definição das equações de movimento para os pontos críticos	6
4.2.1 – Perna no ar	6
4.2.2 – Perna no chão	7
5 - Simulação da locomoção do robot	9
5.1 - Estudo dimensional do robot	10
5.2 - Planeamento de trajectória	12
6 – Estudo dinâmico do robot hexápode	16
6.1 - Estudo das forças e binários envolvidos durante a locomoção	16
6.2 - Cálculo da energia do sistema utilizando o Teorema de Lagrange	17
6.2.1 - Cálculo da energia cinética	18
6.2.2 - Cálculo da energia potencial	20
7 - Estudo dos elementos necessários para a construção do robot	25
8 - Lista de material utilizado	31
9 - Análise dinâmica do robot	32
10 - Máquinas utilizadas	34
11 - Programa autómato	35
12 - Análise do trabalho e conclusões	36
Anexo 1 – Esquemas eléctricos	40
Esquema eléctrico dos motores	41
Esquema eléctrico da unidade de potência	42
Esquema eléctrico das entradas do PLC	43
Esquema de ligações das cartas à unidade de Potência	44
Esquema eléctrico do robot hexápode	45
Anexo 2 – Desenhos de definição	46
Anexo 3 – Programa autómato	58
Anexo 4 – Programa em Matlab	79
Anexo 5 – Documentação	118



1 - Introdução

Este trabalho consiste na concepção, implementação e controlo de um robot com locomoção hexápode, fazendo lembrar um insecto.

Um dos requisitos para a elaboração deste projecto é a utilização de pernas não articuladas, usando um número mínimo de graus de liberdade.

A sincronização adequada das seis pernas, vai originar um conjunto de movimentos que resultam numa translação do robot e possivelmente na rotação em torno de si próprio.

2 - Objectivos

Como objectivos iniciais deste trabalho temos:

- Modelização cinemática e dinâmica;
- Concepção da estrutura mecânica rígida e dos elementos móveis para a movimentação hexapódica, incluindo a transmissão de movimento entre as diversas partes;
- Concepção da componente eléctrica, percepção e selecção de sensores;
- Definição dos motores e unidades de controlo e potência;
- Programa base de controlo de locomoção, para execução dos movimentos básicos com possibilidade de comandos externos;
- Programa de locomoção com alguma autonomia como o desvio de obstáculos.



3 - Especificações do Robot Hexápode

- Sistema com alimentação autónoma (baterias) e controlo autónomo e semi-autónomo (PLC);
- Capacidade de avançar e recuar, e capacidade de rodar sobre si próprio;
- Detectar obstáculos com determinada precisão mínima;
- O sistema deverá poder aceitar teleoperação via linha série RS232 ou sistema similar ou executar um programa pré-definido;
- Para o sistema proposto é suficiente um número finito e reduzido de velocidades de movimentação.



4 - Modelação Cinemática

Como etapa inicial deste projecto procedeu-se à modelação e simulação em MATLAB[®] V5.3 da cinemática do robot hexápode, para se poderem avaliar vários parâmetros dimensionais fundamentais para a concepção da estrutura.

4.1 - Definição dos pontos críticos relativos a uma perna

O movimento do robot é efectuado por intermédio do acoplamento mecânico (rígido) do veio do motor a um elo que vai transmitir movimento à haste da perna. Como na parte superior da perna só há movimento vertical, quando a outra extremidade da haste estiver apoiada numa superfície vai haver uma impulsão de todo o robot na horizontal e também na vertical.

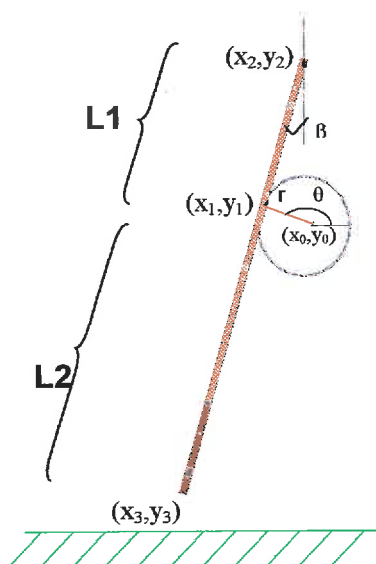


Figura N.º1 – Pontos críticos de uma perna do robot hexápode

Na figura 1 estão representados os quatro pontos críticos. Esses pontos são, o ponto de união da perna ao elo transmissor, $P_1 - (x_1, y_1)$, o ponto de guiamento superior da haste da perna, $P_2 - (x_2, y_2)$, a extremidade da mesma que está em contacto com o chão, $P_3 - (x_3, y_3)$, e a zona de acoplamento ao veio transmissor, $P_0 - (x_0, y_0)$.

De referir que a haste é um único elemento de comprimento L , estando representado na figura como dois segmentos L_1 e L_2 .

4.2 - Definição das equações de movimento para os pontos críticos

4.2.1 – Perna no chão

As equações do movimento para os pontos anteriormente referidos são calculadas em função dos ângulos θ e β . Partindo do princípio que a perna do robot está fixa no chão, as equações são calculadas tomando como ponto de referencia o ponto P_3 , que é o ponto fixo.

$$\beta = -\frac{\pi}{2} + a \cos\left(\frac{-r \cos(\theta)}{L_1}\right)$$

$$\theta = f(t)$$

$$\theta = [\pi, 2\pi]$$

O ponto P_3 está fixo e convencionou-se a origem das coordenadas durante este meio ciclo.

$$\begin{cases} x_3 = 0 \\ y_3 = 0 \end{cases}$$

Sendo x_0 e y_0 as coordenadas do ponto P_0 , temos

$$\begin{cases} x = (L_1 + L_2) \sin(\beta) \\ y = -r \sin(-\theta) + L_2 \cos(\beta) \end{cases}$$

Para o ponto P_1 , as coordenadas são

$$\begin{cases} x_1 = -r \cos(\theta) + (L_1 + L_2) \sin(\beta) \\ y_1 = L_2 \cos(\beta) \end{cases}$$

As coordenadas de P_2 são

$$\begin{cases} x_2 = (L_1 + L_2) \sin(\beta) \\ y_2 = (L_1 + L_2) \cos(\beta) \end{cases}$$



Expressando as equações dos pontos P_0 , P_1 e P_2 só em função do ângulo θ , obtêm-se as seguintes equações:

- para P_0

$$\begin{cases} x_0 = \frac{L_1 + L_2}{L_1} r \cos(\theta) \\ y_0 = r \sin(\theta) + L_2 \sqrt{1 - \frac{r^2}{L_1^2} \cos^2(\theta)} \end{cases}$$

- para P_1

$$\begin{cases} x_1 = r \cos(\theta) + \frac{L_1 + L_2}{L_1} r \cos(\theta) \\ y_1 = L_2 \sqrt{1 - \frac{r^2}{L_1^2} \cos^2(\theta)} \end{cases}$$

- para P_2

$$\begin{cases} x_2 = \frac{L_1 + L_2}{L_1} r \cos(\theta) \\ y_2 = (L_1 + L_2) \sqrt{1 - \frac{r^2}{L_1^2} \cos^2(\theta)} \end{cases}$$

4.2.2 – *Perna no ar*

Durante o movimento do robot estão constantemente três pernas em contacto com o solo e três a rodar livremente. Nesta situação, em que a perna está no ar, as equações que traduzem o seu movimento são expressas em relação ao ponto P_0 .

$$\theta = [0, \pi]$$

As coordenadas para o ponto P_0 , sendo K_x e K_y parâmetros constantes, que dependem do afastamento entre duas pernas consecutivas, são

$$\begin{cases} x_0 = K_x \\ y_0 = K_y \end{cases}$$



No que diz respeito às coordenadas do ponto P_1 ,

$$\begin{cases} x_1 = r \cos(\theta) \\ y_1 = r \sin(\theta) \end{cases}$$

Sendo as coordenadas x_2 e y_2 do ponto P_2 vem

$$\begin{cases} x_2 = x_0 \\ y_2 = L_1 \cos(\beta) + y_1 \end{cases}$$

Finalmente, para o ponto P_3 temos

$$\begin{cases} x_3 = x_2 + (L_1 + L_2) \sin(\beta) \\ y_3 = y_2 - (L_1 + L_2) \cos(\beta) \end{cases}$$

Expressando as equações dos pontos P_1 , P_2 e P_3 só em função do ângulo θ_0 obtêm-se as seguintes equações:

- para P_1

$$\begin{cases} x_1 = r \cos(\theta) + x_0 \\ y_1 = r \sin(\theta) + y_0 \end{cases}$$

- para P_2

$$\begin{cases} x_2 = x_0 \\ y_2 = r \sin(\theta) + L_1 \sqrt{1 - \frac{r^2}{L_1^2} \cos^2(\theta)} + y_0 \end{cases}$$



- para P_3

$$\begin{cases} x_3 = \frac{L_1 + L_2}{L_1} r \cos(\theta) + x_0 \\ y_3 = r \sin(\theta) + L_1 \sqrt{1 - \frac{r^2}{L_1^2} \cos^2(\theta)} - (L_1 + L_2) \sqrt{1 - \frac{r^2}{L_1^2} \cos^2(\theta)} + y_0 \end{cases}$$

5 - Simulação da Locomoção do Robot

Depois de definidos todos os pontos relevantes e encontradas as equações matemáticas que traduzem o seu movimento, efectuou-se a simulação de uma perna e posteriormente do conjunto das três pernas laterais.

O resultado desta simulação ilustra-se na figura 5, que representa a vista lateral do robot.

A simulação permitiu ajustar alguns parâmetros tais como, o comprimento do elo de transmissão de movimento, a dimensão da haste e respectivo ponto de fixação e a distância entre pernas (offset).

Com a simulação foi ainda possível analisar algumas alterações, dando-se especial destaque para a distância entre a extremidade das pernas quando estão todas no chão, que deve ser suficiente para permitir o acoplamento de umas borrachas para evitar escorregamentos e garantir que nunca se tocam durante a sua trajectória.

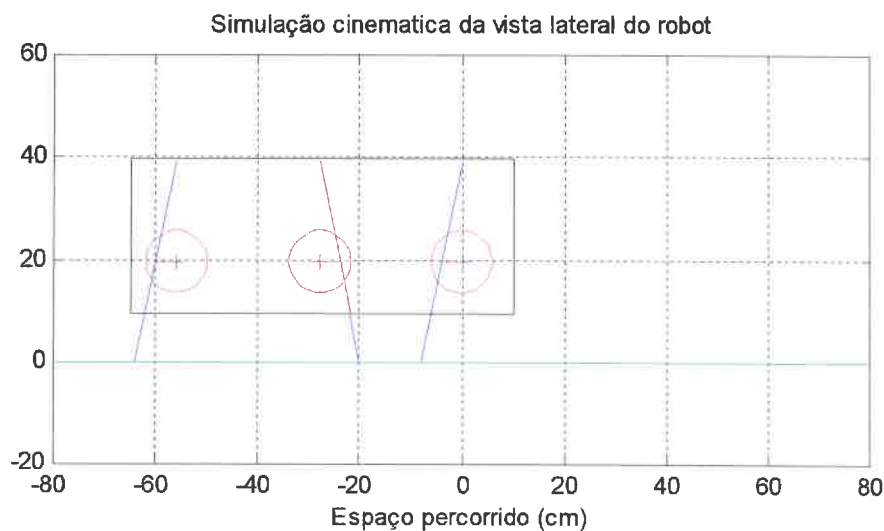


Figura N.º2 – Simulação cinemática da vista lateral do robot

Na simulação da vista de topo representa-se a extremidade das pernas, a estrutura do robot e o seu centro de massa, assumindo que é coincidente com o centro geométrico do robot.

A verde está representado um triângulo imaginário, que serve como fronteira de estabilidade, isto é, o centro de massa (a cor de rosa) não deve passar estas linhas, com o risco do robot sofrer um desequilíbrio e risco de queda.

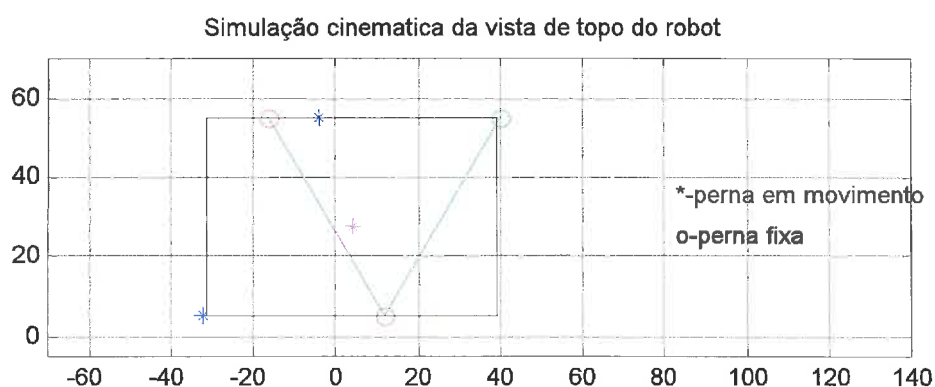


Figura N.º3 – Simulação cinemática da vista de topo do robot

Com as simulações confirmou-se a validade das equações do movimento e pode-se também verificar e aferir parâmetros dimensionais na estrutura do robot.

5.1 - Estudo dimensional do robot

Na tabela seguinte, estão evidenciadas as situações que respeitam algumas restrições impostas no início do dimensionamento, entre as quais se destacam a distância mínima entre pernas, o offset e outros parâmetros que irão ser abordados mais à frente.

Apesar de não estar presente em todas as colunas da tabela as unidades utilizadas, todos os valores apresentados na mesma estão expressos em centímetros.



Dimensões da estrutura	L1 (cm)	L2 (cm)	R= L2/L1	L (cm)	r (cm)	beta (rad)	Passo (cm)	Distancia Máxima	Distancia mínima
50*100	10	50	5	60	4	0.412	96	88	-8
50*100	10	50	5	60	5	0.524	120	100	-20
50*100	15	45	3	60	3	0.201	48	64	16
50*100	15	45	3	60	4	0.270	64	72	8
50*100	15	45	3	60	5	0.340	80	80	0
50*100	20	40	2	60	3	0.151	36	58	22
50*100	20	40	2	60	4	0.201	48	64	16
50*100	20	40	2	60	5	0.253	60	70	10
50*100	25	35	1.4	60	5	0.201	48	64	16
50*100	30	30	1	60	5	0.167	40	60	20
25*50	7.5	22.5	3	30	1.5	0.201	24	32	8
50*70	12.5	30	2.4	42.5	4	0.326	54.4	57.2	2.8
50*70	20	20	1	40	4	0.201	32	44	12
50*80	17	28	1.6	45	6	0.361	63.5	66.8	3.2

Tabela N.º1 – Apresentação dos parâmetros

Antes de se prosseguir com a análise propriamente dita da Tabela 1 fez-se um breve levantamento dos parâmetros presentes na tabela que ainda não foram abordados.

Assim sendo, vem:

r - é o comprimento entre os pontos P_0 e P_1 do elo de transmissão;

β - é o ângulo máximo de abertura que a perna faz com a vertical;

Passo - é o espaço percorrido por uma perna, no fim de uma rotação completa;

Distância Máxima - afastamento máximo entre duas pernas consecutivas;

Distância mínima - afastamento mínimo entre duas pernas consecutivas ou afastamento mínimo de segurança

Offset - distância entre os veios de transmissão de movimento às pernas do robot.

No início deste estudo, as dimensões do robot apontavam para um metro de comprimento por meio metro de largura. A velocidade estipulada seria de meio metro por segundo.

Porém estes parâmetros foram sofrendo alterações, para aproveitar os motores e “drives” de potência existentes no departamento. Uma vez que os motores não suportavam o peso previsto para a dimensão da estrutura inicial, esta foi reduzida para metade. Então outro problema surgiu: falta de espaço dentro da estrutura para a colocação de todos os componentes (motores, elementos de transmissão e controlo, etc).



Após algum estudo no dimensionamento, foi estabelecido a dimensão de setenta centímetros de comprimento por meio metro de largura.

O comprimento total (L) da haste ficou com quarenta centímetros, onde $L_1=L_2=20$ cm para se conseguir uma distância mínima de doze centímetros. Deste modo o passo ficou com trinta e dois centímetros. De referir ainda que o comprimento do elo de ligação (r) ficou com quatro centímetros.

Com este conjunto de parâmetros, o passo foi reduzido para 32 cm, isto é, se a velocidade for de uma rotação por segundo, o robot desloca-se 32 cm por segundo.

Continuando a análise da tabela 1, e ainda abordando a importância da distância mínima entre pernas, é de referir que no caso dos dois primeiro conjunto de parâmetros, esta toma valores negativos, excluindo logo essas duas hipóteses, pois isto significa que as pernas iriam se cruzar umas com as outras durante a sua trajectória.

5.2 - Planeamento de trajectória

Para evitar em que surjam embates violentos no solo tem que se garantir que as pernas tenham uma velocidade muito baixa ou mesmo nula quando se estão a aproximar e a afastar do chão.

O movimento efectuado pelo robot pode-se dividir em duas fases: uma 1ª fase de movimento livre das pernas, quando estas rodam entre 0 e π e uma 2ª fase de impulsão, quando há uma rotação entre π e 2π .

A função do movimento depende do ângulo θ , que varia em função do tempo.

$$\theta = f(t)$$

$$\theta = [0, 2\pi]$$

Partindo do princípio que a perna tem velocidade inicial e final zero nesta fase, pode-se utilizar um polinómio de 3ª ordem para calcular a equação que define a evolução do ângulo em função do tempo e respectivas derivadas.



Um polinómio de 3ª ordem pode-se expressar de uma forma geral por,

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

Aplicando a expressão anterior à variação do ângulo, calculando as suas derivadas, obtêm-se as fórmulas da velocidade e aceleração do movimento efectuado pelo veio motor.

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t$$

Na primeira fase do movimento θ varia entre 0 e π . Então, para uma rotação por segundo, para o instante inicial t_i o ângulo inicial é:

$$\theta_{ii} = 0$$

$$\theta(0) = 0$$

Como,

$$\Downarrow$$

$$\theta = a_0$$

$$\theta_{ii} = a_0$$

então

$$a_0 = 0$$

Utilizando a expressão da velocidade, pode-se calcular a_1 , uma vez que no instante final (t_f) deverá ter velocidade nula. Como

$$\theta_{ff} = \pi$$

$$\dot{\theta}_{ff} = 0$$

então

$$a_1 = 0$$

$$\left(\begin{array}{l} \theta(t_f) = \pi \\ \dot{\theta}(t_f) = 0 \end{array} \right)$$

$$t_f = 1 (?)$$

$$\dot{\theta}(0) = 0$$

$$\Downarrow$$

$$\theta = a_1 + 0 + 0$$

Substituindo os valores de a_0 e a_1 no seguinte sistema de equações, calculam-se as variáveis a_2 e a_3 e a expressão final para a variação do ângulo θ .



$$\begin{aligned}
 t(t_f) &= a_2 t_f^2 + a_3 t_f^3 = \pi \\
 \dot{\theta}(t_f) &= 2a_2 t_f + 3a_3 t_f^2 = 0 \Rightarrow 2a_2 = -3a_3 t_f
 \end{aligned}$$

$$\begin{cases}
 \theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\
 \dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2
 \end{cases}
 \begin{cases}
 a_2 = \frac{3(\theta_f - \theta_i)}{t_f^2} \\
 a_3 = \frac{-2(\theta_f - \theta_i)}{t_f^3}
 \end{cases}$$

\Rightarrow
 $a_0 = 0$
 $a_1 = 1$

Desta forma a expressão geral da variação do ângulo traduz-se por:

$$\theta(t) = \theta_i + \frac{3(\theta_f - \theta_i)}{t_f^2} t^2 + \frac{-2(\theta_f - \theta_i)}{t_f^3} t^3$$

Substituindo os valores do ângulo do instante inicial e do instante final obtêm-se as expressões finais.

derivadas: $t_f = 1$

$$\begin{aligned}
 \theta(t) &= 12\pi t^2 - 16\pi t^3 && 3\pi t^2 - 2\pi t^3 \\
 \dot{\theta}(t) &= 24\pi t - 48\pi t^2 && 6\pi t - 6\pi t^2 \\
 \ddot{\theta}(t) &= 24\pi - 96\pi t && 6\pi - 12\pi t
 \end{aligned}$$

Para a segunda fase do movimento, $\theta_i = \pi$ e $\theta_f = 2\pi$, substituindo estes valores na expressão geral anteriormente calculada obtêm-se as seguintes equações:

$$\begin{aligned}
 \theta(t) &= \pi + 12\pi t^2 - 16\pi t^3 \\
 \dot{\theta}(t) &= 24\pi t - 48\pi t^2 \\
 \ddot{\theta}(t) &= 24\pi - 96\pi t
 \end{aligned}$$

Quando não há planeamento da trajetória, a velocidade varia linearmente durante a locomoção, e as equações do movimento são muito simples:

$$\begin{aligned}
 \theta(t) &= kt \\
 \dot{\theta}(t) &= k \\
 \ddot{\theta}(t) &= 0
 \end{aligned}$$



Depois de determinadas as expressões do movimento com e sem planeamento de trajectória, elaborou-se um programa em Matlab a partir do qual se extraíram os seguintes gráficos:

- Sem planeamento da trajectória:

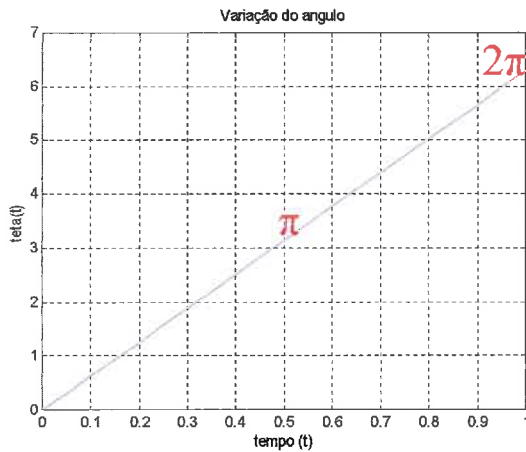


Gráfico N.º1 - Variação do ângulo

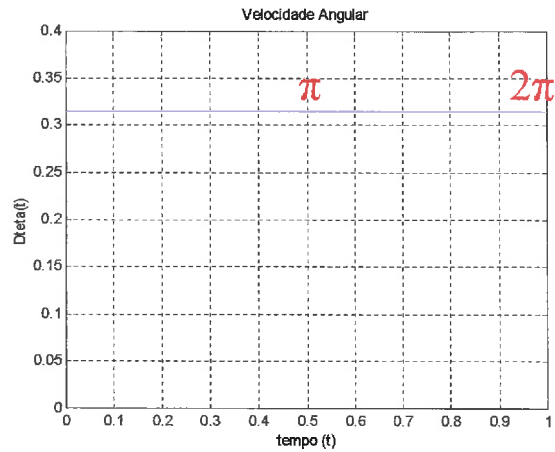


Gráfico N.º2 - Velocidade angular

- Com planeamento da trajectória:

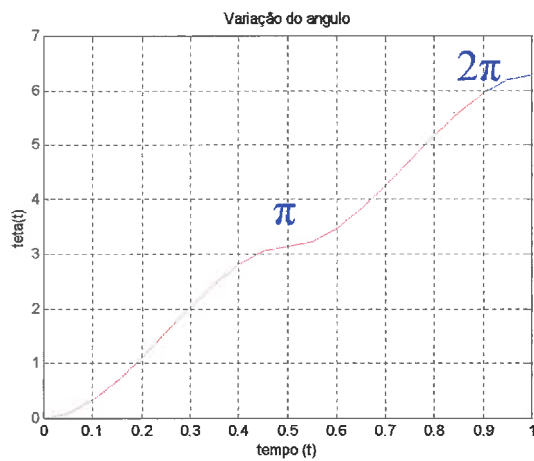


Gráfico N.º3 - Variação do ângulo

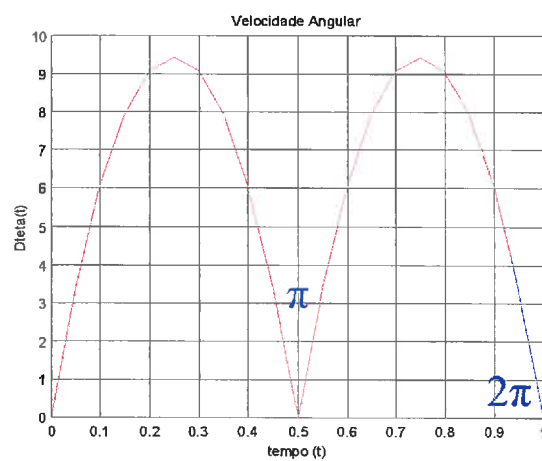


Gráfico N.º4 - Variação da velocidade angular



Os gráficos 1 e 2 representam a variação do ângulo θ , e a variação da velocidade angular, $d(\theta)$, respectivamente, para a situação em que o ângulo evolui linearmente ao longo do movimento, resultando uma velocidade angular constante.

Os gráficos 3 e 4 são representativos do funcionamento de uma perna do robot numa situação em que existe o planeamento de trajectória.

Analisando os gráficos do estudo de planeamento de trajectória, destaca-se o facto da variação do ângulo ser mais suave quando as pernas do robot se estão a aproximar e a afastar do solo, coincidindo com os valores de π e 2π (periodicamente). Esta atenuação do ângulo reflecte-se na redução da velocidade angular para valores cada vez mais baixos até assumir o valor zero, quando se encontra em π e 2π , tal como era pretendido. A velocidade será máxima quando a perna estiver na posição vertical ($\pi/2$ e $3/2\pi$) e mínima nas posições atrás referidas.

6 – Estudo Dinâmico do Robot Hexápode

6.1- Estudo das forças e binários envolvidos durante a locomoção

Utilizaram-se as equações do movimento com planeamento de trajectória do ponto P_0 , ponto de união ao veio do motor, para o cálculo do binário e da potência.

O ângulo de rotação do veio transmissor, a velocidade e aceleração angular expressam-se segundo as equações já conhecidas.

$$\theta = \pi + 12\pi t^2 - 16\pi t^3$$

$$\dot{\theta} = 24\pi t - 48\pi t^2$$

$$\ddot{\theta} = 24\pi - 96\pi t$$



As equações do movimento do ponto $P_0(x_0, y_0)$ em função do ângulo de rotação θ , apresentam-se a seguir.

$$\begin{cases} x_0 = \frac{L_1 + L_2}{L_1} r \cos(\theta) \\ y_0 = r \sin(\theta) + L_2 \sqrt{1 - \frac{r^2}{L_1^2} \cos^2(\theta)} \end{cases}$$

Para simplificar estas equações, atribuiu-se

$$\begin{cases} k_1 = \frac{L_1 + L_2}{L_1} r \\ k_2 = \frac{r^2}{L_1^2} \end{cases}$$

Substituindo

$$\begin{cases} x_0 = k_1 \cos(\theta) \\ y_0 = r \sin(\theta) + L_2 \sqrt{1 - k_2 \cos^2(\theta)} \end{cases}$$

6.2 - Cálculo da energia do sistema utilizando o Teorema de Lagrange

O Teorema de Lagrange define a energia total de um sistema (L) como a diferença entre a energia cinética (E_c) e a energia potencial (P).

A energia cinética e potencial do sistema podem-se expressar segundo várias coordenadas, não sendo obrigatório o uso das coordenadas cartesianas, podendo haver deste modo uma simplificação do problema. Desta forma, pelo teorema de Lagrange é possível calcular o momento de uma força τ .



Teorema de Lagrange:

$$L = E_c - P$$

Analisando o Teorema de Lagrange e desdobrando a energia total do sistema na sua energia cinética e potencial, rapidamente se chega a uma expressão em que o binário vai estar relacionado com as derivadas parciais destas duas energias.

$$\tau = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} \quad \Rightarrow \quad \tau = \frac{d}{dt} \frac{\partial E_c}{\partial \dot{\theta}} - \left[\frac{\partial E_c}{\partial \theta} - \frac{\partial P}{\partial \theta} \right]$$

6.2.1 - Cálculo da energia cinética:

É necessário proceder ao cálculo da energia cinética, para o qual é fundamental conhecer a expressão da velocidade.

$$E_c = \frac{1}{2} m v^2$$

Sabendo que o quadrado da velocidade não é mais do que a soma dos quadrados das velocidades segundo os eixos x e y ,

$$v^2 = (\dot{x}_0^2 + \dot{y}_0^2)$$

Derivando as equações do movimento do ponto P_0 , obtêm-se as equações da velocidade.

$$\begin{cases} \dot{x}_0 = -k_1 \sin(\theta) \dot{\theta} \\ \dot{y}_0 = r \cos(\theta) \dot{\theta} + \frac{L_2 k_2 \cos(\theta) \sin(\theta)}{\sqrt{1 - k_2 \cos(\theta)^2}} \dot{\theta} \end{cases}$$

Por substituição na fórmula geral da energia cinética, resulta a seguinte expressão para o sistema:

$$E_c = \frac{1}{2} m \left[\left[k_1 \sin(\theta) \dot{\theta} \right]^2 + \left(r \cos(\theta) \dot{\theta} + \frac{L_2 k_2 \cos(\theta) \sin(\theta) \dot{\theta}}{\sqrt{1 - k_2 \cos(\theta)^2}} \right)^2 \right]$$

Nesta fase, também foi necessário calcular as expressões da derivada da energia cinética em relação ao ângulo de rotação θ .

$$\frac{\partial E_c}{\partial \dot{\theta}} = \frac{1}{2} m \left[2k_1^2 \sin(\theta) \dot{\theta}^2 \cos(\theta) + 2 \left(r \cos(\theta) \dot{\theta} + \frac{L_2 k_2 \cos(\theta) \sin(\theta) \dot{\theta}}{\sqrt{1 - k_2 \cos(\theta)^2}} \right) \times \right. \\ \left. \times \left(r \cdot \sin(\theta) \dot{\theta} - \frac{L_2 k_2 \sin(\theta)^2 \dot{\theta} + L_2 k_2 \cos(\theta)^2 \dot{\theta}}{\sqrt{1 - k_2 \cos(\theta)^2}} - \frac{L_2 k_2^2 \cos(\theta)^2 \sin(\theta)^2 \dot{\theta}}{(1 - k_2 \cos(\theta)^2)^{3/2}} \right) \right]$$

Calculou-se a equação referente à derivada da energia cinética em função da velocidade angular.

$$\frac{\partial E_c}{\partial \dot{\theta}} = \frac{1}{2} m \left[2k_1^2 \sin(\theta)^2 \dot{\theta} + 2 \left(r \cos(\theta) \dot{\theta} + \frac{L_2 k_2 \cos(\theta) \sin(\theta) \dot{\theta}}{\sqrt{1 - k_2 \cos(\theta)^2}} \right) \times \right. \\ \left. \times \left(r \cos(\theta) + \frac{L_2 k_2 \cos(\theta) \sin(\theta)}{(1 - k_2 \cos(\theta)^2)^{1/2}} \right) \right]$$

Para simplificar estas equações, atribuiu-se

$$\begin{cases} k_3 = L_2 k_2 \\ k_4 = 2rL_2 k_2 = 2rk_3 \end{cases}$$

Para finalizar a formulação do estudo da energia cinética, obteve-se a expressão da derivada em ordem ao tempo, da derivada da energia cinética em ordem à velocidade angular.

$$\begin{aligned} \frac{d}{dt} \frac{\partial E_c}{\partial \dot{\theta}} = m \ddot{\theta} & \left[k_1^2 \sin(\theta)^2 + r^2 \cos(\theta)^2 + \frac{k_3^2 \cos(\theta)^2 \sin(\theta)^2}{1 - k_2 \cos(\theta)^2} + \frac{k_4 \cos(\theta)^2 \sin(\theta)}{\sqrt{1 - k_2 \cos(\theta)^2}} \right] + \\ & + m \dot{\theta} \left[2k_1^2 \sin(\theta) \cos(\theta) \dot{\theta} - 2r^2 \sin(\theta) \cos(\theta) \dot{\theta} + \right. \\ & + k_3^2 \dot{\theta} \left[\frac{(2 \cos(\theta) \sin(\theta)^3 + 2 \cos(\theta)^3 \sin(\theta))(1 - k_2 \cos(\theta)^2) - 2k_2 \cos(\theta)^3 \sin(\theta)^3}{(1 - k_2 \cos(\theta)^2)^2} \right] + \\ & \left. + k_4 \left[\frac{(\cos(\theta)^3 \dot{\theta} - 2 \cos(\theta) \sin(\theta)^2 \dot{\theta})(1 - k_2 \cos(\theta)^2) - k_2 \cos(\theta)^3 \sin(\theta)^2 \dot{\theta}}{(1 - k_2 \cos(\theta)^2) \sqrt{1 - k_2 \cos(\theta)^2}} \right] \right] \end{aligned}$$

6.2.2 - Cálculo da energia potencial

No que diz respeito à energia potencial, para além do valor da aceleração da gravidade, g , e da massa total do robot, m , apenas se necessita de conhecer as coordenadas em y do ponto em estudo (P_0).

A fórmula geral da energia potencial é do conhecimento geral e é expressa da seguinte forma:

$$P = mgy$$

Por substituição da expressão de y_0 obtém-se a fórmula da energia potencial específica do sistema.

$$P = mg(r \sin(\theta) + L_2 \sqrt{1 - k_2 \cos(\theta)^2})$$



Finalmente, calculou-se a derivada da energia potencial em ordem ao ângulo de rotação θ , para a qual se apresenta a expressão.

$$\frac{\partial P}{\partial \theta} = mg \left[r \cos(\theta) + L_2 \frac{k_2 \cos(\theta) \sin(\theta)}{\sqrt{1 - k_2 \cos(\theta)^2}} \right]$$

Por substituição directa das fórmulas já mencionadas obtém-se a seguinte expressão para o momento da força envolvida durante a locomoção do robot.

$$\begin{aligned} \tau = m \ddot{\theta} & \left[k_1^2 \sin(\theta)^2 + r^2 \cos(\theta)^2 + \frac{k_3^2 \cos(\theta)^2 \sin(\theta)^2}{1 - k_2 \cos(\theta)^2} + \frac{k_4 \cos(\theta)^2 \sin(\theta)}{\sqrt{1 - k_2 \cos(\theta)^2}} \right] + \\ & + m \dot{\theta} [2k_1^2 \sin(\theta) \cos(\theta) \dot{\theta} - 2r^2 \sin(\theta) \cos(\theta) \dot{\theta} + \\ & + k_3^2 \dot{\theta} \left[\frac{(-2 \cos(\theta) \sin(\theta)^3 + 2 \cos(\theta)^3 \sin(\theta))(1 - k_2 \cos(\theta)^2) - 2k_2 \cos(\theta)^3 \sin(\theta)^3}{(1 - k_2 \cos(\theta)^2)^2} \right] + \\ & + k_4 \left[\frac{(\cos(\theta)^3 \dot{\theta} - 2 \cos(\theta) \sin(\theta)^2 \dot{\theta})(1 - k_2 \cos(\theta)^2) - k_2 \cos(\theta)^3 \sin(\theta)^2 \dot{\theta}}{(1 - k_2 \cos(\theta)^2) \sqrt{1 - k_2 \cos(\theta)^2}} \right] \times \\ & \times \left[\frac{1}{2} m \left[2k_1^2 \sin(\theta) \dot{\theta}^2 \cos(\theta) + 2 \left(r \cos(\theta) \dot{\theta} + \frac{L_2 k_2 \cos(\theta) \sin(\theta)}{\sqrt{1 - k_2 \cos(\theta)^2}} \dot{\theta} \right) \times \right. \right. \\ & \left. \left. \times \left(-r \sin(\theta) \dot{\theta} \frac{L_2 k_2 \sin(\theta)^2 \dot{\theta} + L_2 k_2 \cos(\theta)^2 \dot{\theta}}{\sqrt{1 - k_2 \cos(\theta)^2}} - \frac{L_2 k_2^2 \cos(\theta)^2 \sin(\theta)^2 \dot{\theta}}{(1 - k_2 \cos(\theta)^2)^{3/2}} \right) \right] - \right. \\ & \left. - mg \left[r \cos(\theta) + L_2 \frac{k_2 \cos(\theta) \sin(\theta)}{\sqrt{1 - k_2 \cos(\theta)^2}} \right] \right] \end{aligned}$$

Nesta fase, uma vez calculado o binário, rapidamente se calcula a potência envolvida, uma vez que esta se pode expressar pelo produto do binário (τ) com a velocidade angular (ω).

$$P = \tau\omega$$

Para finalizar este estudo, implementaram-se estas equações, recorrendo ao Matlab, representando-se graficamente a evolução do momento e da potência despendida durante a locomoção.

Este estudo foi efectuado para as duas situações possíveis já mencionadas (com e sem planeamento de trajectória). As equações para o estudo sem planeamento não são aqui apresentadas, pois são exactamente as mesmas, tendo em atenção que a componente da aceleração é nula.

Os parâmetros utilizados neste estudo foram extraídos da tabela 1, dando maior destaque aos que cumpriram da melhor forma as especificações iniciais (versatilidade, distância de segurança entre pernas, passo, etc...). Para estes parâmetros representaram-se os gráficos de binário e potência sobre os quais se fez uma breve análise.

Parâmetros: L1 = 0.15mm;
L2 = 0.45mm;
r = 3mm;
M = 25kg

- Sem planeamento

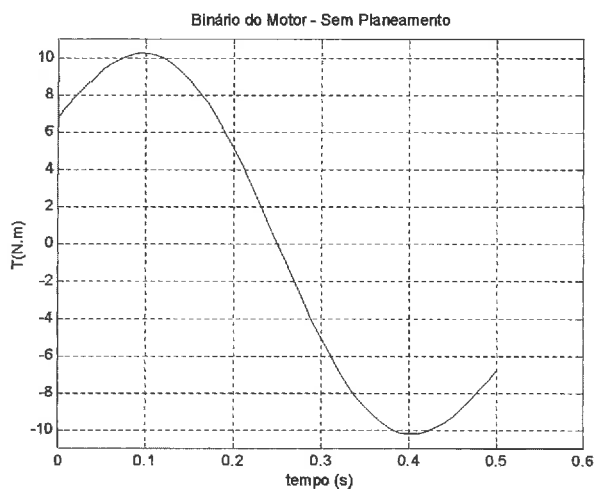


Gráfico N.º5 - Binário do motor para o movimento sem planeamento de trajectória

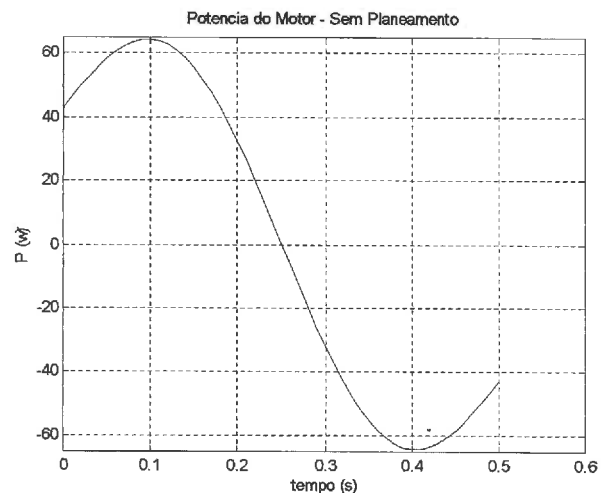


Gráfico N.º6 - Potência do motor para o movimento sem planeamento de trajectória



- Com planeamento

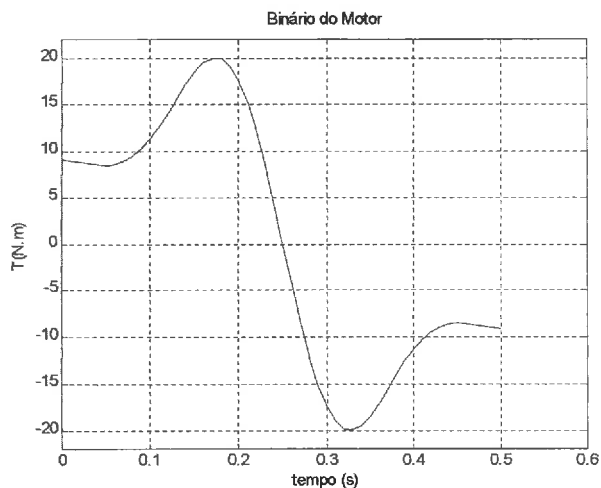


Gráfico N.º7 - Binário do motor para o movimento com planeamento de trajectória

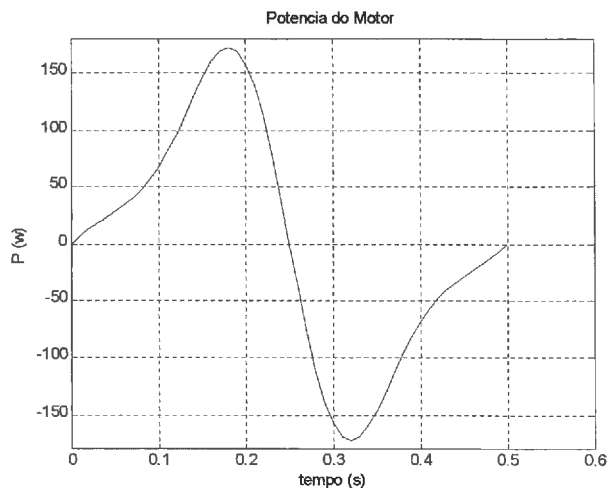


Gráfico N.º8 - Potência do motor para o movimento sem planeamento de trajectória

Parâmetros:

$L1 = 0.20\text{mm};$

$L2 = 0.20\text{mm};$

$r = 4\text{mm};$

$M = 25\text{ kg}; \rightarrow M = 40\text{ kg}$

- Sem planeamento

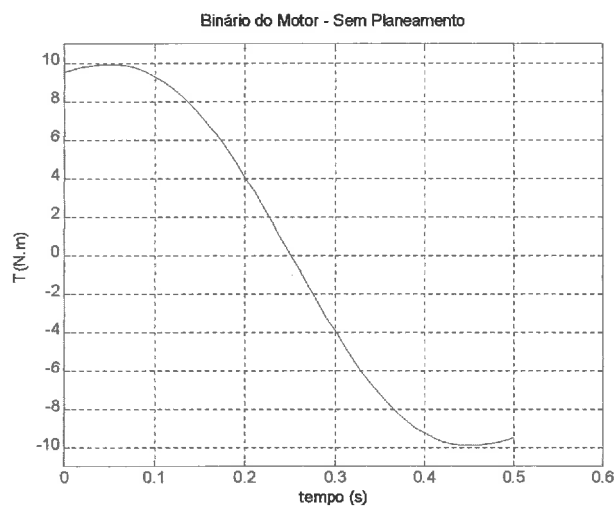


Gráfico N.º9 - Binário do motor para o movimento sem planeamento de trajectória

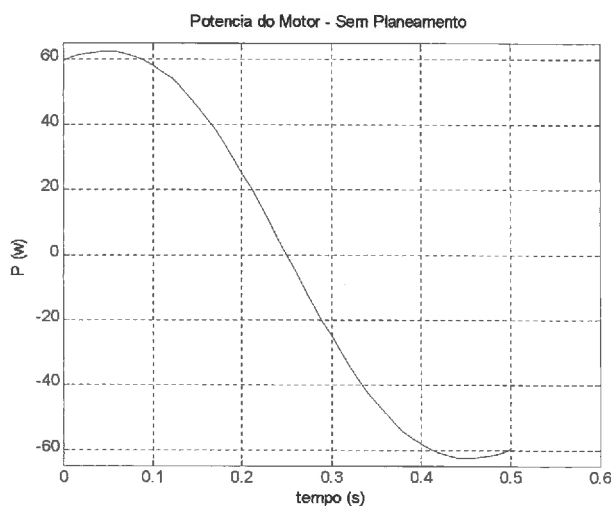


Gráfico N.º10 - Potência do motor para o movimento sem planeamento de trajectória



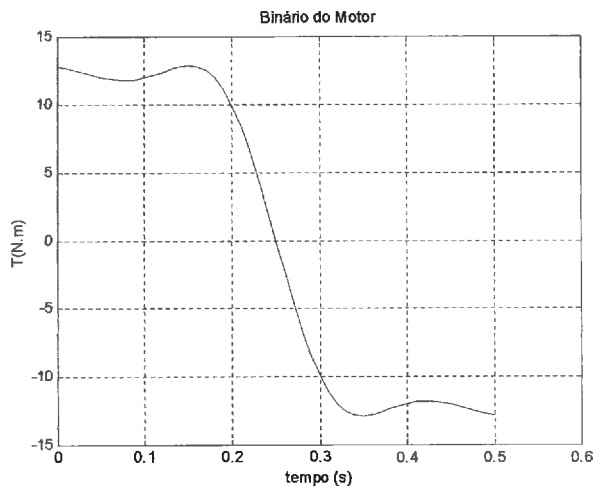
- Com planeamento

Gráfico N.º11 - Binário do motor para o movimento com planeamento de trajectória

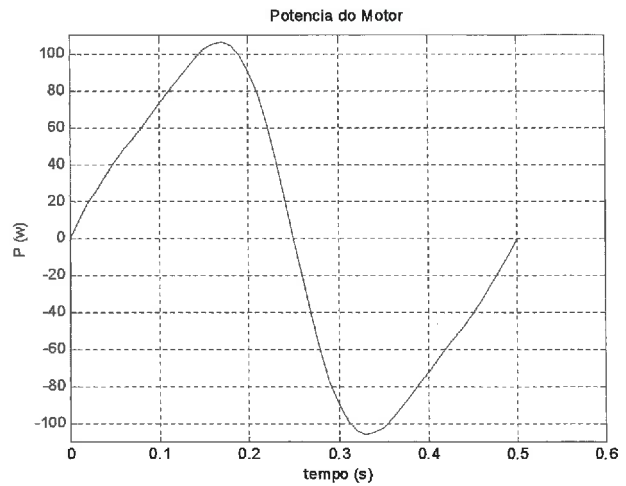


Gráfico N.º12 - Potência do motor para o movimento sem planeamento de trajectória

Analisando os gráficos verifica-se que o comportamento das curvas de binário e de potência são muito diferentes nas situações em que o movimento se realiza com e sem planeamento de trajectória.

Comparando os valores do binário para os dois conjuntos de parâmetros apresentados anteriormente, observa-se que no primeiro caso, o binário máximo atingido é de 20 Nm enquanto que para a segunda situação é de 13 Nm.

Por este motivo, o estudo e projecto do robot seguiram em frente utilizando estes últimos parâmetros como base.

7 - Estudo dos Elementos Necessários Para a Construção do Robot

O objectivo principal deste estudo foi fazer, de uma maneira sequencial, todo o raciocínio necessário para a construção do robot.

A primeira coisa que se fez foi dimensionar estruturalmente o robot, afastando logo qualquer outra solução que não fosse a utilização de perfil de alumínio. Deste modo, foi ganho algum tempo na montagem da estrutura, assim como alguma flexibilidade na sua estruturação.

Assim, optou-se por um perfil quadrado 35x35mm e para as pernas do robot utilizou-se um perfil 20x10mm.

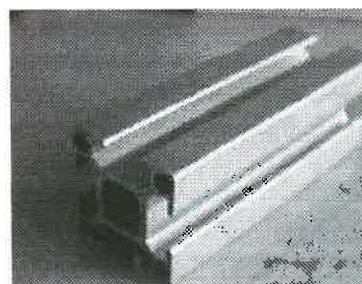


Figura N.º4 – Perfil utilizado na estrutura.



Figura N.º5 – Motor I.

Se se trata de um robot, este vai ter que se mover, e para isso precisa de actuadores. Foram utilizados dois motores, visto que um só motor não era o suficiente, pois além de ser necessária mais potência, com o conseqüente aumento de tamanho do motor, não era possível a rotação do robot, impossibilitando-o de virar.

De seguida veio a escolha do tipo de motores a utilizar. Para isso foi necessário pensar no tipo de movimento pretendido para o robot, o que levou à escolha de motores de maneira que fosse possível controlar a velocidade.

Pensou-se então nos servomotores e nos motores passo a passo.

Devido ao elevado preço dos servomotores, que foi um factor preponderante, e à existência de motores passo a passo no departamento, optou-se por estes.

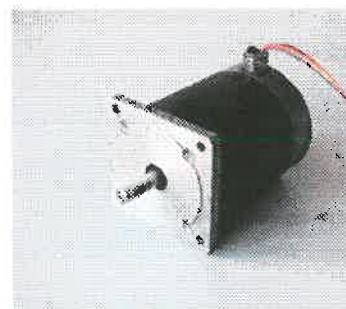


Figura N.º6 – Motor II.

Como se pode ver nas figuras 5 e 6, os dois motores utilizados são diferentes. Utilizando a internet (via e-mail – ver anexo 5), conseguiu-se obter informação sobre o motor ilustrado na figura 5; em relação ao outro motor, não se possui documentação.

De seguida, pensou-se no PLC (Programmable Logical controller) e nas respectivas cartas de eixo, necessárias duas, uma para cada motor. Era preciso um PLC com poucas entradas e saídas e que fosse alimentado a 24V.

Então, o PLC escolhido foi o Misubishi FX_{2n} 16MR – DS com 8 entradas e saídas, e as cartas de eixo são as FX_{2n} -1PG.

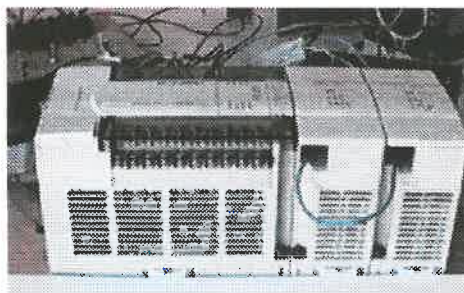


Figura N.º7 – Controlador programável, Misubishi FX_{2n} 16MR – DS.



Figura N.º8 – Regulador de tensão para 12 V.

As cartas necessitam de um regulador de tensão, figura 8, uma vez que estas não devem ser alimentadas a 24V, mas sim a 12V para garantir essa tensão nos impulsos por elas geradas para a unidade de potência, visto que é a tensão recomendada para o seu perfeito funcionamento.

Deste modo, o PLC e as cartas, vão gerar a frequência dos impulsos necessários ao funcionamento dos motores.

No que diz respeito à unidade de potência, como já foi dito anteriormente, não havia qualquer informação sobre esta unidade, o que levou algumas horas a estudá-la, para tentar perceber o seu funcionamento. Através deste estudo, foi elaborado o seu esquema de funcionamento que está representado no anexo 1. De referir ainda que esta unidade também foi “reciclada”, aqui do departamento, de uma máquina que foi desmantelada.

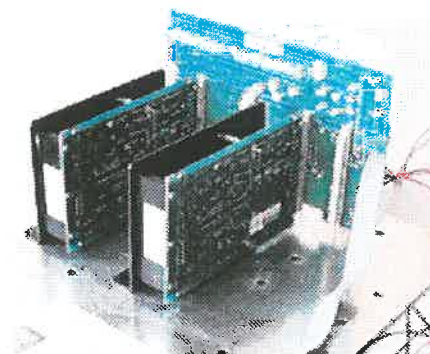


Figura N.º9 – Unidade de potência.

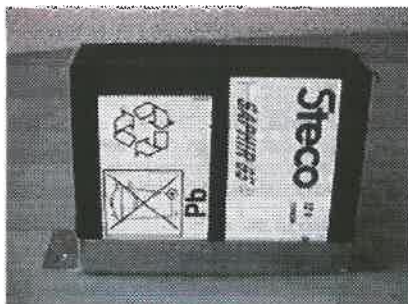


Figura N.º10 – Baterias 12V.

Uma vez que o robot é autónomo, é necessário alimentá-lo a baterias. Para isso adquiriram-se quatro baterias de chumbo Steco, de 12V com 6,5A e um recarregador Bradex BX60.



Figura N.º11 Carregador de baterias

As baterias estão ligadas duas a duas em série, para se conseguir os 24V, dispostas em duas caixas independentes.

Uma vez que os motores não têm potência suficiente para elevar o peso total do robot, foi necessário recorrer a redutores para aumentar o binário.

O binário máximo fornecido por um dos motores é de 3,5Nm. Assim decidiu-se comprar dois redutores com uma redução de 1:15. Este valor foi baseado no estudo dinâmico feito na simulação, ou seja, para uma massa de 25kg conforme apresentado nos gráficos 11 e 12.

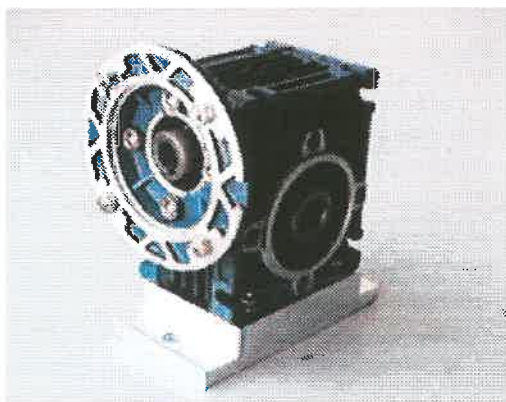


Figura N.º12– Redutor 1:15.

Tendo em conta que este valor era um valor provável, e que o outro motor teria, em princípio, um binário máximo inferior aos 3,5Nm, optou-se por uma redução maior para garantir que à saída do redutor há força suficiente para a locomoção do robot.

É importante referir que a aquisição dos redutores foi um dos principais motivos, senão o principal, do atraso da junção de todos os componentes e, deste modo, da

montagem final deste projecto. Mesmo assim, pode-se dizer que não foi a solução ideal, mas sim uma solução de recurso, já que foi necessário fazer um veio para unir estes dois elementos e deste modo ser possível a transmissão de movimento.



Figura N.º13– Veio de união entre o redutor e o motor.

O movimento é transmitido às pernas centrais, que por sua vez, por intermédio de rodas dentadas e correias, fazem mover os outros membros do robot.

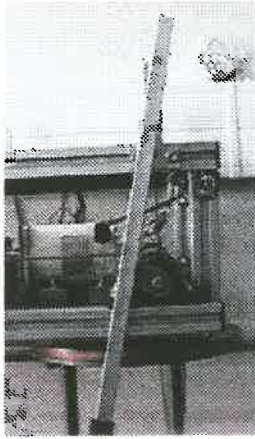


Figura N.º14
Perna do robot

Observando a figura 14 e comparando com a figura 1, que não é mais que um esquema do estudo da simulação efectuada inicialmente, verifica-se que esta simulação em nada fugiu da realidade.

É possível também observar que o movimento transmitido à perna do robot passa, entre outros elementos, por um veio que é acoplado a um outro veio que sai do redutor (figura 15), por intermédio de um acoplamento flexível (figura 16), um elo de ligação do veio à haste, uma chumaceira para apoiar o veio, e uma calha onde vai ser efectuada uma restrição do



Figura N.º15– *Veio transmissor.*

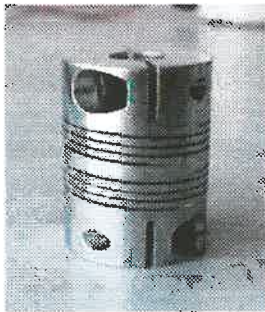


Figura N.º16
Acoplamento flexível

movimento horizontal, isto é, só existe movimento vertical na parte superior da perna. Este sistema garante o alinhamento da haste, mas em contrapartida verifica-se a existência de alguns atritos, que são atenuados com uma lubrificação eficiente.

Em relação ao elo de ligação, numa extremidade é fixo ao veio de maneira a que ele rode solidariamente com este, e na outra extremidade, foi colocado um rolamento para permitir a rotação da haste.

Os dois pontos de fixação da haste a toda estrutura, são feitos por dois parafusos que dão alguma flexibilidade nos comprimentos a que se chamou L1 e L2 na simulação.

Resta referir que nas pontas das pernas do robot que têm contacto directo com o chão, foram colocadas umas borrachas para que, deste modo se minimizem alguns deslizamentos.

Procedeu-se então à fase de montagem do robot. A primeira preocupação foi o alinhamento de todos os veios.

Foi então necessário fazer vários apoios, nomeadamente para os redutores e para os motores, tal como se pode constatar nas figuras 17, 18 e 19.

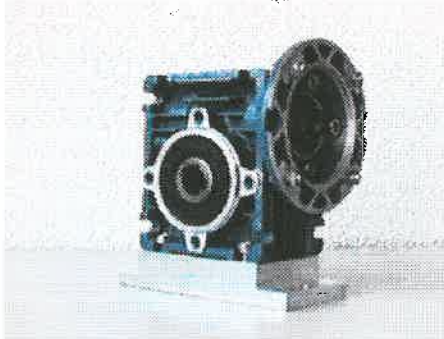


Figura N.º17 – Redutor e respectiva base.

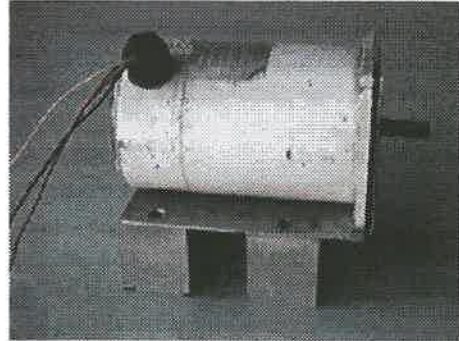


Figura N.º18 – Bases dos motores.

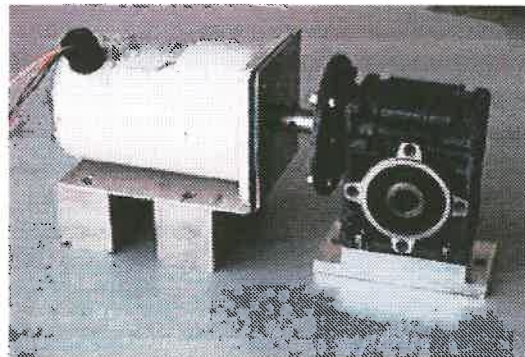


Figura N.º19 – Conjunto motor e reductor

Numa chapa única, foram distribuídos os vários componentes do robot, tendo sempre em atenção o centro de massa, para que o peso fosse distribuído de uma forma equilibrada. Assim sendo, os motores foram colocados em lados opostos, não só para equilibrar o já referido peso, mas também para terem o mesmo sentido de rotação. No espaço restante foram colocados o PLC e a unidade de potência.

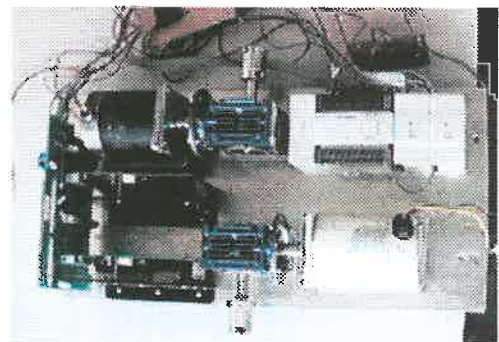


Figura N.º20 – Elementos funcionais do robot.

Com esta chapa, a colocação de todos os elementos dentro da estrutura foi uma tarefa muito mais simplificada, bem como a sua fixação à mesma.

Em relação à parte eléctrica, foi colocada uma calha à volta da estrutura do robot, por onde passam todos os cabos eléctricos.

No que diz respeito ao controlo, foi montado um painel onde se colocaram alguns botões que ligam e desligam motores, PLC, unidade de potência, ou seja, todo o sistema eléctrico. Neste painel

está também colocado um botão de emergência.

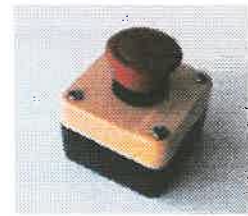


Figura N.º21 – Botão de emergência.

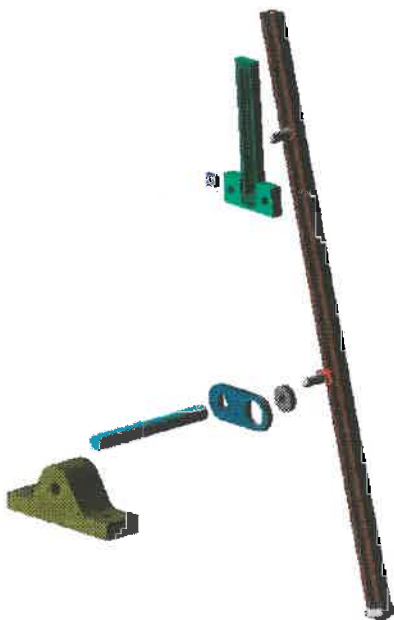


Figura N.º22 - Vista explodida

Apresenta-se na figura 22 uma vista explodida de todo o sistema de locomoção do robot. Conseguem-se visualizar a disposição e sequência lógica de montagem de uma perna, sendo todas as outras iguais.

8 - Lista de Material Utilizado

Após a descrição de alguns componentes do robot, foi elaborada a seguinte lista de todo o material utilizado na montagem do mesmo:

- 1 autómato FX_{2n} – 16MR – DR
- 2 módulos FX_{2n} – 1PG
- 2 drives de potência SD2 – Parker
- 1 unidade de potência
- 4 baterias 12V (6,5 A) e um carregador
- 2 caixas para fixação das baterias
- 1 regulador de tensão para 12V
- 2 motores passo a passo
- 4 suportes para os motores
- 2 bases de fixação para os motores
- 2 redutores 1:15
- 2 veios à saída do redutor
- 2 bases para os redutores
- 6 perfis 20x10 para as pernas
- 15 perfis 35x35 para a estrutura
- 8 tampas de perfil
- 24 cantos de união 35x35x35
- 2 acoplamentos flexíveis (φ12)
- 2 veios de acoplamento entre o motor e o redutor
- 6 rolamentos (φ8)
- 6 chumaceiras (φ12)
- 8 polias (40 dentes)
- 4 correias (750 mm)
- 4 calhas para condução de fio eléctrico
- 6 elos de transmissão de movimento
- 6 calhas para restringirem o movimento superior das pernas
- 6 tampas para as calhas
- 6 borrachas para os “pés” do robot
- 1 chapa de base para todos os componentes para fixação à estrutura
- 4 calhas para condução de fio eléctrico
- 6 botões on/off e um botão de Stop de emergência



9 - Análise Dinâmica do Robot

Depois de se definirem os vários componentes a utilizar e de se montar todo o conjunto procedeu-se à análise dinâmica, mas desta vez com o peso real do robot.

Então, para o peso de 40 Kg, o resultado é o seguinte:

Parâmetros:

$$L_1 = 0.20\text{mm};$$

$$L_2 = 0.20\text{mm};$$

$$r = 4\text{mm};$$

$$M = 40 \text{ kg};$$

- Sem planeamento

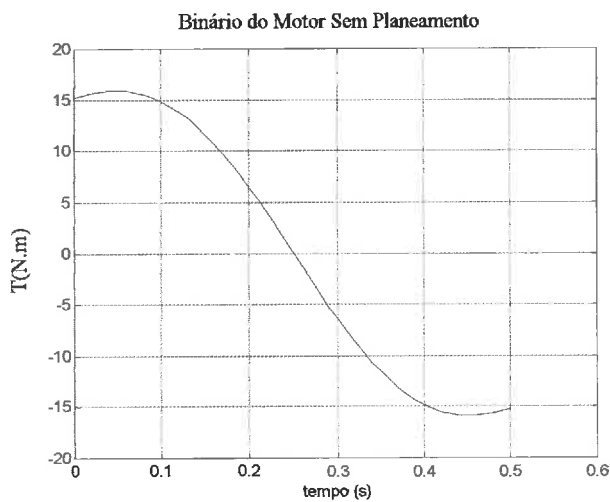


Gráfico N.º13 - Binário do motor para o movimento com planeamento de trajectória

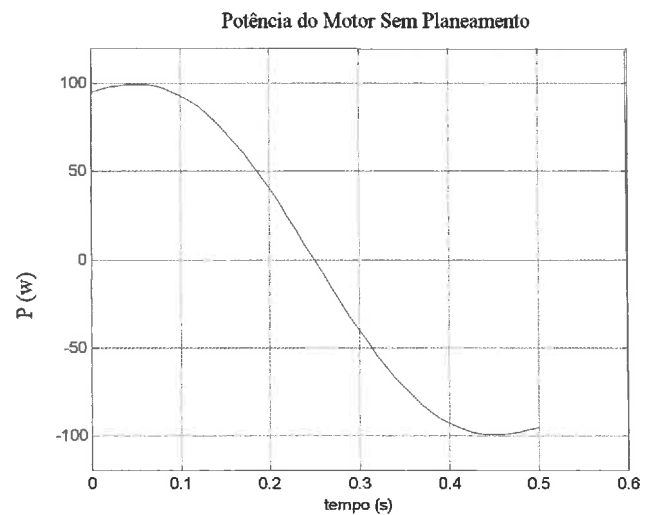


Gráfico N.º14 - Potência do motor para o movimento sem planeamento de trajectória



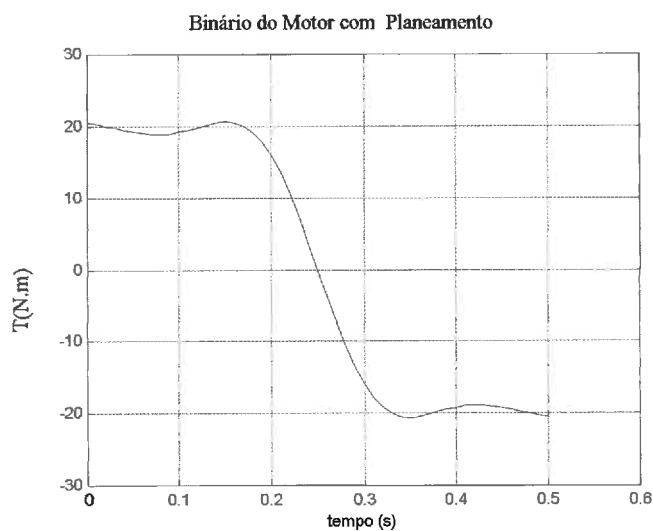
- Com planeamento

Gráfico N.º15 - Binário do motor para o movimento com planeamento de trajectória

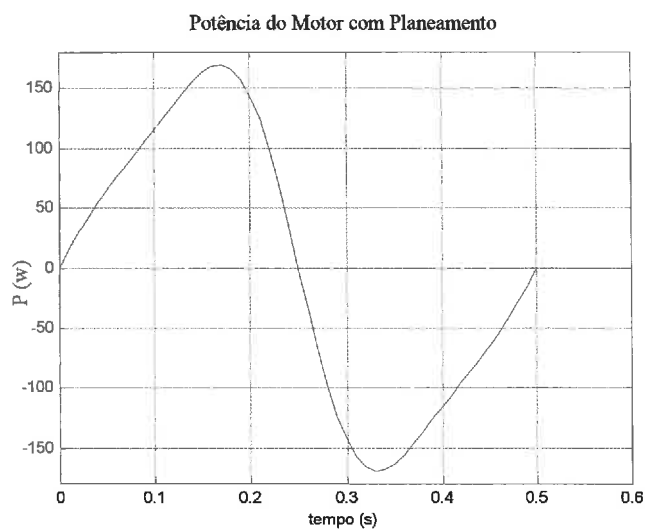


Gráfico N.º16 - Potência do motor para o movimento sem planeamento de trajectória

Analisando estes gráficos, quando comparados com os obtidos para o mesmo conjunto de parâmetros dimensionais, representados anteriormente na fase de estudo, verifica-se que ocorreu um aumento bastante significativo nos valores máximos de binário e potência devido ao aumento do peso do robot.

Segundo o estudo efectuado, seria necessário um binário médio da ordem dos 10Nm para a locomoção do robot, mas verificou-se insuficiente, exigindo-se um binário médio de 15Nm.



10 - Máquinas Utilizadas

Para o fabrico de algumas das peças que constituem o robot, nomeadamente todos os apoios, suportes, veios de acoplamento, etc, utilizaram-se as máquinas em baixo ilustradas, as quais se encontram no Laboratório de Tecnologia Mecânica na Secção Autónoma de Engenharia Mecânica.



Figura N.º23 - Furadora vertical de bancada

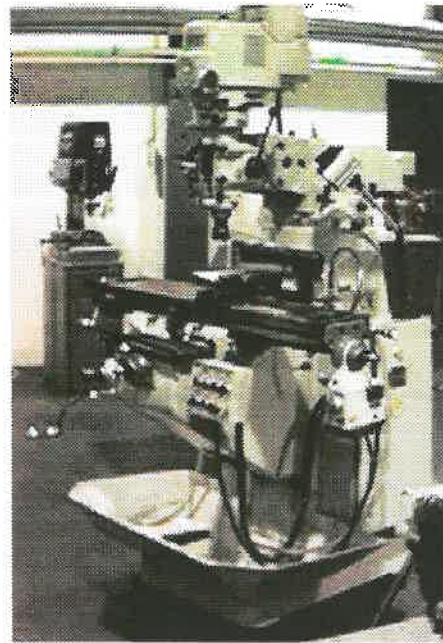


Figura N.º24 – Fresadora ferramenteira

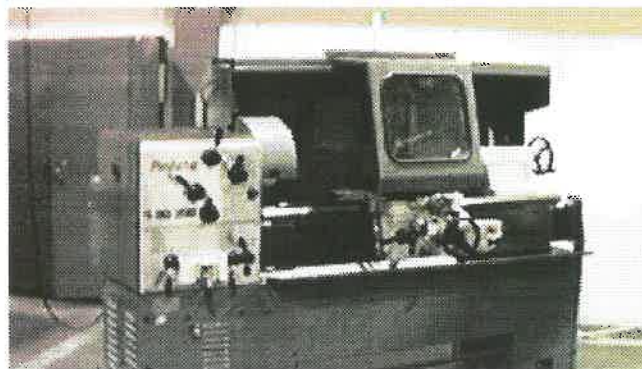


Figura N.º25 - Torno paralelo

11 - Programa Autómato

O programa autómato, apresentado no anexo 3, é desenvolvido no programa da Mitsubishi, Melsec Medoc plus, e é o responsável pelo movimento, visto que vai determinar a frequência dos impulsos que são enviados aos motores do robot, que uma vez alimentado pelas baterias se torna completamente autónomo.

Esta programação encarrega-se de efectuar o movimento com planeamento de trajectória, já definido anteriormente, isto é, baixa velocidade antes do embate da perna do robot no solo, e alta velocidade quando esta roda livre.

É através deste programa que se efectua todo o controlo do robot, nomeadamente o seu sentido de movimento e a sua paragem. Este programa prevê a rotação do robot em torno de si próprio, conseguida com a troca do sentido de rotação de um dos motores.

Este tipo de controlo é feito recorrendo a interruptores on/off e a um botão de emergência.

Para não se perder o sincronismo entre os dois motores, e conseqüentemente o deslocamento do robot, um só funciona se o outro também o fizer, não sendo possível que um motor esteja em rotação individualmente.



12 - Análise do Trabalho e Conclusões

Analisando os objectivos iniciais deste projecto e comparando com o resultado final, foram cumpridos, a modelização cinemática e dinâmica ainda no decorrer do primeiro semestre, a concepção da estrutura e dos elementos envolvidos na transmissão do movimento, toda a parte eléctrica, o uso de motores e unidades de controlo e potência e o desenvolvimento do programa automático para a sua locomoção, onde foram implementados todos os comandos externos que orientam o seu movimento. O que ficou por cumprir foi somente o uso dos sensores, o que impede desde já o desvio de obstáculos no caminho do robot.

Ao longo deste projecto foram surgindo várias dificuldades, onde umas se foram resolvendo, algumas contornando e outras sem solução temporariamente viável.

É sobre essas mesmas dificuldades que se vai falar um pouco.

Logo no início do trabalho, existiram alguns problemas na definição das equações matemáticas, uma vez que era necessário relacionar dois ângulos. Acusando um pouco o peso da responsabilidade, visto que, uma vez definidas estas equações serviram de suporte para todo o estudo deste projecto, a sua dedução foi algo demorada. Não foi muito complicado chegar às equações de posição (x, y) , mas sim nas derivadas das mesmas, devido à extensão com que ficava a equação da velocidade, e conseqüentemente a equação da aceleração. Recorrendo ao Matlab, foi possível resolver este passo que se previa algo demorado e ganhar algum tempo nesta fase do projecto.

Fez-se o estudo da dinâmica, mas sem qualquer problema de maior, visto as equações estarem bem definidas.

Seguiu-se então um levantamento de todos os componentes do robot com a finalidade de se avaliar o peso total, para se analisar a potência e binários que estariam envolvidos e para fazer um planeamento da execução e montagem.



Deste modo conseguia-se prever se os motores disponíveis tinham as especificações necessárias para o correcto funcionamento.

Recorrendo aos redutores, verificou-se que em princípio estes motores dariam conta da resposta ao movimento pretendido.

Um dos motores, motor I, suspeita-se que tenha pelo menos um enrolamento em contacto com o seu chassis, uma vez que este, conduz corrente através do mesmo.

Surgiu a dúvida em relação à unidade de controlo, se aguentaria com a potência exigida pelos motores.

Foi aqui que apareceu a maior dificuldade deste projecto. Depois de muito tempo perdido a estudar esta unidade, a tentar perceber o seu funcionamento, visto que não se dispõe de qualquer documentação acerca da mesma, pode-se dizer que há alguma frustração nestes resultados finais, pois se o robot não anda, a esta unidade se deve.

Muitos testes foram feitos para tentar pôr em marcha o robot; foi montado e desmontado várias vezes para tentar perceber o porquê de tal anomalia, mas todas sem sucesso, pois a unidade de potência ligada ao gerador de impulsos funciona, ligada ao PLC, os pulsos não chegam aos motores. Está afastada a hipótese de ser um erro do programa automático ou do próprio funcionamento do PLC, pois este gera os impulsos correctamente até à unidade, não chegando qualquer impulso aos motores.

Na parte de programação, quando se desligavam os motores, era necessário garantir que os pulsos deixavam de ser enviados e incrementados. Se tal não acontecesse, quando se voltassem a ligar, devido à variação de velocidade, deixava de haver sincronismo entre a velocidade e a posição da perna do robot, já que, quando a perna deveria estar em movimento mais lento, poderia estar com uma velocidade rápida.

O botão de emergência, geralmente serve para cortar a corrente do sistema. Mas, neste caso poderia ser crítico que tal acontecesse, pois uma vez tirada a tensão aos motores, devido ao peso total do robot, perder-se-ia a posição das pernas, já que as seis hastes ficariam simultaneamente em contacto com o solo



Deste modo, está visto que este botão apenas corta o envio de impulsos do autómato para as unidades de potência.

Em relação ao peso final, o robot atingiu um peso de 40Kg superior ao que foi previsto inicialmente (25Kg).

As diversas causas que apontam para tal acontecimento são, entre outras, o peso dos redutores, pois são mais pesados do que aqueles pensados no início, as chumaceiras foram outro factor que levou ao seu aumento, uma vez que também são um pouco mais pesadas do que era esperado. As baterias poderão ter sido outra causa, pois só à conta destas são 8Kg.

As outras causas que levaram ao aumento de peso são os suportes para fixar alguns elementos, o número elevado de porcas e parafusos utilizado, e mesmo a própria estrutura que saiu um pouco mais pesada daquilo que se esperava

Se se comparar estas duas situações, em que uma não passava ainda de um projecto, e a outra é o resultado final desse mesmo projecto, é bem elucidativo a diferente distribuição dos vários componentes do robot. As causas para tal mudança, já foram de uma maneira geral todas referidas.

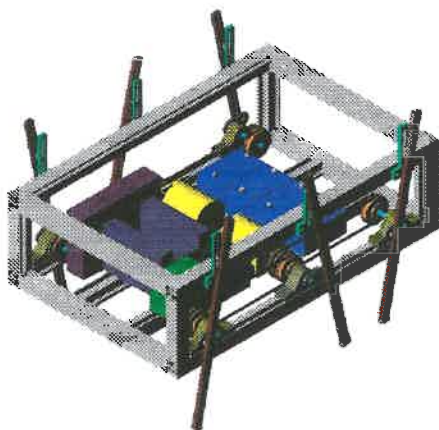


Figura N.º26 – Robot final previsto

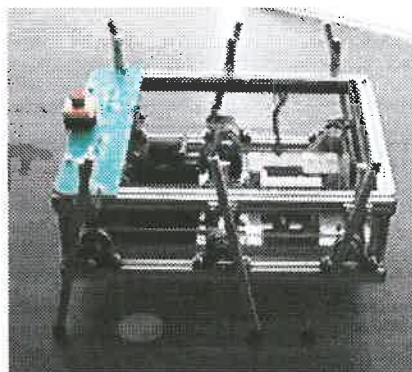


Figura N.º27 – Robot final

Com a componente prática deste projecto, foi necessário relembrar alguns conceitos já abordados em disciplinas de anos anteriores, conciliar tudo e montar. Devido à inexperiência, muitos erros de projecto passariam sem sequer se dar conta que eles existiam, não fosse esta parte prática deste projecto. Só assim foi possível passar pelos mais diversificados problemas e encontrar as melhores soluções para os superar e continuar em frente.



É importante referir aqui o elevado custo no fabrico de dois componentes essenciais que foram necessários fabricar fora do departamento, o elo de transmissão de movimento, e a calha de fixação da parte superior da haste, custo este que poderia ter sido empregue na aquisição de uma unidade de potência necessária ao funcionamento do robot, se o laboratório de tecnologia mecânica não tivesse tanto trabalho pendente. Quem sabe se o robot não estaria já a passear pelos corredores do Departamento de Engenharia Mecânica...



AneXO 1

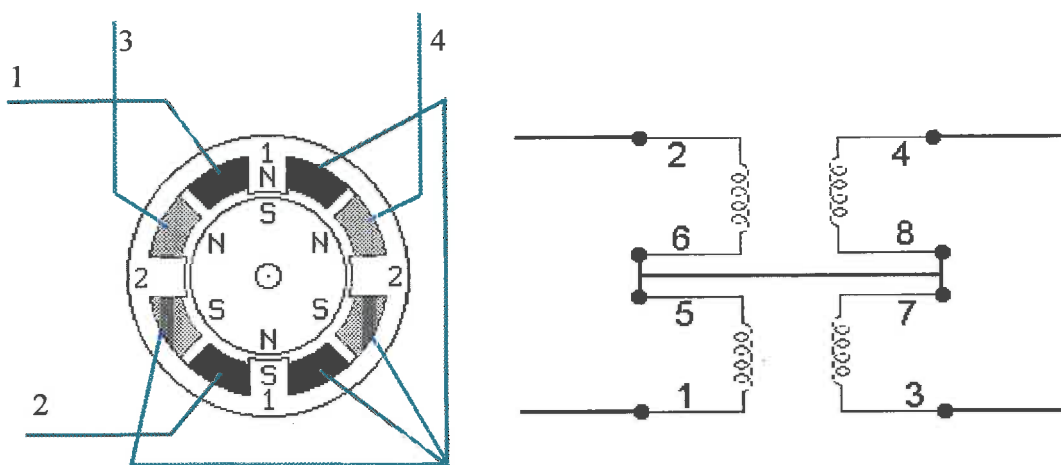
Esquemas Eléctricos



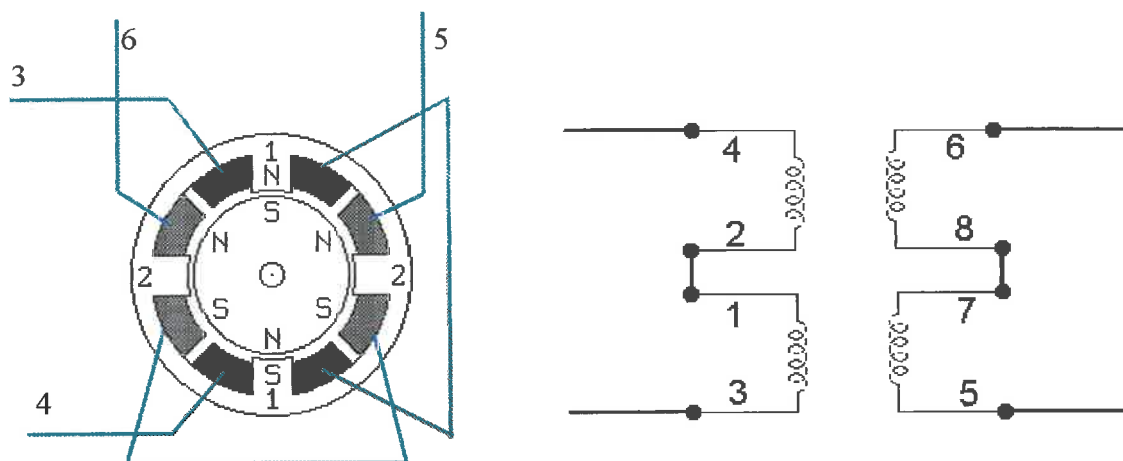
Esquema eléctrico dos motores

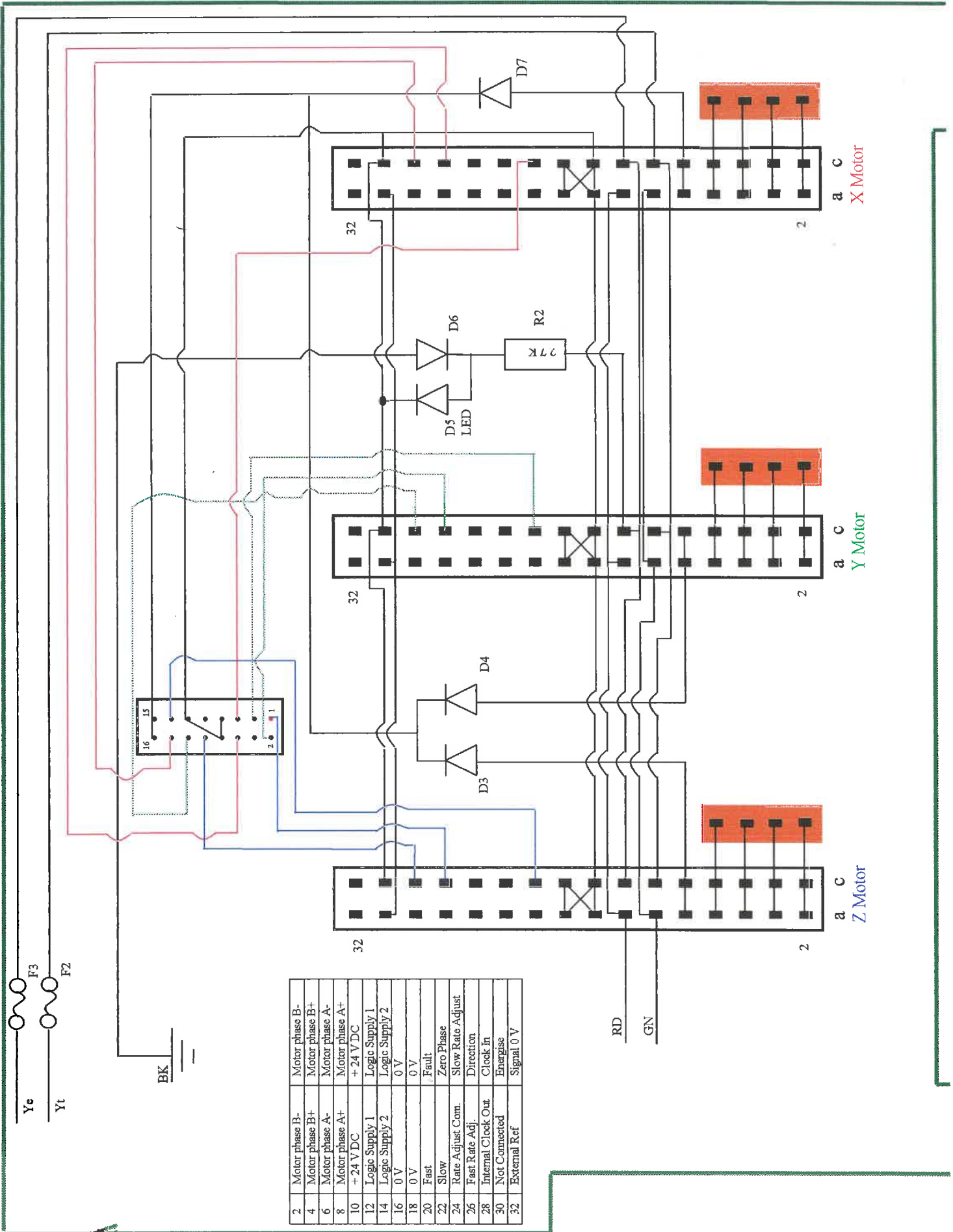
Aqui pretende-se fazer a ilustração de como são feitas as ligações eléctricas dos motores passo a passo. Se estas ligações não forem feitas correctamente, os impulsos não seguem uma ordem eficaz para que o motor rode de uma forma suave e contínua.

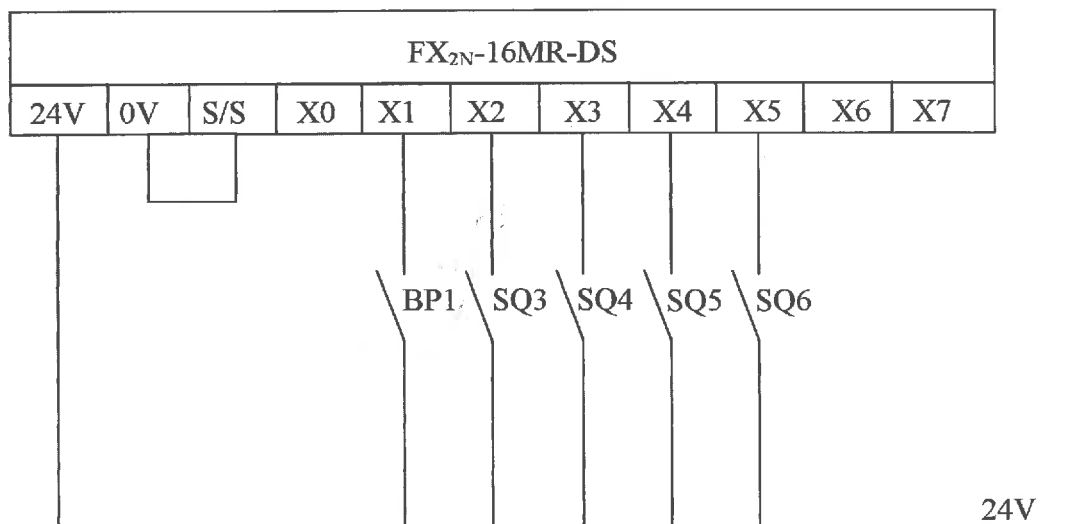
Então, para o motor I (motor branco), o esquema que se seguiu foi o seguinte:



Para o motor II (motor preto), por ser diferente, vem o seguinte esquema:





Esquema eléctrico das entradas do PLC

BP1 - Botão de paragem dos impulsos para os motores → *sem +abreco?*

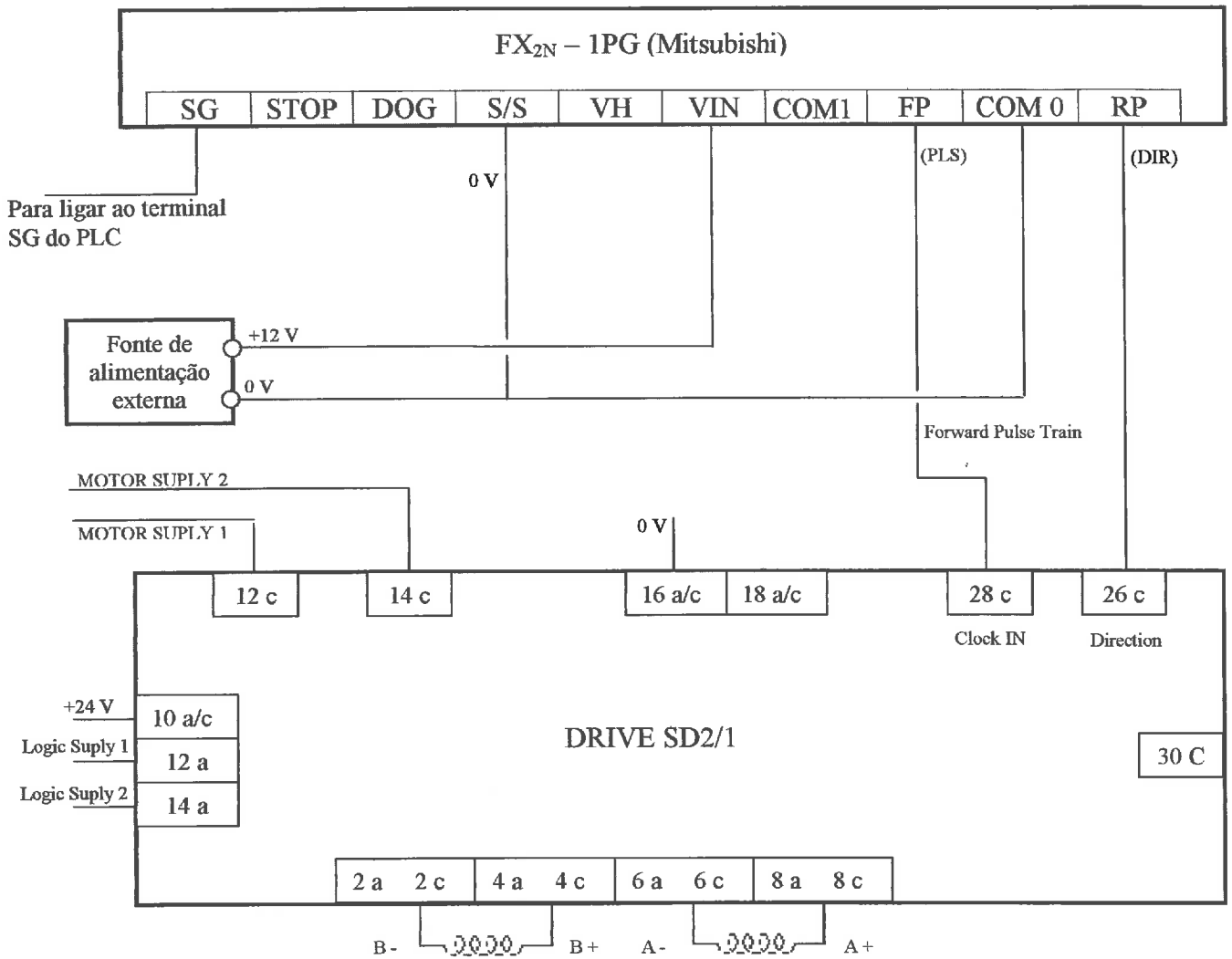
SQ3 - Motor I - velocidade_RP (reverse pulse)

SQ4 - Motor I - velocidade_FP (forward pulse)

SQ5 - Motor II - velocidade_RP (reverse pulse)

SQ6 - Motor II - velocidade_FP (forward pulse)

Esquema de ligações das cartas à unidade de Potência



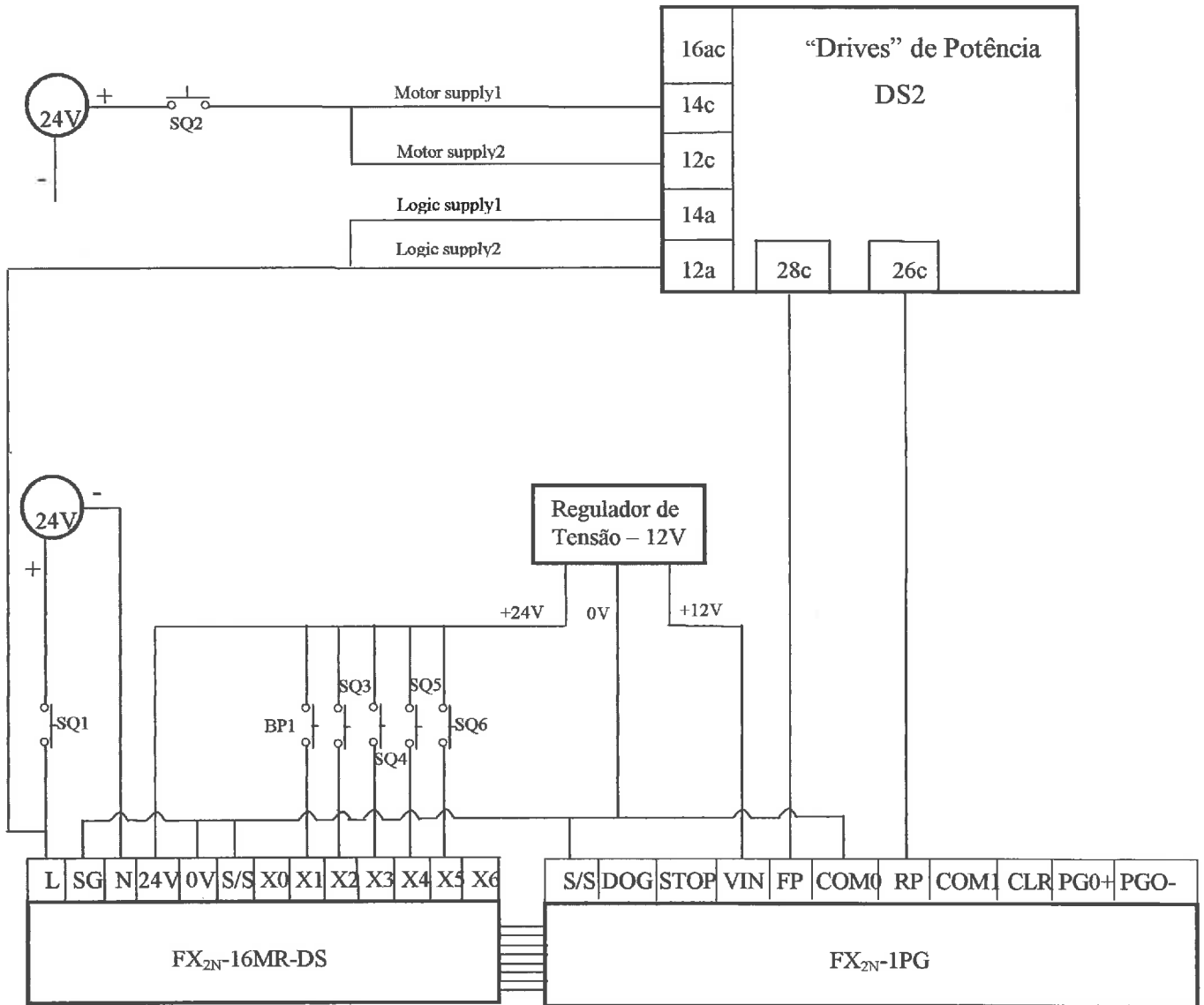
SG – Signal Ground

STOP – Paragem impulsos para o motor (24V)

DOG – Ponto de Referência



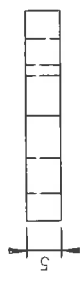
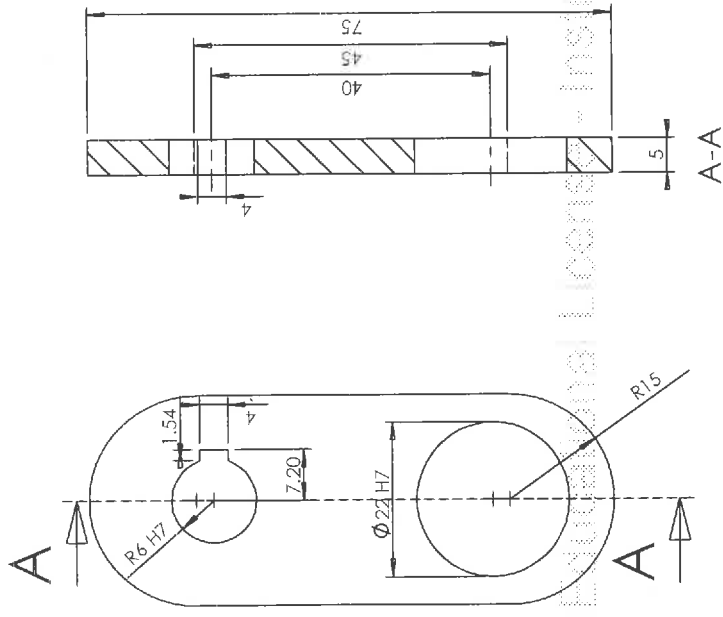
Esquema eléctrico do robot hexápode



Anexo 2

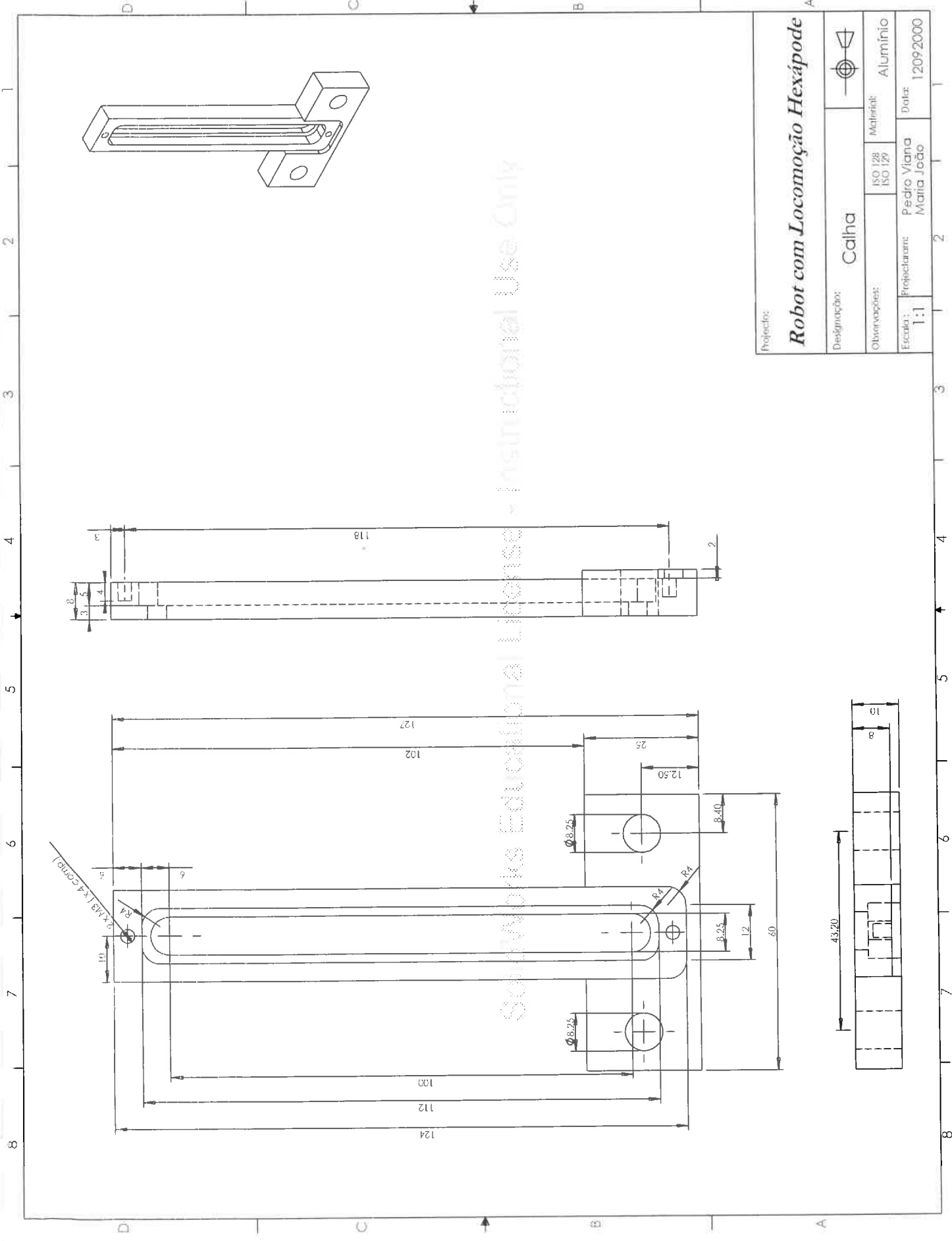
Desenhos de Definição





Projecto:		Robot com Locomoção Hexápode	
Designação:		Elo de transmissão de movimento	
Observações:		ISO 128 ISO 129	Material: Alumínio
Escola:	Elaboração:	Data:	
1:1	Pedro Viana Maria João	12092000	

SolidWorks Educational License - Instructional Use Only

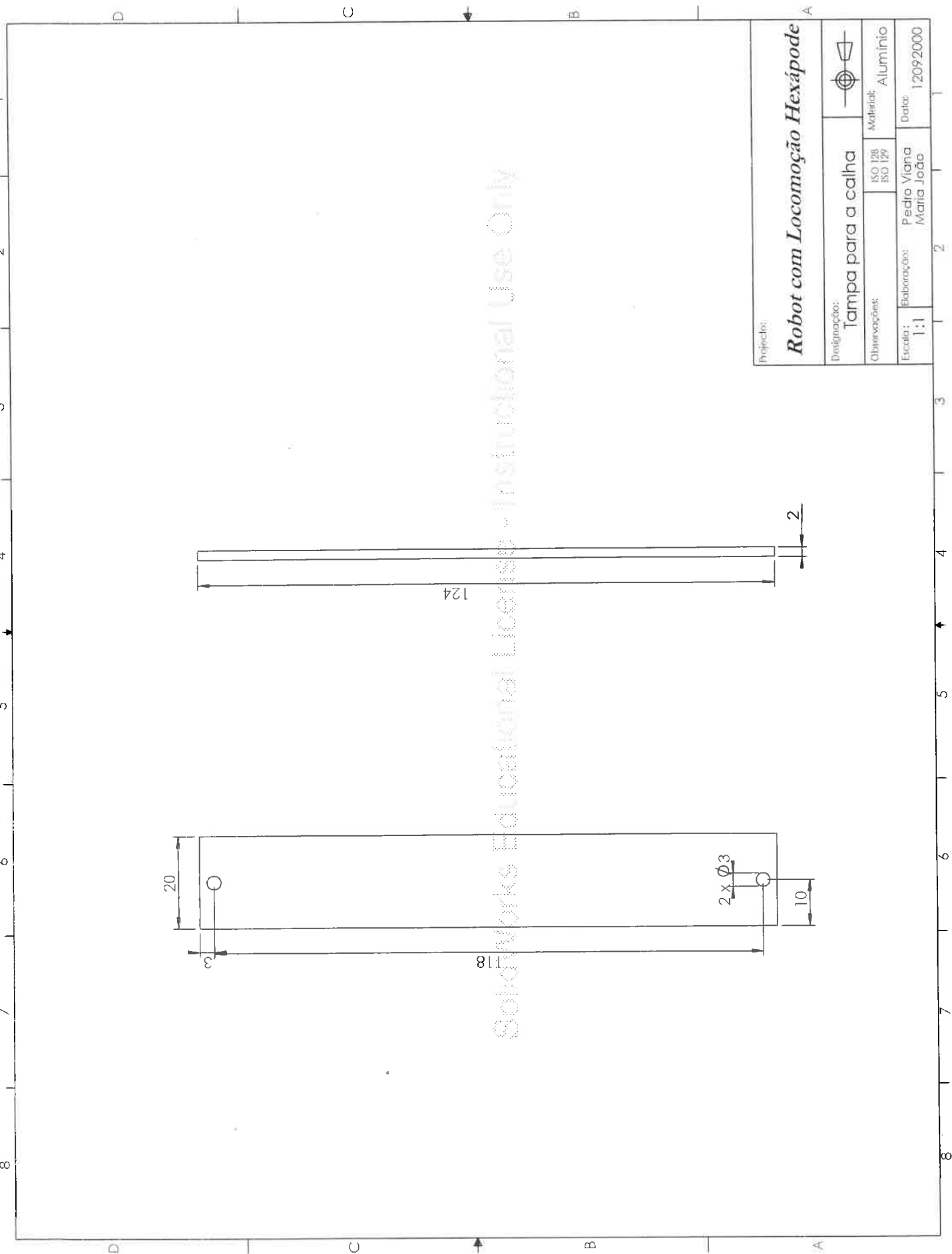


Projeto:

Robot com Locomoção Hexápode

Designação:	Calha	Material:	Alumínio
Observações:	ISO 128 ISO 129	Projecante:	Pedro Viana Matia João
Escala:	1:1	Data:	12092000

Source: <http://www.cadcamforum.com>

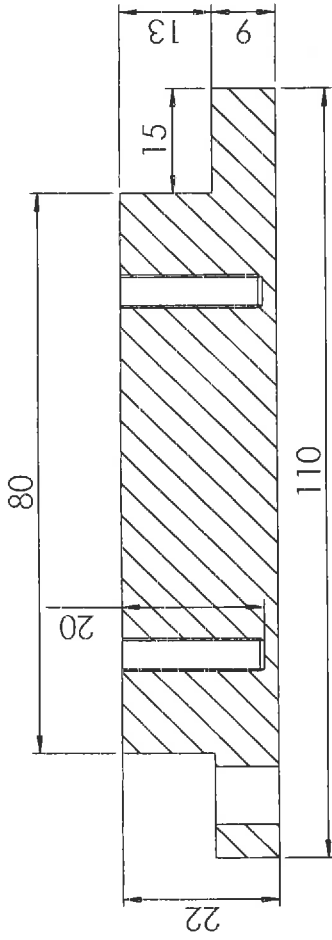


Projeto:

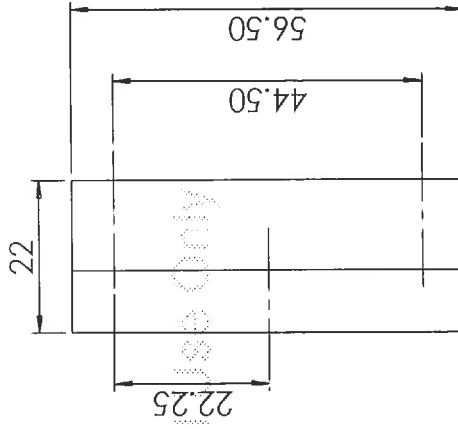
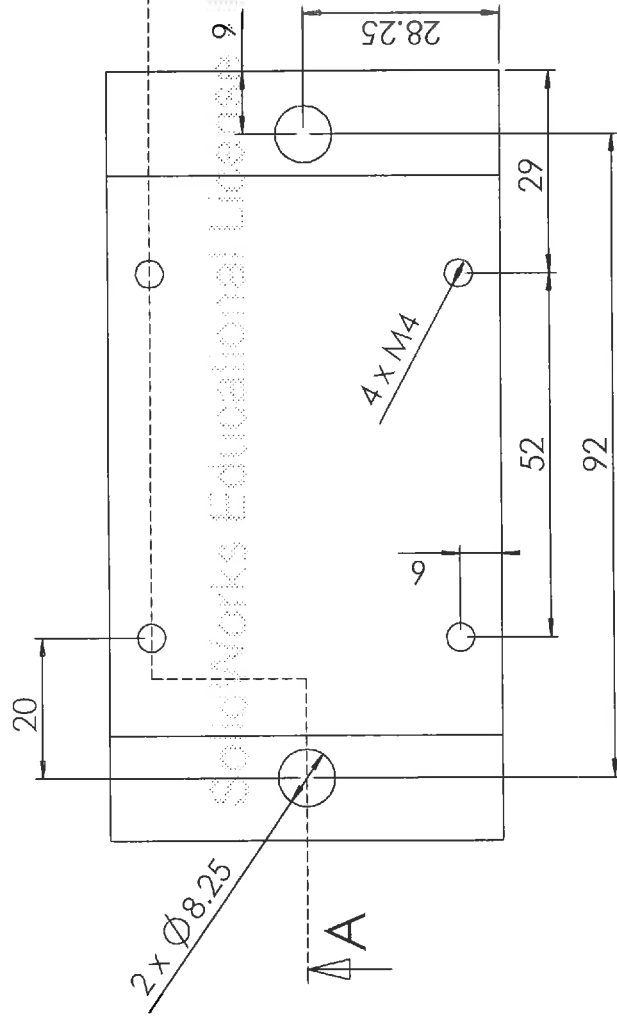
Robot com Locomoção Hexápode

Designação:	Tampa para a calha			
Observações:	ISO 128 ISO 129	Material:	Alumínio	
Escala:	1:1	Elaboração:	Pedro Viana Maria João	Data:
				12092000

SolidWorks Educational License - Instructional Use Only



A-A



Projeto:

Robot com Locomoção Hexápode

Designação: Apoio para os redutores



Observações:

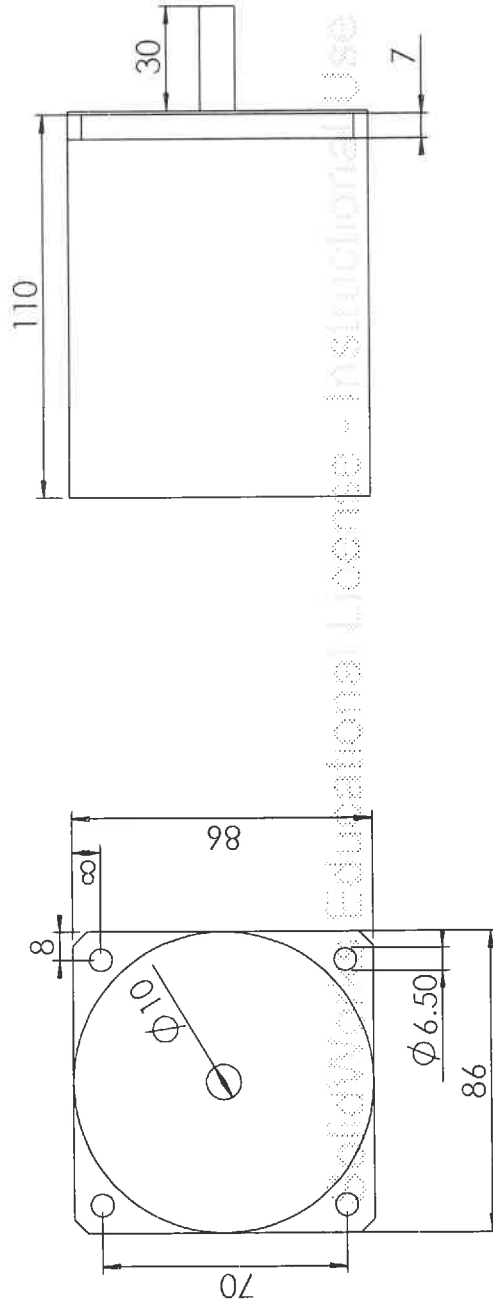
ISO 128
ISO 129

Material: Alumínio

Escala: 1:1

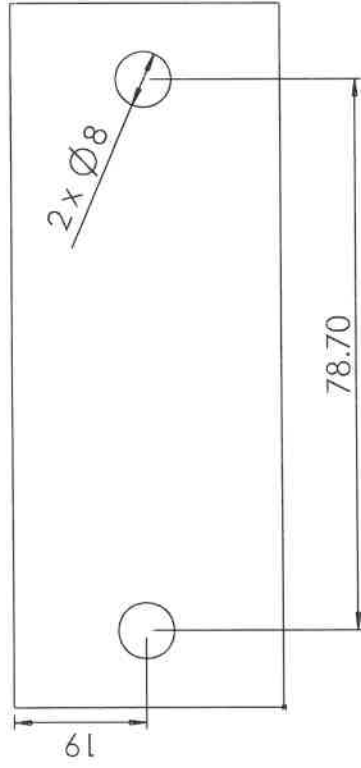
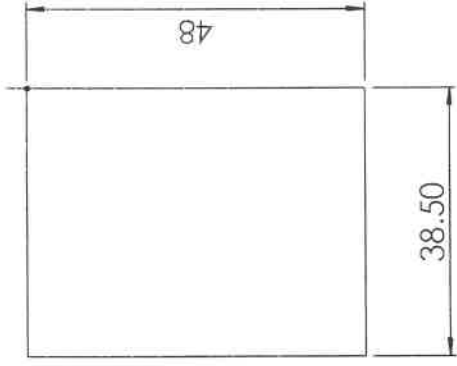
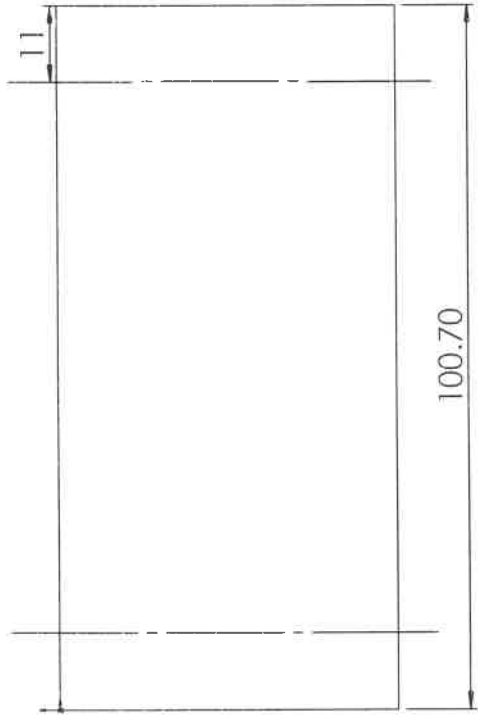
Elaboração: Pedro Viana
Maria João

Data: 12/09/2000



Projeto:		Robot com Locomoção Hexápode	
Designação:	Motor	Material:	
Observações:	ISO 128 ISO 129	Material:	
Escala:	1:1	Elaboração:	Pedro Viana Marta Jogo
		Data:	12/09/2000

Exclusivo para Uso Educativo - Institucional Use Only

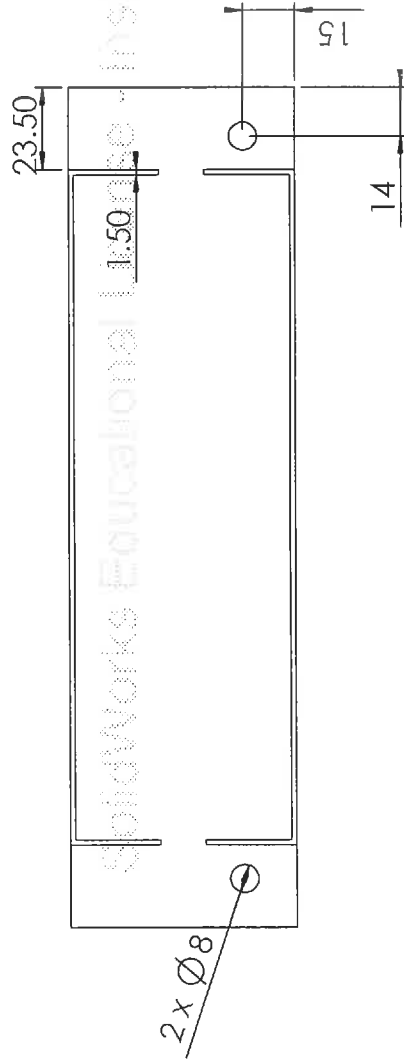
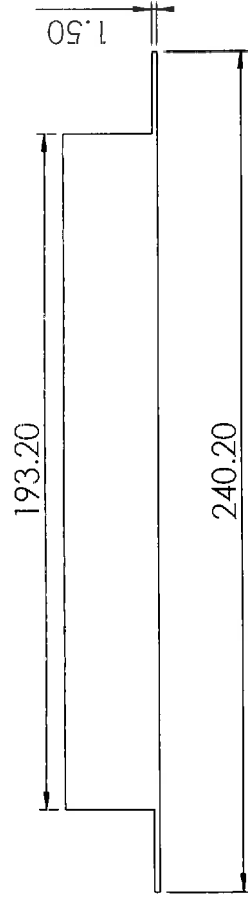
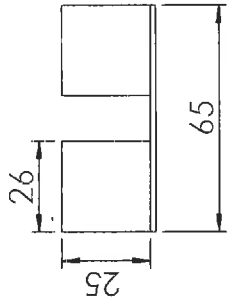


SolidWorks Educational License - Instructional Use Only

Projeto:

Robot com Locomoção Hexápode

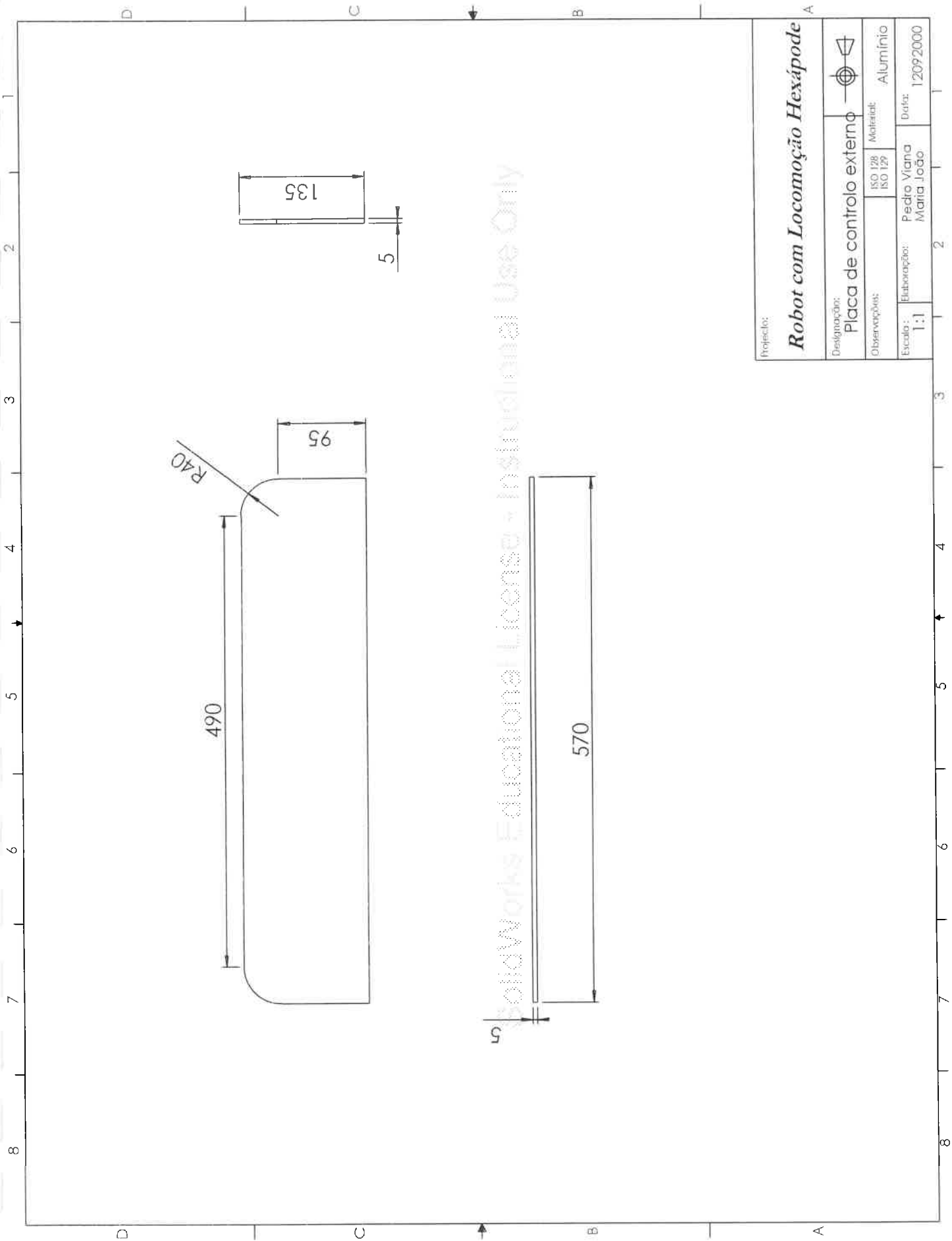
Destinação: Apoio para os motores		Material: Ufial	
Observações:		ISO 128: ISO 129	Data: 12/09/2000
Escala: 1:1	Elaboração: Pedro Viana Maria João		



Projeto:

Robot com Locomoção Hexápode

Designação:	Caixa para fixar baterias	Material:	Alumínio
Observações:	ISO 128 ISO 129	Elaboração:	Pedro Viana Maria João
Escala:	1:1	Data:	12/09/2000

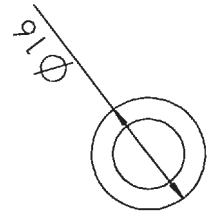
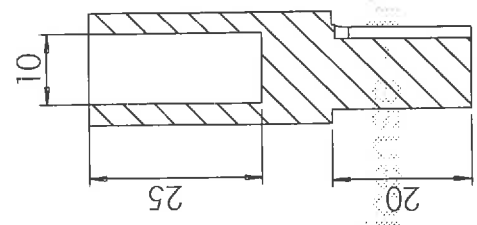
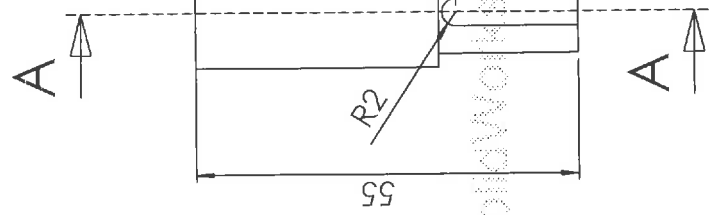
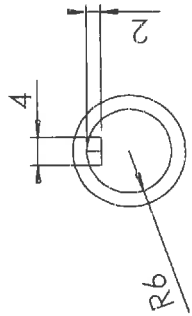


Projeto:

Robot com Locomoção Hexápode

Designação: Placa de controlo externo		Material: Alumínio	
Observações: ISO 128 ISO 129		Símbolo: 	
Escala: 1:1	Elaboração: Pedro Viana Maria João	Data: 12/09/2000	

SolidWorks Educational License - Instructional Use Only



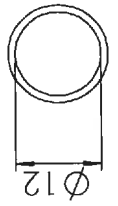
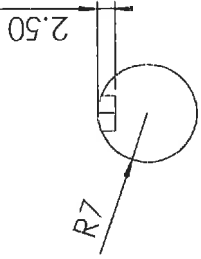
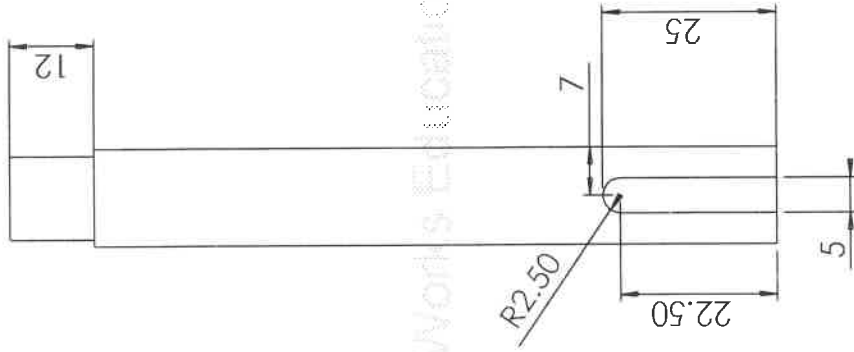
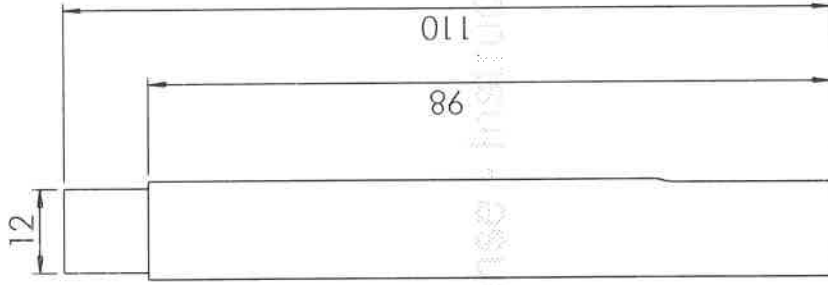
Projecto:			
Robot com Locomoção Hexápode			
Designação:		Veio de Acoplamento	
Observações:		ISO 128	Material: Aço
		ISO 179	
Escala:		Elaboração:	Data:
1:1		Pedro Viana Maria João	12/09/2000

SolidWorks Educational License - Instructional Use Only

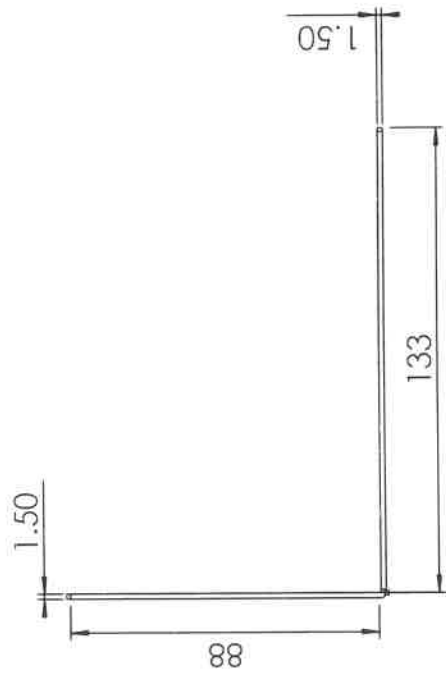
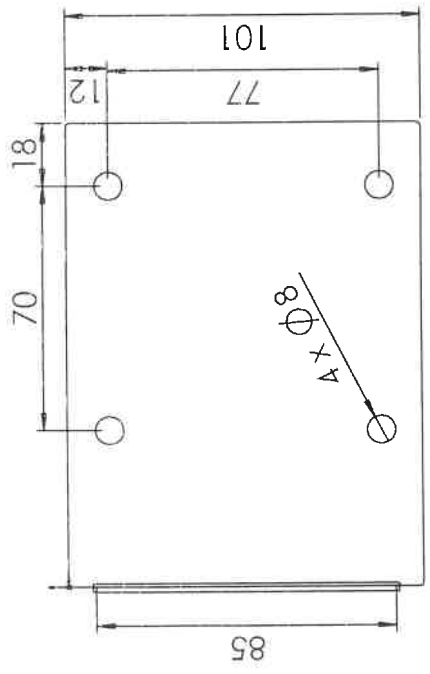
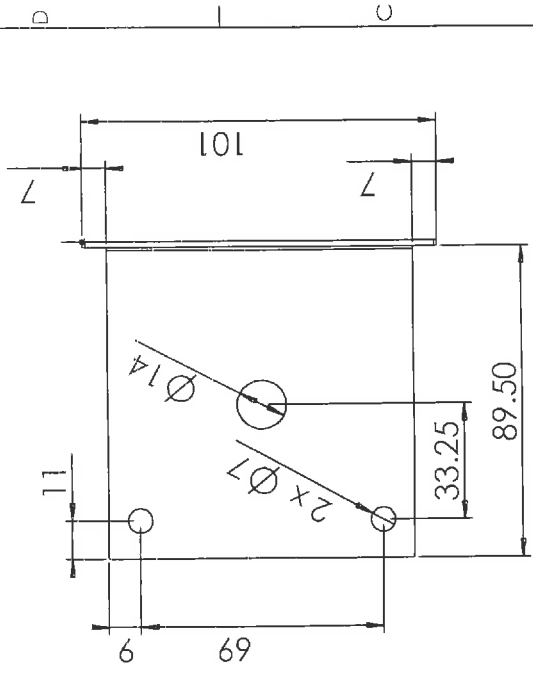
Projecto:

Robot com Locomoção Hexápode

Designação:	Viejo de transmissão de movimento	
Objetivos:	ISO 128 ISO 129	Material: AÇO
Elaboração:	Pedro Viana Maria João	Data: 12092000
Escala:	1:1	



SolidWorks Educational License - Institutional Use Only



SolidWorks Educational License - Instructional Use Only

Projecto:		Robot com Locomoção Hexápode	
Designação:	Fixação do motor	Material:	Alumínio
Observações:	ISO 128 ISO 129	Elaboração:	Pedro Viana Maria João
Escala:	1:1	Data:	12/09/2000

AneXo 3

Programa AutómatO



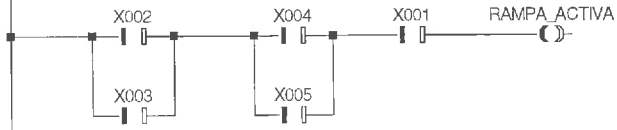


MITSUBISHI
ELECTRIC



	POU name	Comment
0	MOTOR_1	
1	VARIA_VELOCIDADE	
2	MOTOR_2	
3	VARIA_VELOCIDAD2	
4	CNTR_RAMPA	

Class	Identifier	Type	Initial	Comment
VAR_EXTERNAL	RAMPA_ACTIVIA	BOOL	FALSE	Início da variação da velocidade



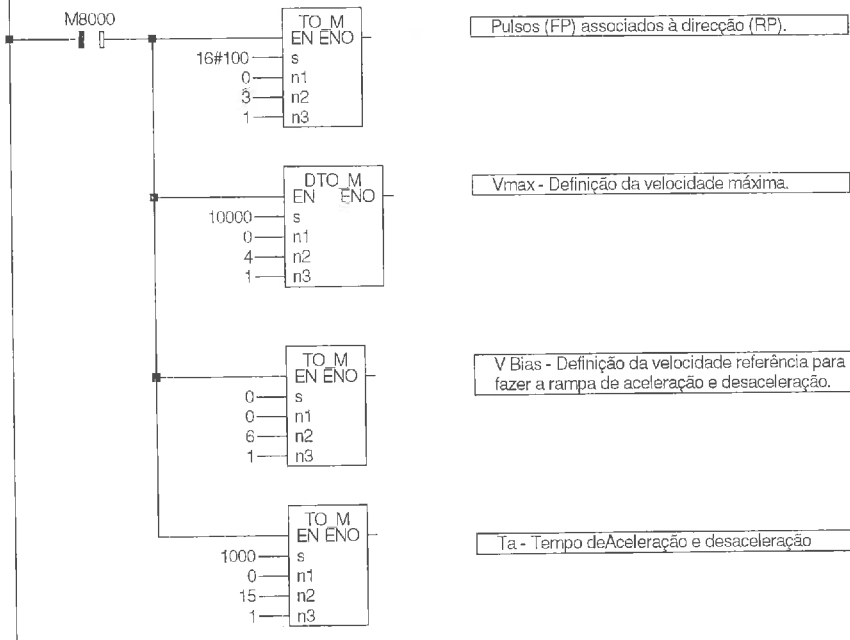
Só quando os dois motores estão ligados e enquanto o botão de STOP não for activado é que se efectua a variação de velocidade.



MITSUBISHI
ELECTRIC



	Class	Identifier	Type	Initial	Comment
0	VAR_EXTERNAL	VELOCIDADE_FP1	DINT	0	Velocidade do motor I no sentido positivo
1	VAR_EXTERNAL	VELOCIDADE_RP1	DINT	0	Velocidade do motor I no sentido negativo

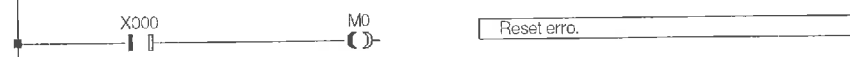


Pulsos (FP) associados à direcção (RP).

Vmax - Definição da velocidade máxima.

V Bias - Definição da velocidade referência para fazer a rampa de aceleração e desaceleração.

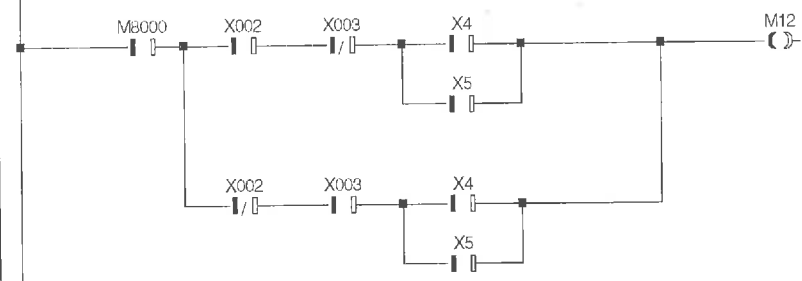
Ta - Tempo de Aceleração e desaceleração



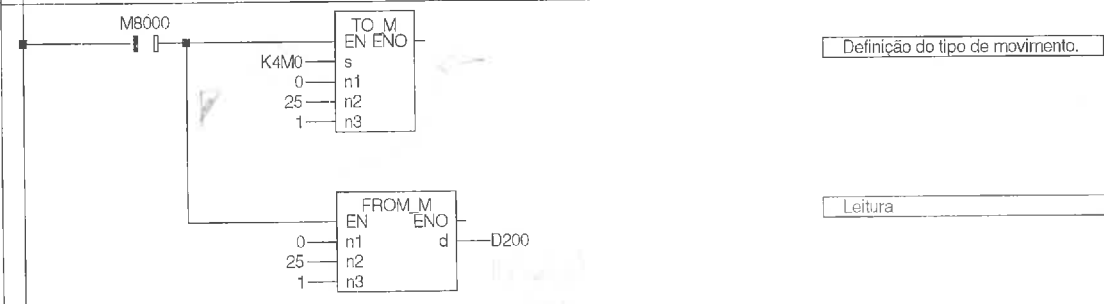
Reset erro.



STOP Impulsos - O botão de emergência está ligado à entrada X1 do PLC (normalmente ligado). Quando é accionado deixam de ser enviados impulsos para o motor.

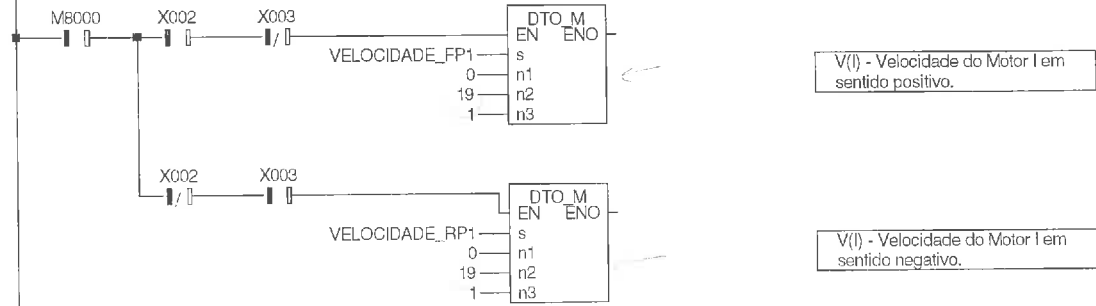


Variable speed operation start - Só existe rotação do motor se houver ordem para este rodar num sentido ou noutro e o motor estiver em funcionamento.



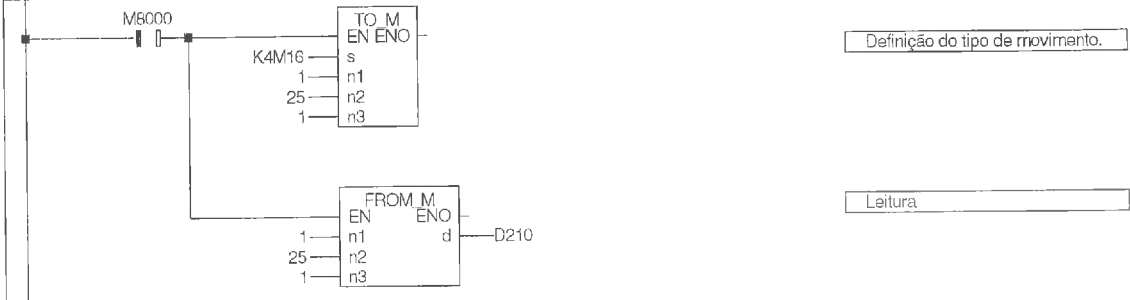
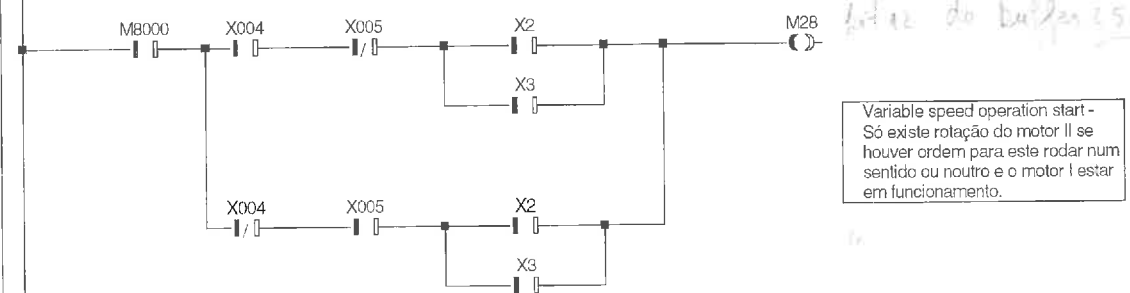
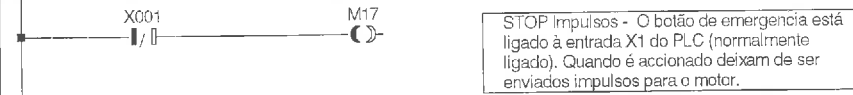
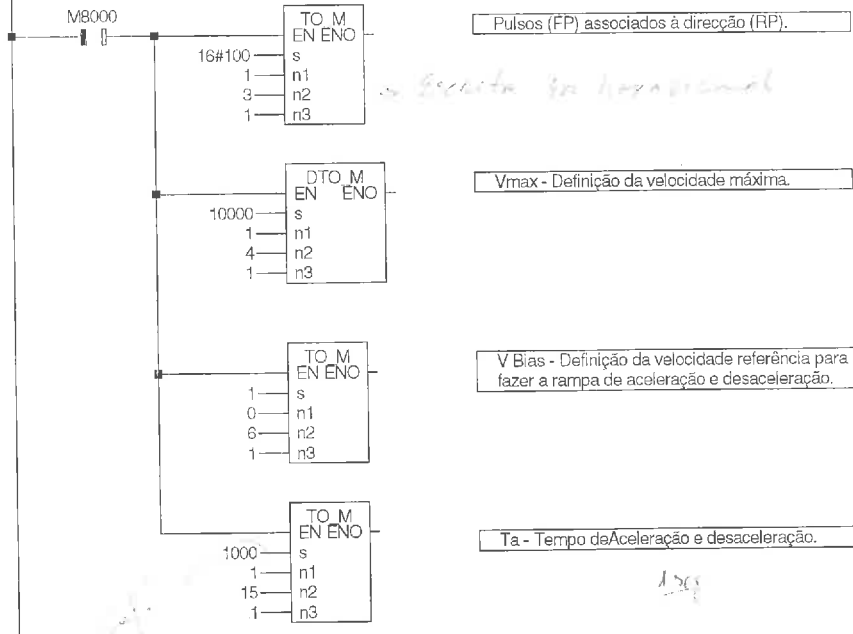
Definição do tipo de movimento.

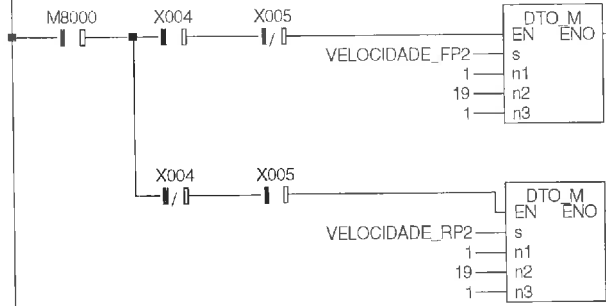
Leitura





	Class	Identifier	Type	Initial	Comment
C	VAR_EXTERNAL	VELOCIDADE_FP2	DINT	0	Velocidade do motor II no sentido positivo
1	VAR_EXTERNAL	VELOCIDADE_RP2	DINT	0	Velocidade do motor II no sentido negativo



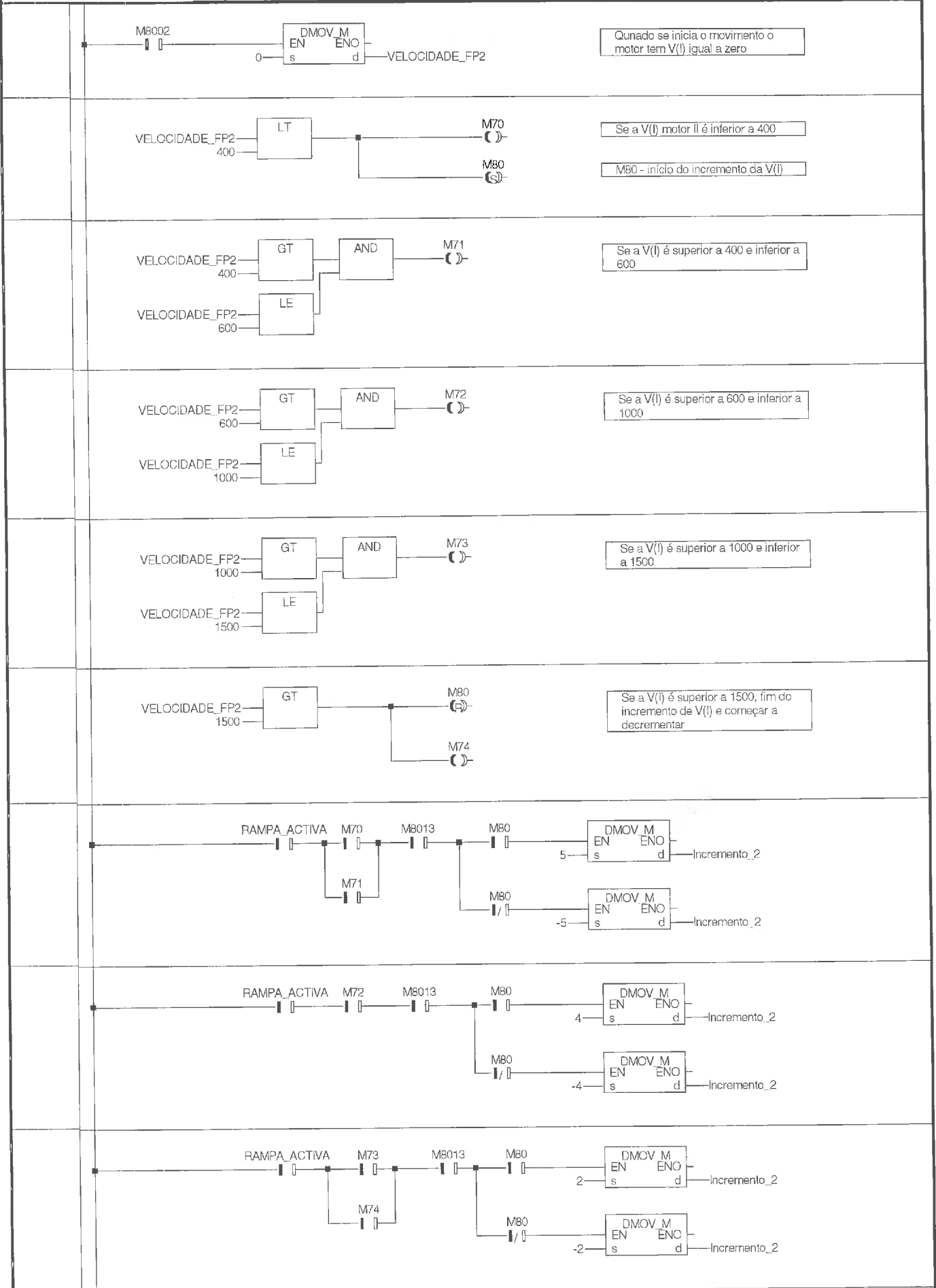


V(f) - Velocidade do Motor II em sentido positivo.

V(f) - Velocidade do Motor II em sentido negativo.



	Class	Identifier	Type	Initial	Comment
0	VAR_EXTERNAL	VELOCIDADE_FP2	DINT	0	Velocidade do motor II no sentido positivo
1	VAR	Incremento_2	DINT	0	Incremento da velocidade do motor II
2	VAR_EXTERNAL	VELOCIDADE_RP2	DINT	0	Velocidade do motor II no sentido negativo
3	VAR_EXTERNAL	RAMPA_ACTIVIA	BOOL	FALSE	Inicio da variaçao da velocidade



Quando se inicia o movimento o motor tem V(l) igual a zero

Se a V(l) motor l é inferior a 400

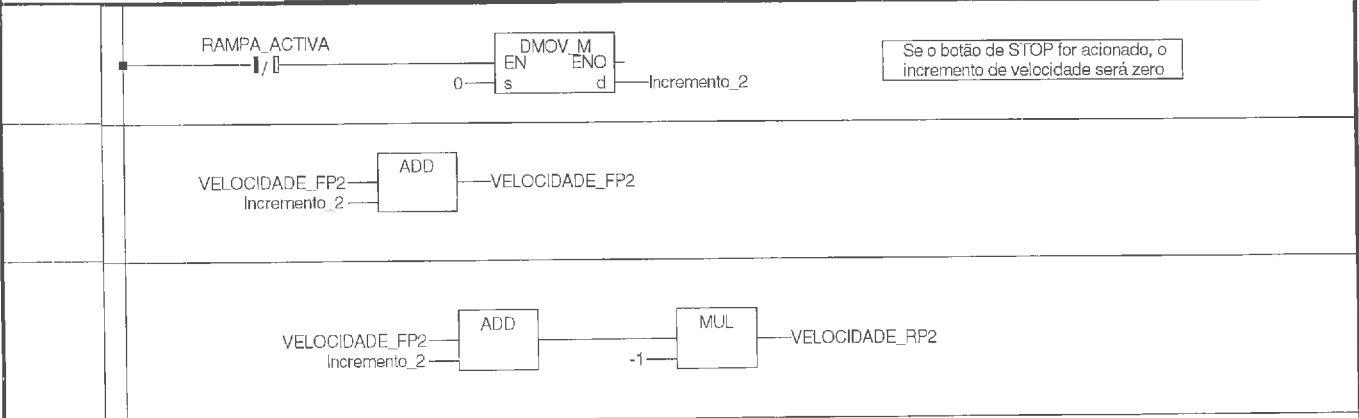
M80 - inicio do incremento da V(l)

Se a V(l) é superior a 400 e inferior a 600

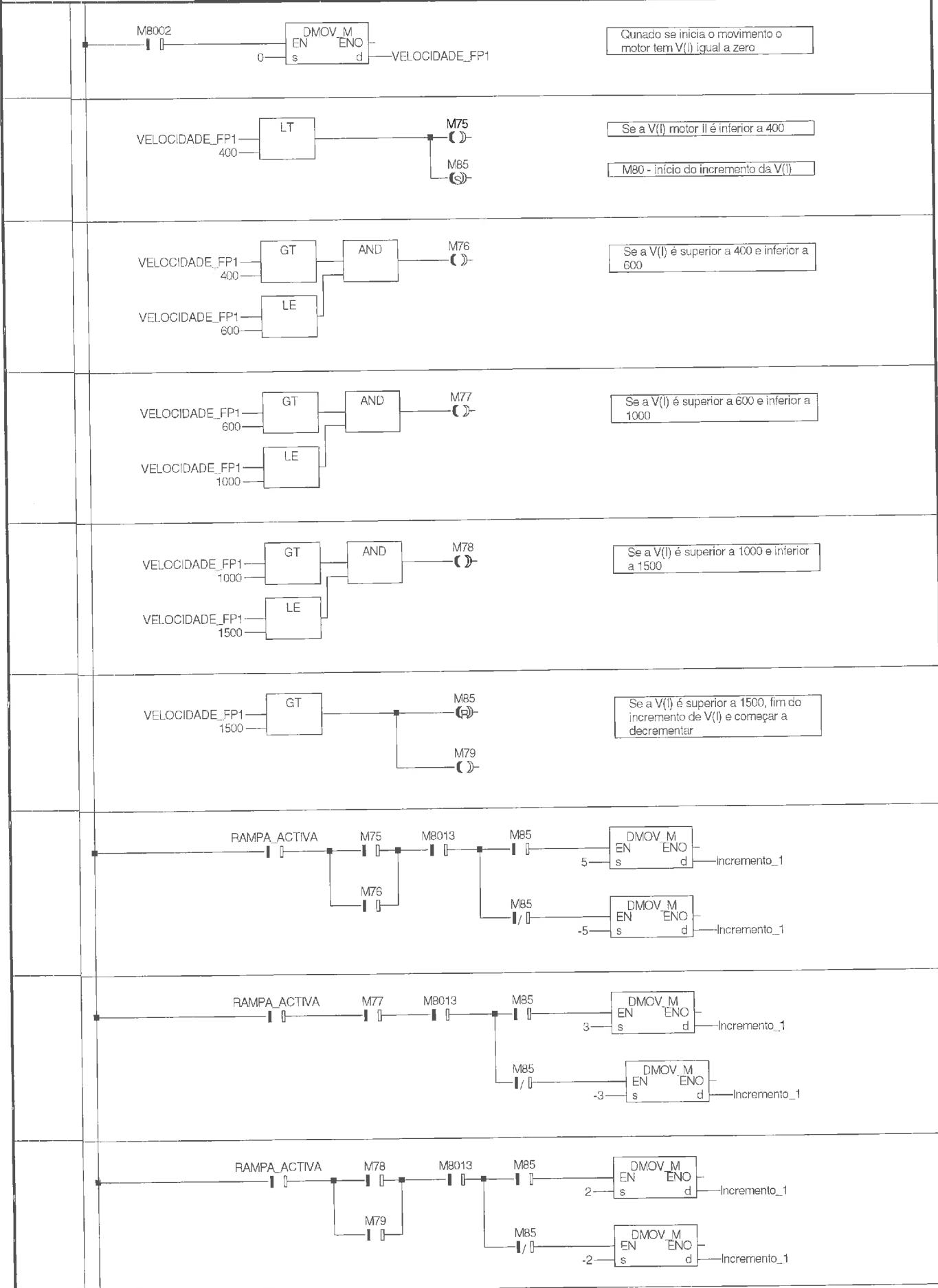
Se a V(l) é superior a 600 e inferior a 1000

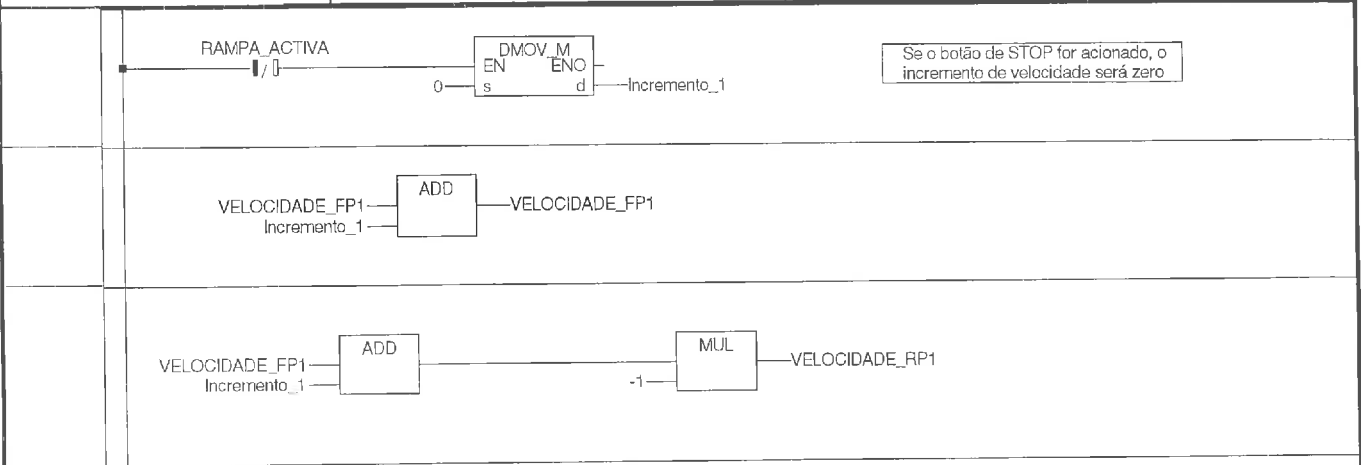
Se a V(l) é superior a 1000 e inferior a 1500

Se a V(l) é superior a 1500, fim do incremento de V(l) e começar a decrementar



	Class	Identifier	Type	Initial	Comment
0	VAR_EXTERNAL	VELOCIDADE_FP1	DINT	0	Velocidade do motor I no sentido positivo
1	VAR	incremento_1	DINT	0	Incremento da velocidade do motor I
2	VAR_EXTERNAL	VELOCIDADE_RP1	DINT	0	Velocidade do motor I no sentido negativo
3	VAR_EXTERNAL	RAMPA_ACTIVIA	BOOL	FALSE	Inicio da variaçao da velocidade





	Class	Autoextern	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial	Comment
0	VAR_GLOBAL		VELOCIDADE_FP1	D105	%MD0.105	DINT	0	Velocidade do motor I no sentido positivo
1	VAR_GLOBAL		VELOCIDADE_FP2	D107	%MD0.107	DINT	0	Velocidade do motor II no sentido positivo
2	VAR_GLOBAL		VELOCIDADE_RP2	D109	%MD0.109	DINT	0	Velocidade do motor II no sentido negativo
3	VAR_GLOBAL		VELOCIDADE_RP1	D111	%MD0.111	DINT	0	Velocidade do motor I no sentido negativo
4	VAR_GLOBAL		RAMPA_ACTIVIA	M90	%MX0.90	BOOL	FALSE	Início da variação da velocidade



Remark

VELOCIDADE_FP1:
defined in Global Variable List as VAR_GLOBAL
used in MOTOR_1 [PRG] Header
used in VARIA_VELOCIDADE [PRG] Header

VELOCIDADE_FP2:
defined in Global Variable List as VAR_GLOBAL
used in MOTOR_2 [PRG] Header
used in VARIA_VELOCIDAD2 [PRG] Header

VELOCIDADE_RP2:
defined in Global Variable List as VAR_GLOBAL
used in MOTOR_2 [PRG] Header
used in VARIA_VELOCIDAD2 [PRG] Header

VELOCIDADE_RP1:
defined in Global Variable List as VAR_GLOBAL
used in MOTOR_1 [PRG] Header
used in VARIA_VELOCIDADE [PRG] Header

RAMPA_ACTIVA:
defined in Global Variable List as VAR_GLOBAL
used in CNTR_RAMPA [PRG] Header
used in VARIA_VELOCIDAD2 [PRG] Header
used in VARIA_VELOCIDADE [PRG] Header



CPU Family & Type

CPU Family FX
CPU Type FX2N / FX2NC

Memory Param

Main Sequence 8 KSteps
Sub Sequence 0 KSteps
File Register 0 KPoints
Comment Space 0 KPoints
Main Micro 0 KByte
Sub Micro 0 KByte
Sampling Trace No

Timer & Counter

Standard 100 ms 0
Standard 10 ms 200
Standard Retentive 246
Extended 100 ms
Extended 10 ms
Extended Retentive
Watchdog 20 * 10 ms

MAIN_LD (Prio = 31, Event = TRUE)	1
CNTR_RAMPA [PRG] Header	2
CNTR_RAMPA [PRG] Body [LD]	3
MOTOR_1 [PRG] Header	4
MOTOR_1 [PRG] Body [LD]	5
MOTOR_2 [PRG] Header	7
MOTOR_2 [PRG] Body [LD]	8
VARIA_VELOCIDAD2 [PRG] Header	10
VARIA_VELOCIDAD2 [PRG] Body [LD]	11
VARIA_VELOCIDADE [PRG] Header	13
VARIA_VELOCIDADE [PRG] Body [LD]	14
Global Variable List	16
Cross Reference	18
PLC Config	19
Table of contents	20

AneXo 4

Programa em Matlab



Planeamento da trajectória

```

%-----
% nome do ficheiro : Planeamento
%-----
%Planeamento de trajectória - como a equação do movimento do robot
%depende da variação de teta, para garantir que não ocorrem embates
%no solo, teta vai variar segundo um polinómio do 3º grau
%-----

clc
clear all

%Variação do ângulo teta sem planeamento de trajectória

step = pi/10;
teta = 0:step:2*pi;
t = 0:1/20:1;           % 1 rotação demora 1 segundo (2pi)
Dteta(1:21) = 2*pi;    %Dteta - velocidade angular

%representação gráfica da velocidade angular

figure(1)
plot(t,teta,'b')
axis([0 1 0 7])
title('Variação do ângulo (Teta)'),xlabel('tempo (s)'),ylabel('teta(rad)')
grid on

%representação grafica da aceleração angular

figure(2)
plot(t,Dteta,'b')
axis([0 1 0 7])
title('Velocidade Angular'),xlabel('tempo (s)'),ylabel('Dteta(rad/s^2)')
grid on

%Planeamento de trajectória -----

%representação gráfica da velocidade angular

figure(3)
tt = 0:1/20:0.5;

tetal = 12*pi.*tt.*tt - 16*pi.*tt.*tt.*tt;
teta2 = pi + 12*pi.*tt.*tt - 16*pi.*tt.*tt.*tt;
tetal = tetal(1:10);
ang = [tetal teta2];

plot (t,ang,'r')
title('Variação do ângulo'),xlabel('tempo (t)'),ylabel('teta(s)')
axis([0 1 0 7])
%text(0.5,3.3,'Pi')
grid on

%representação gráfica da velocidade angular

```



```
figure(4)

Dteta1 = 24*pi.*tt - 48*pi.*tt.*tt;
Dteta2 = 24*pi.*tt - 48*pi.*tt.*tt;
Dteta1 = Dteta1(1:10);
Dang = [Dteta1 Dteta2];

plot (t,Dang,'r')
title('Velocidade Angular'),xlabel('tempo (t)'),ylabel('Dteta(s)')
axis([0 1 0 10])
grid on

%-----
```



Simulação de topo

```

%-----
% nome do ficheiro : movimento
%-----
% programa global - efectua-se a simulação do movimento das 6 pernas do
robot de topo.
%               Pode-se visualizar a simulação de 2 passos do robot com
a transição
%               das pernas que estão em contacto com o chão
representadas
%               graficamente por círculos vermelhos.
%-----
clc
clear all

global L1 L2 r  offset

N = 50;          %nº de iterações efectuadas na simulação
step = pi/N;
teta = 0:step:pi; %neste estudo não se considerou o planeamento
                  %de trajectória para a variação do ângulo

L1 = 20;        %comprimento da parte superior da perna
L2 = 20;        %comprimento da parte inferior da perna
r = 4;          %raio do elo
offset = 28;    %distância entre os veios transmissores de 2 pernas
                consecutivas

%1º PASSO - o ponto P3(x3,y3) da perna 1 serve de referência e tem
coordenadas iguais a zero
    x3 = 0;
    y3 = 0;

%calculo de todos os pontos críticos (P0,P1,P2,P3) para a perna 1 e 2

[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa2d(teta,
x3,y3);

    %representação gráfica da extremidade de todas as 6 pernas do robot

    [c]=grafico1(x32,y32,y3,x3,x02)

%2º PASSO - o ponto P3(x3,y3) da perna 1 serve de referência e tem coordenadas iguais ao
ultimo valor de x32 para que no instante de transição haja continuidade no movimento

    x3 = x32(N+1);
    y3 = 0;

    %calculo de todos os pontos críticos (P0,P1,P2,P3) para a perna 1 e 2

[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa2d(teta,
x3,y3);

```



%representação gráfica da extremidade de todas as 6 pernas do robot

[c]=grafico2(x32,y32,y3,x3,x02)



```
%-----  
% nome do ficheiro : pernafixa2d  
%-----  
%função para calcular as coordenadas de todos os pontos criticos de uma  
perna durante uma rotação completa.  
%-----  
  
function  
[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa2d(teta,  
x3,y3);  
  
global L1 L2 r raio offset  
  
[x0,y0,x1,y1,x2,y2,x3,y3] = fixa(teta,x3,y3);  
  
[x02,y02,x12,y12,x22,y22,x32,y32] = D_suspensa(teta,x0,y0);  
  
%-----
```




```
%-----  
% nome do ficheiro : fixa  
%-----  
%função para calcular as coordenadas dos pontos P1, P2, P3 de uma perna quando o ponto P3  
está em contacto com solo. O ponto P3 é o ponto de referencia.  
%-----  
  
function[x0,y0,x1,y1,x2,y2,x3,y3] = fixa(teta,x3,y3);  
global L1 L2 r raio  
  
%Beta é o ângulo que a haste faz com a vertical no decorrer do movimento. Este ângulo  
expressa-se em função de Teta.  
  
beta = -pi/2 + acos(-r*cos(teta)/L1);  
  
% P0 -----  
  
x0 = (L1+L2)*sin(beta);  
y0 = -r*sin(-teta) + L2*cos(beta);  
  
% P1 -----  
  
x1 = -r*cos(teta) + (L1+L2)*sin(beta);  
y1 = L2*cos(beta);  
  
% P2 -----  
  
y2 = (L1+L2)*cos(beta);  
x2 = (L1+L2)*sin(beta);  
  
% P3 -----  
x3= x3;  
y3= y3;  
%-----
```



```
%-----  
% nome do ficheiro : D_suspensa  
%-----  
%função para calcular as coordenadas dos pontos P1, P2, P3 de uma perna quando esta roda  
livremente. O ponto P0 é o ponto de referencia.  
%-----  
  
function[x02,y02,x12,y12,x22,y22,x32,y32] = D_suspensa(teta,x0,y0);  
global L1 L2 r offset  
  
%Beta é o ângulo que a haste faz com a vertical no decorrer do movimento. Este ângulo  
expressa-se em função de Teta.  
  
beta = -pi/2 + acos(-r*cos(teta)/L1);  
%P0 -----  
  
x02 = x0 + offset;  
y02 = y0;  
  
% P1 -----  
x12 = r*cos(teta) + x02;  
y12 = r*sin(teta) + y02;  
  
% P2 -----  
y22 = y12 + L1*cos(beta);  
x22 = x02;  
  
% P3 -----  
x32= x22 + (L1+L2)*sin(beta);  
y32= y22 - (L1+L2)*cos(beta);  
%-----
```



```

%-----
% nome do ficheiro : graficol
%-----
%representação gráfica da extremidade de todas as 6 pernas do robot, estrutura e centro de
massa no 1º passo
%-----

function[c]=graficol(x32,y32,y3,x3,x02)
global offset

Dim = length(x32);
%calcula as coordenadas dos pontos das pernas 3,4,5 e 6

y3 = 5;
y32 = y3;

x33 = x3 + 2*offset;
y33 = y3;

x35 = x3 + offset;
y35 = y3 + 50;

x34 = x32 + offset;
y34 = y35;

x36 = x32 - offset;
y36 = y35;

%rectângulo representativo da estrutura

a = x02 + 35;
b = a - 70;
c = y3 + 50;
d = y3;

%centro de massa = cento geométrico

xc = x02;
yc = c/2;

%representação gráfica - com asteriscos (vermelho) representam-se as pernas que estão em
contacto com o solo e com círculos (azul) as que estão a rodar livremente

M = moviein(Dim);
for k=1:Dim
    plot(x3,y3,'ro',x32(k),y32,'b*',x33,y33,'ro')
    hold on
    plot(x34(k),y34,'b*',x35,y35,'ro',x36(k),y36,'b*')

    plot([x3,x33],[y3,y33],'g-',[x3,x35],[y3,y35],'g-
    ',[x33,x35],[y33,y35],'g-')

    plot([a(k),a(k)],[d,c],'k',[b(k),b(k)],[d,c],'k')
    plot([b(k),a(k)],[c,c],'k',[b(k),a(k)],[d,d],'k')

    plot(xc(k),yc,'m*')

```



```
axis equal
axis([-70 140 -5 70])
grid on
text(83,35,'*-perna em movimento')
text(83,25,'o-perna fixa')
title('Simulação cinemática da vista de topo do robot')

M(:,k) = getframe
hold off
end
hold on

%-----
```



```

%-----
% nome do ficheiro : grafico2
%-----
%representação gráfica da extremidade de todas as 6 pernas do robot, estrutura e centro de
massa no 2º passo
%-----

function[c]=grafico2(x32,y32,y3,x3,x02)
global offset

Dim = length(x32);
%calculo das coordenadas dos pontos das pernas3,4,5 e 6
y3 = 5;

x32 = x32 + 12;
y32 = y3;

x33 = x32 - 2*offset ;
y33 = y3;

x35 = x32 - offset;
y35 = y3 + 50;

x34 = x3 + offset;
y34 = y35;

x36 = x3 - offset;
y36 = y35;

%rectângulo representativo da estrutura

a = x02 - offset + 12+ 35;
b = a - 70;
c = y3 + 50;
d = y3;

%centro de massa = cento geométrico

xc = x02 - offset+12;
yc = c/2;

%representação gráfica - com asteriscos (vermelho) representam-se as pernas que estão em
contacto com o solo e com círculos (azul) as que estão a rodar livremente

M = moviein(Dim);
for k=1:Dim
    plot(x3,y3,'ro',x32(k),y32,'b*',x33(k),y33,'b*')
    hold on
    plot(x34,y34,'ro',x35(k),y35,'b*',x36,y36,'ro')

    plot ([x3,x34],[y3,y34],'g-', [x3,x36],[y3,y36],'g-
    ', [x34,x36],[y34,y36],'g-')

    plot ([a(k),a(k)],[d,c],'k',[b(k),b(k)],[d,c],'k')
    plot ([b(k),a(k)],[c,c],'k',[b(k),a(k)],[d,d],'k')

```



```
plot(xc(k),yc,'m*')

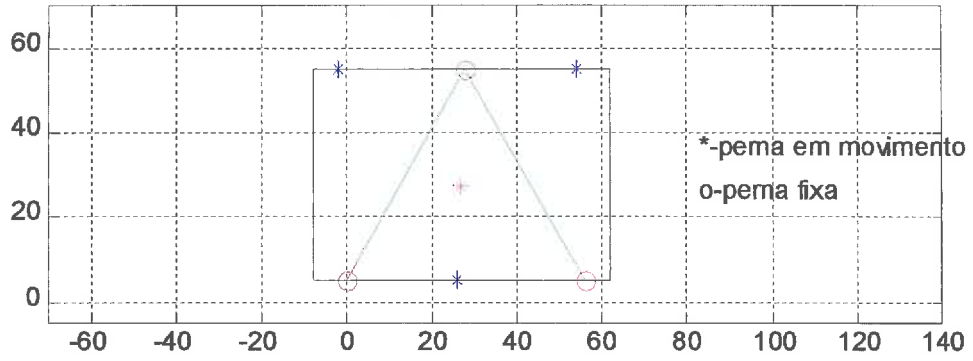
axis equal
axis([-70 140 -5 70])
grid on
text(83,35,'*-perna em movimento')
text(83,25,'o-perna fixa')
title('Simulação cinemática da vista de topo do robot')

M(:,k) = getframe
hold off
end
hold on

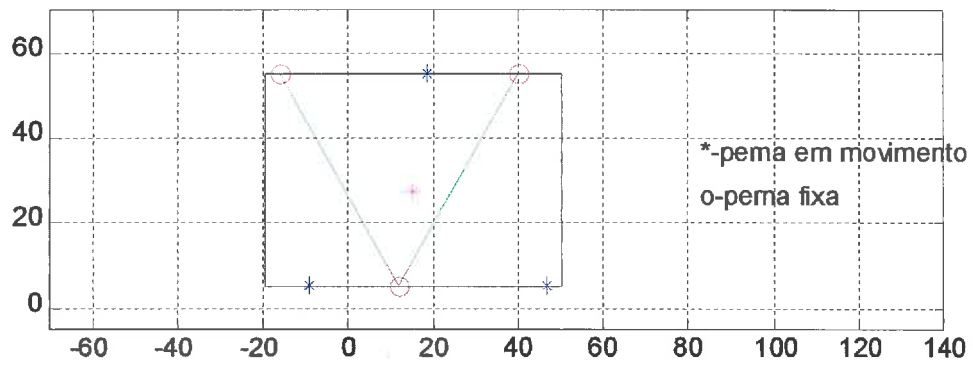
%-----
```



Simulação cinemática da vista de topo do robot



Simulação cinemática da vista de topo do robot



Simulação lateral

```

%-----
% nome do ficheiro : Pernas
%-----
% simulação de 3 pernas de um robot vistas lateralmente sem planeamento de
trajectória
%-----

clc
clear all

global L1 L2 r raio offset

step = pi/30;
teta = 0:step:pi;
offset = 28;

r = 4;      %raio do elo
L1 = 20;    %comprimento da 1ª parte da pata
L2 = 20;    %comprimento da 2ª parte da pata
g = 9.81;   %aceleração da gravidade
m = 25;     %peso total do robot
%-----
%1º PASSO
%-----
%- calculo de todos os pontos críticos (P0,P1,P2,P3) para a perna 1 e 2
durante 1 rotação(pernafixa2d e D_pernafixa2d )
%- calculo das coordenadas dos pontos para a perna 3 e representação
gráfica do conjunto das 3 pernas e estrutura do robot no 1º passo
(grafico_lpasso_inicio e grafico_lpasso_final)
%-----

x3 = 0;
y3 = 0;

[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa2d(teta,x3,y3);

[graficoF2P] =
grafico_lpasso_inicio(teta,x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,
x32,y32);

x3a = x32(31);
y3a = y32(31);

x0a = x02(31);
y0a = y02(31);

[x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2,x3,y3]=D_pernafixa2d(tet
a,x3a,y3a,x0a,y0a);

```




```
[graficoF2P] =
grafico_lpasso_final(teta,x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2
,x3,y3);

%-----
%2° PASSO
%-----
%- calculo de todos os pontos críticos (P0,P1,P2,P3) para a perna 1 e 2
durante 1 rotação ( pernafixa_2p2 e D_pernafixa2d )
%- calculo das coordenadas dos pontos para a perna 3 e representação
gráfica do conjunto das 3 pernas e estrutura do robot no 2° passo
(grafico_lpasso_inicio e grafico_lpasso_final)
%-----

x3 = x3(31);
y3 = y3(31);

[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa_2p2(tet
a,x3,y3);

[graficoF2P] =
grafico_lpasso_inicio(teta,x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,
x32,y32);

x3a = x32(31);
y3a = y32(31);

x0a = x02(31);
y0a = y02(31);

[x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2,x3,y3]=D_pernafixa2d(tet
a,x3a,y3a,x0a,y0a);

[graficoF2P] =
grafico_lpasso_final(teta,x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2
,x3,y3);

%-----
```



```

%-----
% nome do ficheiro : Pernas_PT
%-----
% simulação de 3 pernas de um robot vistas lateralmente com planeamento de trajectória
%-----

clc
clear all

global L1 L2 r raio offset

offset = 28;
N=50;

%PLANEAMENTO DE TRAJECTORIA

tt = 0:1/N:0.5;
teta = 12*pi.*tt.*tt - 16*pi.*tt.*tt.*tt;

r = 4;          %raio do elo
L1 = 20;        %comprimento da 1ª parte da pata
L2 = 20;        %comprimento da 2ª parte da pata
g = 9.81;      %aceleração da gravidade
m = 25;        %peso total do robot

%-----
%1º PASSO
%-----
%- calculo de todos os pontos críticos (P0,P1,P2,P3) para a perna 1 e 2
durante 1 rotação (pernafixa2d e D_pernafixa2d )
%- calculo das coordenadas dos pontos para a perna 3 e representação gráfica do conjunto das
3 pernas e estrutura do robot no 1º passo (grafico_1passo_inicio e grafico_1passo_final)
%-----

x3 = 0;
y3 = 0;

[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa2d(teta,
x3,y3);

[graficoF2P] =
grafico_1passo_inicio(teta,x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,
x32,y32);

x3a = x32 (N/2+1);
y3a = y32 (N/2+1);

x0a = x02 (N/2+1);
y0a = y02 (N/2+1);

[x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2,x3,y3]=D_pernafixa2d(tet
a,x3a,y3a,x0a,y0a);

```



```

[graficoF2P] =
grafico_lpasso_final(teta,x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2
,x3,y3);

%-----
%2° PASSO
%-----
%- calculo de todos os pontos críticos (P0,P1,P2,P3) para a perna 1 e 2
durante 1 rotação ( pernafixa_2p2 e D_pernafixa2d )
%- calculo das coordenadas dos pontos para a perna 3 e representação
gráfica do conjunto das 3 pernas e estrutura do robot no 2° passo (
grafico_lpasso_inicio e grafico_lpasso_final)
%-----

x3 = x3(N/2+1);
y3 = y3(N/2+1);

[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa_2p2(tet
a,x3,y3);

[graficoF2P] =
grafico_lpasso_inicio(teta,x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,
x32,y32);

x3a = x32(N/2+1);
y3a = y32(N/2+1);

x0a = x02(N/2+1);
y0a = y02(N/2+1);

[x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2,x3,y3]=D_pernafixa2d(tet
a,x3a,y3a,x0a,y0a);

[graficoF2P] =
grafico_lpasso_final(teta,x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2
,x3,y3);

%-----

```

```
%-----  
% nome do ficheiro : pernafixa2d  
%-----  
%função para calcular as coordenadas de todos os pontos críticos de uma  
perna durante uma rotação completa.  
%-----  
  
function  
[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa2d(teta,  
x3,y3);  
  
global L1 L2 r raio offset  
  
[x0,y0,x1,y1,x2,y2,x3,y3] = fixa(teta,x3,y3);  
  
[x02,y02,x12,y12,x22,y22,x32,y32] = D_suspensa(teta,x0,y0);  
%-----
```



```
%-----  
% nome do ficheiro : fixa  
%-----  
%função para calcular as coordenadas dos pontos P1, P2, P3 de uma perna quando o ponto P3  
está em contacto com solo. O ponto P3 é o ponto de referencia.  
%-----  
  
function[x0,y0,x1,y1,x2,y2,x3,y3] = fixa(teta,x3,y3);  
global L1 L2 r  
  
%Beta é o ângulo que a haste faz com a vertical no decorrer do movimento. Este ângulo  
expressa-se em função de Teta.  
  
beta = -pi/2 + acos(-r*cos(teta)/L1);  
  
% P0 -----  
  
x0 = (L1+L2)*sin(beta);  
y0 = -r*sin(-teta) + L2*cos(beta);  
  
% P1 -----  
  
x1 = -r*cos(teta) + (L1+L2)*sin(beta);  
y1 = L2*cos(beta);  
  
% P2 -----  
  
y2 = (L1+L2)*cos(beta);  
x2 = (L1+L2)*sin(beta);  
  
% P3 -----  
x3= x3;  
y3= y3;  
%-----
```



```
%-----  
% nome do ficheiro : D_suspensa  
%-----  
%função para calcular as coordenadas dos pontos P1, P2, P3 de uma perna  
quando esta roda livremente. O ponto P0 é o ponto de referencia.  
%-----  
  
function[x02,y02,x12,y12,x22,y22,x32,y32] = D_suspensa(teta,x0,y0);  
global L1 L2 r offset  
  
%Beta é o ângulo que a haste faz com a vertical no decorrer do movimento. Este ângulo  
expressa-se em função de Teta.  
  
beta = -pi/2 + acos(-r*cos(teta)/L1);  
%P0 -----  
  
x02 = x0 + offset;  
y02 = y0;  
  
% P1 -----  
x12 = r*cos(teta) + x02;  
y12 = r*sin(teta) + y02;  
  
% P2 -----  
y22 = y12 + L1*cos(beta);  
x22 = x02;  
  
% P3 -----  
x32= x22 + (L1+L2)*sin(beta);  
y32= y22 - (L1+L2)*cos(beta);  
%-----
```



```

%-----
% nome do ficheiro : grafico_lpasso_inicio
%-----

function [graficoF2P] =
grafico_lpasso_inicio(teta,x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,
x32,y32)
global L1 L2 r raio offset

Dim = length(x1);

%M = moviein(Dim);
%coordenadas dos pontos da 3ª perna

x03 = x0 + 2*offset;
y03 = y0 ;

x13 = x1 + 2*offset;
y13 = y1 ;

x23 = x2 + 2*offset;
y23 = y2 ;

x33 = x3 + 2*offset;
y33 = y3 ;

% estrutura
a = x03 + 10;
b = a - 76;
c = y0 + 15;
d = y0 - 10;

for K=1:Dim,
    %representação gráfica da 1ª perna
    plot (x0(K),y0(K), 'b+')
    hold on
    plot (x1(K),y1(K), 'm')
    plot (x2(K),y2(K), 'g')
    plot ([x0(K),x1(K)], [y0(K),y1(K)], 'r')
    plot ([x1(K),x2(K)], [y1(K),y2(K)], 'b')
    plot([x1(K),x3], [y1(K),y3], 'b')

    %representação gráfica da 2ª perna

    plot (x02(K),y02(K), 'b+')
    plot (x12(K),y12(K), 'm')
    plot (x22(K),y22(K), 'g')
    plot ([x02(K),x12(K)], [y02(K),y12(K)], 'r')
    plot ([x12(K),x22(K)], [y12(K),y22(K)], 'b')
    plot([x12(K),x32(K)], [y12(K),y32(K)], 'b')

    %representação gráfica da 3ª perna

    plot (x03(K),y03(K), 'b+')
    plot (x13(K),y13(K), 'm')
    plot (x23(K),y23(K), 'g')
    plot ([x03(K),x13(K)], [y03(K),y13(K)], 'r')
    plot ([x13(K),x23(K)], [y13(K),y23(K)], 'b')
    plot([x13(K),x33], [y13(K),y33], 'b')

```



```
plot ([-150,150],[0,0],'g',[-150,150],[-0.3,-0.3],'g')
plot ([a(K),a(K)],[d(K),c(K)'],'k',[b(K),b(K)],[d(K),c(K)'],'k')
plot ([b(K),a(K)],[c(K),c(K)'],'k',[b(K),a(K)],[d(K),d(K)'],'k')

axis equal
axis([-150 150 -20 100]), grid on,
pause(0.01)
%title('Simulação cinemática da vista lateral');
hold off

end
hold on
%-----
```




```
%-----  
% nome do ficheiro : D_pernafixa2d  
%-----  
%função para calcular as coordenadas de todos os pontos críticos de uma perna durante uma  
rotação completa, para o 2º passo da simulação.  
%-----  
function  
[x02,y02,x12,y12,x22,y22,x32,y32,x0,y0,x1,y1,x2,y2,x3,y3]=D_pernafixa2d(teta,  
a,x3a,y3a,x0a,y0a);  
  
global L1 L2 r raio offset  
  
[x02,y02,x12,y12,x22,y22,x32,y32] = D_fixa(teta,x3a,y3a,x0a,y0a);  
  
[x0,y0,x1,y1,x2,y2,x3,y3] = D_suspensaD(teta,x02,y02);  
  
%-----
```



```
%-----  
% nome do ficheiro : D_fixa  
%-----  
%função para calcular as coordenadas dos pontos P1, P2, P3 de uma perna quando o ponto P3  
está em contacto com solo. O ponto P3 é o ponto de referencia.  
%-----  
  
function[x02,y02,x12,y12,x22,y22,x32,y32] = D_fixa(teta,x3a,y3a,x0a,y0a);  
global L1 L2 r  
  
beta = -pi/2 + acos(-r*cos(teta)/L1);  
  
% P1 -----  
  
x12 = -r*cos(teta) + (L1+L2)*sin(beta) + x3a ;  
y12 = L2*cos(beta) + y3a;  
  
% P2 -----  
  
y22 = (L1+L2)*cos(beta) + y3a;  
x22 = (L1+L2)*sin(beta) + x3a ;  
  
% P3 -----  
  
x32= x3a;  
y32= y3a;  
  
% P0 -----  
  
x02 = (L1+L2)*sin(beta) + x3a ;  
y02 = -r*sin(-teta) + L2*cos(beta) + y3a ;  
  
%-----
```



```
%-----  
% nome do ficheiro : D_suspensaD  
%-----  
%função para calcular as coordenadas dos pontos P1, P2, P3 de uma perna  
quando esta roda livremente. O ponto P0 é o ponto de referencia.  
%-----  
  
function[x0,y0,x1,y1,x2,y2,x3,y3] = D_suspensaD(teta,x02,y02)  
global L1 L2 r offset  
  
%Beta é o ângulo que a haste faz com a vertical no decorrer do movimento. Este ângulo  
expressa-se em função de Teta.  
  
beta = -pi/2 + acos(-r*cos(teta)/L1);  
  
%P0 -----  
  
x0 = x02 - offset;  
y0 = y02;  
  
% P1 -----  
  
x1 = r*cos(teta) + x0;  
y1 = r*sin(teta) + y0;  
  
% P2 -----  
  
y2 = y1 + L1*cos(beta);  
x2 = x0;  
  
% P3 -----  
  
x3= x2 + (L1+L2)*sin(beta);  
y3= y2 - (L1+L2)*cos(beta);  
%-----
```



```

%-----
% nome do ficheiro : grafico_lpasso_final
%-----

function [graficoF2P] =
grafico_lpasso_final(teta,x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x
32,y32)
global L1 L2 r raio offset

Dim = length(x1);

%M = moviein(Dim);
%coordenadas dos pontos da 3ª perna
x03 = x02 + 2*offset;
y03 = y02 ;

x13 = x12 + 2*offset;
y13 = y12 ;

x23 = x22 + 2*offset;
y23 = y22 ;

x33 = x32 + 2*offset;
y33 = y32 ;

% estrutura
a = x03 + 10;
b = a - 76;
c = y0 + 15;
d = y0 - 10;

for K=1:Dim,
    %representação gráfica da 1ª perna

    plot (x0(K),y0(K), 'b+')
    hold on
    plot (x1(K),y1(K), 'm')
    plot (x2(K),y2(K), 'g')
    plot ([x0(K),x1(K)], [y0(K),y1(K)], 'r')
    plot ([x1(K),x2(K)], [y1(K),y2(K)], 'b')
    plot([x1(K),x3], [y1(K),y3], 'b')

    %representação gráfica da 2ª perna

    plot (x02(K),y02(K), 'b+')
    plot (x12(K),y12(K), 'm')
    plot (x22(K),y22(K), 'g')
    plot ([x02(K),x12(K)], [y02(K),y12(K)], 'r')
    plot ([x12(K),x22(K)], [y12(K),y22(K)], 'b')
    plot([x12(K),x32(K)], [y12(K),y32(K)], 'b')

    %representação gráfica da 3ª perna

    plot (x03(K),y03(K), 'b+')
    plot (x13(K),y13(K), 'm')
    plot (x23(K),y23(K), 'g')
    plot ([x03(K),x13(K)], [y03(K),y13(K)], 'r')
    plot ([x13(K),x23(K)], [y13(K),y23(K)], 'b')
    plot([x13(K),x33(K)], [y13(K),y33(K)], 'b')

```



```
plot ([-150,150],[0,0], 'g', [-150,150], [-0.3,-0.3], 'g')
plot ([a(K),a(K)], [d(K),c(K)], 'k', [b(K),b(K)], [d(K),c(K)], 'k')
plot ([b(K),a(K)], [c(K),c(K)], 'k', [b(K),a(K)], [d(K),d(K)], 'k')

axis equal
axis([-150 150 -20 100]), grid on,
pause(0.01)
%M(:,K) = getframe;
%title('Simulação cinemática da vista lateral');
hold off
end
hold on

%-----
```

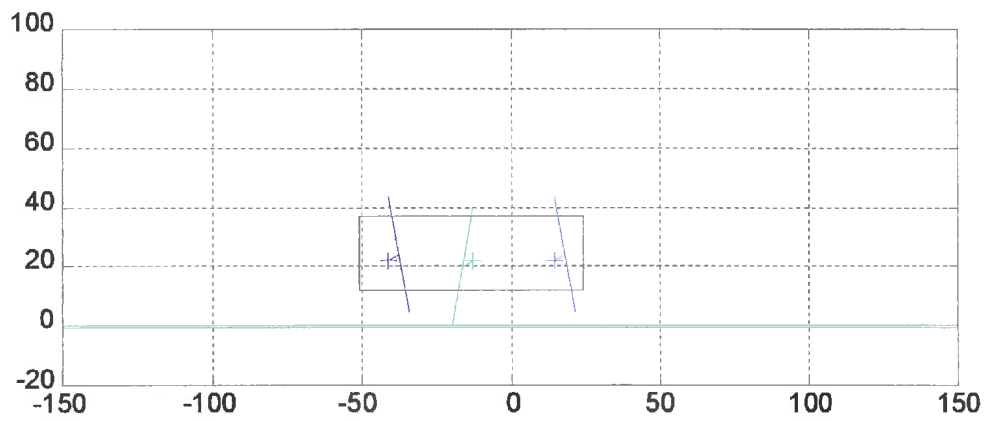
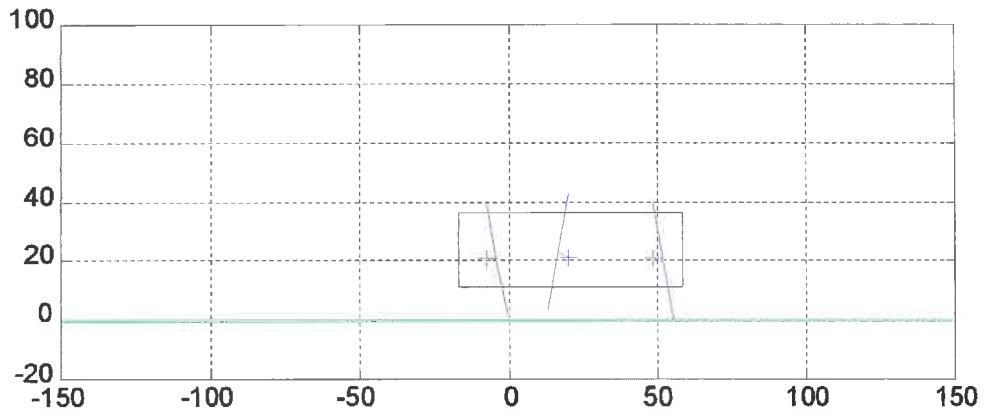


```
%-----  
% nome do ficheiro : pernafixa_2p2  
%-----  
%função para calcular as coordenadas de todos os pontos críticos de uma  
perna durante uma rotação completa, para o 2º passo da simulação.  
%-----  
function  
[x0,y0,x1,y1,x2,y2,x3,y3,x02,y02,x12,y12,x22,y22,x32,y32]=pernafixa_2p2(teta,  
a,x3,y3)  
  
global L1 L2 r raio  
  
[x0,y0,x1,y1,x2,y2,x3,y3] = fixa_2p(teta,x3,y3);  
  
[x02,y02,x12,y12,x22,y22,x32,y32] = D_suspensa(teta,x0,y0);  
  
%-----
```



```
%-----  
% nome do ficheiro : fixa_2p  
%-----  
%função para calcular as coordenadas dos pontos P1, P2, P3 de uma perna  
quando o ponto P3 está em contacto com solo. O ponto P3 é o ponto de  
referencia.  
%-----  
function[x0,y0,x1,y1,x2,y2,x3,y3] = fixa_2p(teta,x3,y3)  
global L1 L2 r raio  
  
beta = -pi/2 + acos(-r*cos(teta)/L1);  
  
% P1 -----  
  
x1 = -r*cos(teta) + (L1+L2)*sin(beta) + x3;  
y1 = L2*cos(beta) - y3;  
  
% P2 -----  
  
y2 = (L1+L2)*cos(beta) - y3;  
x2 = (L1+L2)*sin(beta) + x3;  
  
% P3 -----  
  
x3= x3;  
y3= y3;  
  
% P0 -----  
  
x0 = (L1+L2)*sin(beta) + x3;  
y0 = -r*sin(-teta) + L2*cos(beta) -y3;  
%-----
```





Simulação dinâmica

```

%-----
%NOME DO FICHEIRO: Dinamica
%-----
%- Representação gráfica dos vários gráficos resultantes do estudo dinâmico do robot com e
sem planeamento de trajectória
%-----

clc
clear all
global r L2 m k1 k2 k3 k4 g co so c s

%Definição dos parâmetros conhecidos

r = 0.04;      %raio do elo
L1 = 0.20;    %comprimento da 1ª parte da pata
L2 = 0.20;    %comprimento da 2ª parte da pata
g = 9.81;    %aceleração da gravidade
m = 25;      %peso total do robot

%-----
%Planeamento de trajectoria - tetal
%-----
%Representação grafica da variação de tetal ao longo do movimento-
figure(1) %-----

t = 0:1/100:0.5;
tetotal = pi + 12*pi.*t.*t - 16*pi.*t.*t.*t;
ang = [tetotal];
co = cos(tetotal);
so = sin(tetotal);

plot (t,ang,'r')
title('Variação do ângulo'),xlabel('tempo (s)'),ylabel('Ang (rad)')
axis([0 0.6 3 6.5])
grid on

%Representação gráfica da velocidade angular ao longo do movimento-
figure(2)%-----

Dttotal = 24*pi.*t - 48*pi.*t.*t;
Dang = [Dttotal];

plot (t,Dang,'m')
title('Velocidade Angular'),xlabel('tempo (s)'),ylabel('W (rad/s)')
axis([0 0.6 0 10])
grid on

%Representação gráfica da aceleração angular ao longo do movimento-
figure(3)%-----

DDttotal = 24*pi - 96*pi.*t;
plot (t,DDttotal,'g')

```



```
title('Aceleração Angular'),xlabel('tempo (s)'),ylabel('dW (rad/s^2)')
axis([0 0.6 -80 80])
grid on
```

```
%Representação gráfica do binário exigido durante o movimento-
figure(4)%-----
```

```
k1 = (L1+L2)*r/L1;
k2 = r^2/L1^2;
k3 = L2*k2;
k4 = 2*r*k3;
```

```
[DDEc]=DDEcinetica(tetal,Dtetal,DDtetal);
[DEc]=DEcinetica(tetal,Dtetal);
[DEp]=DEpotencial(tetal);
```

```
T = DDEc - (DEc + DEp);           %calculo do binário
plot (t,T,'k')
title('Binário do Motor'),xlabel('tempo (t)'),ylabel('T(N.m)')
axis([0 0.6 -15 15])
grid on
```

```
%Representação gráfica da potência exigida durante o movimento-
figure(5)%-----
```

```
P = T .* Dtetal;                 %calculo da potência
plot (t,P,'k')
title('Potência do Motor'),xlabel('tempo (t)'),ylabel('P (w)')
axis([0 0.6 -110 110])
grid on
```

```
%-----
%Sem Planeamento de trajectória - teta
%-----
```

```
step = pi/50;
teta = pi:step:2*pi;
Dteta(1:51) = 2*pi;             %Derivada de Teta - velocidade angular
```

```
c = cos(teta);
s = sin(teta);
```

```
%Representação gráfica da variação de teta ao longo do movimento-
figure(6) %-----
```

```
plot(t,teta,'b')
axis([0 0.6 3 6.5])
title('Variação do ângulo'),xlabel('tempo (t)'),ylabel('Ang(rad)')
grid on
```

```
%Representação gráfica da velocidade angular ao longo do movimento-
figure(7) %-----
```

```
plot(t,Dteta,'b')
axis([0 0.6 0 7])
title('Velocidade Angular'),xlabel('tempo (t)'),ylabel('W(rad/s)')
grid on
```



```
%Representação gráfica do binário exigido durante o movimento-
figure(8) %-----

DDteta=0;
[DDEc1]=DDEcineticaSp(teta,Dteta,DDteta);
[DEc1]=DEcineticaSp(teta,Dteta);
[DEp1]=DEpotencialSp(teta);
T1 = DDEc1 - (DEc1 + DEp1);          %calculo do binário

plot (t,T1,'k')
title('Binário do Motor Sem Planeamento'),xlabel('tempo
(t)'),ylabel('T(N.m)')
axis([0 0.6 -11 11])
grid on

%Representação gráfica da potência exigida durante o movimento-
figure(9)%-----

P1 = T1 .* Dteta;                    %calculo da potência
plot (t,P1,'k')
title('Potência do Motor Sem Planeamento'),xlabel('tempo (t)'),ylabel('P
(w)')
axis([0 0.6 -65 65])
grid on
%-----
```



```

%-----
%NOME DO FICHEIRO: DEcinetica
%-----
%Calculo da derivada parcial da energia cinética em ordem a teta1 - com planeamento
%-----
%m - massa do robot
%K1,K2,K3,K4 - constantes de calculo definidas no programa principal
%r,L1,L2 - parâmetros da perna
%so - seno de teta1
%co - coseno de teta1
%Dteta1 - derivada de teta1 em relação ao tempo
%DDteta1 - derivada de 2ª ordem de teta1 em relação ao tempo
%-----

function [DEc]=DEcinetica(teta1,Dteta1)

global r L2 m k1 k2 k3 k4 g co so
DEc = 0.5 * m * Dteta1.*Dteta1.*(((2*k1*k1-2*r*r)*co.*so + (k3*k3*(-
2*co.*so.*so.*so+2*so.*co.*co.*co)...
.*(1-k2*co.*co) - 2*k3*k3*co.*co.*co.*so.*so.*so*k2)./(1-k2*co.*co).*(1-
k2*co.*co)) ...
+ (k4*(co.*co.*co-2*co.*so.*so).*(1-k2*co.*co)-
k4*k2*so.*so.*co.*co.*co)./((1-k2*co.*co)...
.*sqrt(1-k2*co.*co)));

%-----

```



```

%-----
%NOME DO FICHEIRO: DDEcinetica
%-----
%Calculo da derivada em ordem ao tempo, da derivada parcial da energia cinética em ordem
a tetal - com planeamento
%-----
%m - massa do robot
%K1,K2,K3,K4 - constantes de calculo definidas no programa principal
%r,L1,L2 - parametros da perna
%so - seno de tetal
%co - coseno de tetal
%Dtetal - derivada de tetal em relação ao tempo
%DDtetal - derivada de 2ª ordem de tetal em relação ao tempo
%-----

function[DDEc]=DDEcinetica(tetal,Dtetal,DDtetal)

global r L2 m k1 k2 k3 k4 g co so
DDEc = m*DDtetal.*(k1*k1*so.*so+r*r*co.*co+(k3*k3*co.*co.*so.*so)/(1-
k2*co.*co)...
    +(k4*so.*co.*co)./sqrt(1-k2*co.*co)) +
m*Dtetal.*(2*k1*k1*so.*co.*Dtetal...
    -2*r*r*co.*so.*Dtetal + k3*k3*((-
2*co.*so.*so.*so+2*so.*co.*co.*co).*(1-k2*co.*co).*Dtetal...
    -2*k2*co.*co.*co.*so.*so.*so.*Dtetal)/(1-k2*co.*co).*(1-k2*co.*co)) +
k4*((co.*co.*co.*Dtetal-2*co.*so.*so.*Dtetal)...
    *(1-k2*co.*co)-k2*so.*so.*co.*co.*co.*Dtetal)/((1-k2*co.*co).*sqrt(1-
k2*co.*co)));

%-----

```



```
%-----  
%NOME DO FICHEIRO: DEpotencial  
%-----  
% calculo da derivada parcial da energia Potencial em ordem a tetal  
%-----  
%m - massa do robot  
%K1,K2,K3,K4 - constantes de calculo definidas no programa principal  
%r,L1,L2 - parâmetros da perna  
%so - seno de tetal  
%co - coseno de tetal  
%-----  
  
function[DEp]=DEpotencial(tetal)  
  
global r L2 m k1 k2 k3 k4 g co so  
DEp = m*g*(r*co + L2*k2*co.*so./(sqrt(1-k2*co.*co)));  
%-----
```



```

%-----
%NOME DO FICHEIRO: DEcineticaSp
%-----
%Calculo da derivada parcial da energia cinetica em ordem a teta1 - sem planeamento
%-----
%m - massa do robot
%K1,K2,K3,K4 - constantes de calculo definidas no programa principal
%r,L1,L2 - parametros da perna
%s - seno de teta
%c - coseno de teta
%Dteta1 - derivada de teta em relação ao tempo
%DDteta1 - derivada de 2ª ordem de teta em relação ao tempo
%-----

function[DEc1]=DEcineticaSp(teta,Dteta)

global r L2 m k1 k2 k3 k4 g c s
DEc1 = 0.5 * m * Dteta.*Dteta.*(((2*k1*k1-2*r*r)*c.*s + (k3*k3*(-
2*c.*s.*s.*s+2*s.*c.*c.*c)...
.*(1-k2*c.*c) - 2*k3*k3*c.*c.*s.*s.*s*k2)./(1-k2*c.*c).*(1-k2*c.*c))
...
+ (k4*(c.*c.*c-2*c.*s.*s).*(1-k2*c.*c)-k4*k2*s.*s.*c.*c.*c)./((1-
k2*c.*c)...
.*sqrt(1-k2*c.*c)));

%-----

```



```

%-----
%NOME DO FICHEIRO: DDEcineticaSp
%-----
%Calculo da derivada em ordem ao tempo, da derivada parcial da energia cinética em ordem
a teta - sem planeamento
%-----
%m - massa do robot
%K1,K2,K3,K4 - constantes de calculo definidas no programa principal
%r,L1,L2 - parâmetros da perna
%s - seno de teta
%c - coseno de teta
%Dteta - derivada de teta em relação ao tempo
%DDteta - derivada de 2ª ordem de teta em relação ao tempo
%-----

function[DDEc1]=DDEcineticaSp(teta,Dteta,DDteta)

global r L2 m k1 k2 k3 k4 g c s
DDEc1 = m*Dteta.*(2*k1*k1*s.*c.*Dteta...
    -2*r*r*c.*s.*Dteta + k3*k3*((-2*c.*s.*s.*s+2*s.*c.*c.*c).*(1-
k2*c.*c).*Dteta...
    -2*k2*c.*c.*c.*s.*s.*Dteta)./(1-k2*c.*c).*(1-k2*c.*c)) +
k4*((c.*c.*c.*Dteta-2*c.*s.*s.*Dteta)...
    .*(1-k2*c.*c)-k2*s.*s.*c.*c.*c.*Dteta)./((1-k2*c.*c).*sqrt(1-
k2*c.*c)));

%-----

```




```
%-----  
%NOME DO FICHEIRO: DEpotencialSp  
%-----  
% calculo da derivada parcial da energia Potencial em ordem a teta  
%-----  
%m - massa do robot  
%K1,K2,K3,K4 - constantes de calculo definidas no programa principal  
%r,L1,L2 - parametros da perna  
%s - seno de teta  
%c - coseno de teta  
%-----  
  
function[DEp1]=DEpotencialSp(teta)  
  
global r L2 m k1 k2 k3 k4 g c s  
DEp1 = m*g*(r*c + L2*k2*c.*s./sqrt(1-k2*c.*c));  
  
%-----
```



Anexo 5

Documentação



'S' SERIES Frame 850

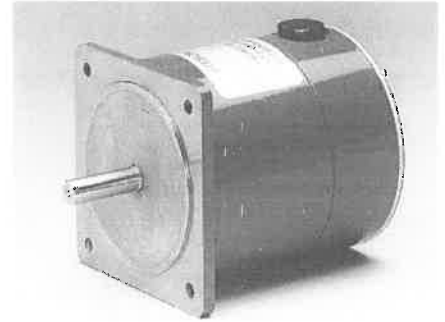
Frame Reference SDL850 and SDT850

Motors manufactured to STEBON 'S' Series frame 850 are available in three basic enclosures:

- o SDL motors are the common flying lead design with 8 leads out and manufactured to IP22 rating.
- o SDT motors have a terminal box incorporated into the NDE bracket. The standard enclosure is to IP44. An alternative IP66 enclosure is also available.

Specification

- o Hybrid design.
- o 1.8° Step angle (200 steps per rev).
- o Unique two-phase winding for use with bipolar drives only.
- o Class F insulation to IEC 60085 and BS2757.
- o NEMA Size 34.
- o Bearing loading, axial, 160N max.
- o Bearing loading, radial, 250N max.

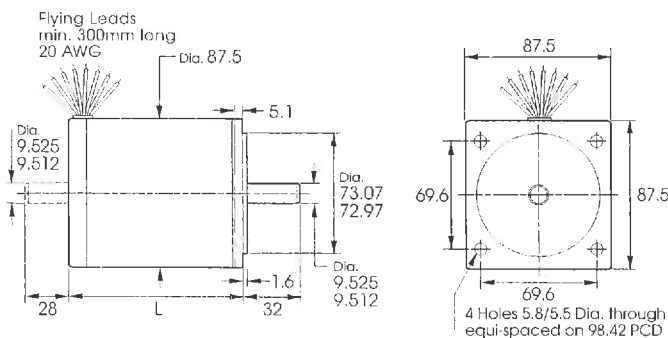


Ratings

Frame reference	Maximum Holding Torque Nm	Maximum Detent Torque Nm	Bipolar Amps per phase Amps	Resistance per phase Ohms	Inductance per phase mH	Rotor Inertia kgcm ²	Weight kg
SDL/SDT851-850-24	1.5	0.05	2.4	0.89	5.2	0.60	2.2
SDL/SDT851-480-32	1.5	0.05	3.2	0.48	3.2	0.60	2.2
SDL/SDT851-250-45	1.5	0.05	4.5	0.24	1.5	0.60	2.2
SDL/SDT851-100-70	1.5	0.05	7.0	0.10	0.6	0.60	2.2
SDL/SDT852-740-39	3.5	0.08	3.9	0.74	6.5	1.25	3.2
SDL/SDT852-380-56	3.5	0.08	5.6	0.37	3.2	1.25	3.2
SDL/SDT852-250-70	3.5	0.08	7.0	0.25	2.0	1.25	3.2
SDL/SDT852-200-79	3.5	0.08	7.9	0.19	1.6	1.25	3.2
SDL/SDT852-150-90	3.5	0.08	9.0	0.15	1.2	1.25	3.2
SDL/SDT853-1000-37	5.0	0.10	3.7	0.99	11.6	2.00	4.0
SDL/SDT853-500-54	5.0	0.10	5.4	0.50	5.6	2.00	4.0
SDL/SDT853-250-75	5.0	0.10	7.5	0.25	2.9	2.00	4.0
SDL/SDT853-200-85	5.0	0.10	8.5	0.20	2.3	2.00	4.0
SDL/SDT853-150-100	5.0	0.10	10.0	0.16	1.6	2.00	4.0
SDL/SDT853-100-120	5.0	0.10	12.0	0.11	1.2	2.00	4.0

Dimensions (mm)

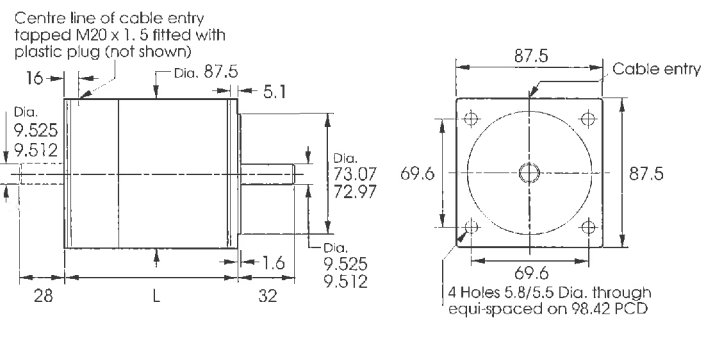
SDL



Frame size	L (IP22)
SDL851	73
SDL852	105
SDL853	138

Note: SDL motors are available with the non-drive end shaft extension shown -----

SDT



Frame size	L (IP44)	L (IP66)
SDT851	98	117
SDT852	131	149
SDT853	163	182

Note: SDT motors are available with the non-drive end shaft extension shown -----

Factory Default Settings
 Table 4-2 below provides the SD Drive factory default settings. If the factory settings are not appropriate for your application, refer to Chapter 3, Installation, for instructions on adjusting the appropriate drive and motherboard DIP switches and links.

Feature	Default Setting	Function Selected
SDC Motherboard Jumpers		
LK1	Position A	Enables use of onboard rate pot
LK2	Position A	Enables use of onboard slow rate pot
LK3	Position A	Enables use of onboard fast rate pot
SD Motherboard Jumpers		
LK1	Position B	Enables use of onboard last rate pot
LK2	Position B	Enables use of onboard slow rate pot
Drive Jumpers		
LK1	Installed	DO NOT CHANGE
LK2	Not installed	DO NOT CHANGE
LK3	Not installed	Selects resolution of 400 steps/rev
LK4	Installed	Motor is energized at all times
Drive DIP Switch	SW1 SW2 SW3 SW4	Maximum current
	OFF OFF OFF OFF	

Table 4-2. SD Drive Factory Default Settings

SD Drive Direct Connections
 You will need to make connections directly to the drive edge connector if you are not using either of the standard motherboards. The edge connector pin functions are detailed below.

Pin	Row a	Row b	Row c	Pinouts
2	Motor Phase B-	Motor Phase B+	Motor Phase B-	Motor Phase B-
4	Motor Phase B+	Motor Phase B+	Motor Phase B+	Motor Phase B+
6	Motor Phase A-	Motor Phase A-	Motor Phase A-	Motor Phase A-
8	Motor Phase A+	Motor Phase A+	Motor Phase A+	Motor Phase A+
10	+24VDC	+24VDC	+24VDC	+24VDC
12	Logic Supply 1	Logic Supply 1	Logic Supply 1	Motor Supply 1
14	Logic Supply 2	Logic Supply 2	Logic Supply 2	Motor Supply 2
16	0V	0V	0V	0V
18	0V	0V	0V	0V
20	Fast	Fast	Fast	Fault
22	Slow	Slow	Slow	Zero Phase
24	Rate Adjust Com.	Rate Adjust Com.	Rate Adjust Com.	Slow Rate Adjust
26	Fast Rate Adj.	Fast Rate Adj.	Fast Rate Adj.	Direction
28	Internal Clock 0V	Internal Clock 0V	Internal Clock 0V	Clock In
30	Not Connected	Not Connected	Not Connected	Energise
32	External Ref.	External Ref.	External Ref.	Signal 0V

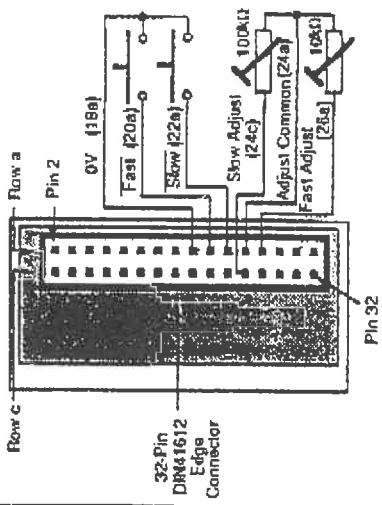


Figure 4-1. Connecting Directly to the Drive

The Drive Signal Descriptions

The signals at the 32 way drive edge connector are as follows:-

SD Drive Specifications

Parameter	Value
Amplifiers	
Type	Bipolar Chopper
Motor resolution	200 or 400 steps/rev (User-selectable)
Protection	Phase to-phase and across phases
Open circuit	If heatsink exceeds 85°C (185°F)
Short circuit	2A/phase (SD2), 3A/phase (SD3), 4.5A/phase (SD5) - DIP switch adjustable
Over-temperature	10kHz @ 200 steps/rev
Nominal output current (two phase-on)	20kHz @ 400 steps/rev
Maximum stepping rate	15kHz
Nonreal chopping frequency	
Command Interface	
SD drive module	Built-in pull-up resistors (4k7) to +12V
Input impedance	Low (logic 0) to +2V or short-circuit
Input logic level	High (logic 1) +10V to +12V or open-circuit
Output circuits	Open-collector NPN transistors
Output logic levels	Low (transistor switched to 0V) +1V max. @ 15mA max. High (transistor off) +25V max.
SD drive mounted in an SC rack	Step input is high going pulse, 10µs min. width
Input	Maximum pulse rate is 20kHz
Power	
Drive supply voltage	18-0-18 to 26-0-26VAC, or 24 to 36VDC
Logic supply voltage	18-0-18VAC or +24VDC at 350mA max.
Drive supply current	1.5A (SD2), 2.2A (SD3), Not recommended for use (SD5)
Fuses	1.5A (SD2), 2.2A (SD3), 4.5A (SD5)
Internal Oscillator	1A
Speed range	3.15A (SD2), 4A (SD3), 6.3A (SD5)
Fast	400 - 10,000 steps/sec (ramped)
Slow	40 - 4,000 steps/sec (not ramped)
Presel acceleration time	60 ms
Presel deceleration time	30 ms
Motors	
Type	2-Phase hybrid or permanent magnet (normally 1.8°)
Number of leads	4, 6, or 8 (5 lead not suitable)
Inductance range	1mH-10mH
Typical current range	SD2 1.5-3A SD3 2-4A SD5 4-6A

* For SD2, lower current can be accommodated by leaving all DIP switches OFF and changing the value of resistor R5 on the SDC motherboard or R1 on the SD motherboard

Table 4-1. SD Drive Specifications

The MELSEC FX0N Series

Description

Small-scale PLC offering excellent value for money.

- Small
- Fast
- Universal
- Modular expansion capability

The ability to combine the compact base units with small, modular expansion units and compact I/O expansion units makes the FX0N enormously flexible, giving you a highly economical combination of the cost benefits of compact systems with the versatile expansion capabilities of modular systems.

System structure

- Base unit with full PLC functionality
- Integrated power supply unit
- CPU
- Maintenance-free EEPROM memory
- Integrated digital inputs and outputs
- Expansion units for adapting the controller system to the required I/O ranges and functionality
- Configurable as a slave station in peer-to-peer and 1:n networks (version 2.00 and above)
- User-friendly programming systems, including hand-held programming units, IEC 1131.3-compatible programming software and MMIs
- Accessories

Equipment features

Base units are available in a number of versions with different power supply and output type configurations.

You can choose between units with 230 V AC or 24 V DC power supplies and relay or transistor outputs.

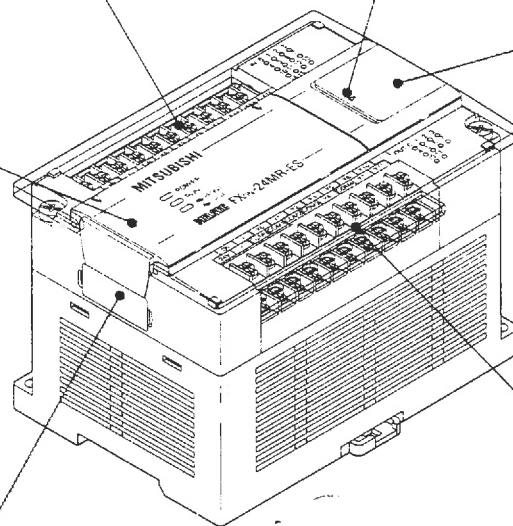
All the base unit versions have the same basic CPU and performance specifications.

High-speed inputs for fast counting tasks with counting frequencies of up to 5 kHz and **interrupt processing capabilities**.

Integrated RUN/STOP switch and two **analog potentiometers** for setpoint value entry.

The base units can be expanded for up to 128 I/Os with both **modular and compact expansion units**.

The base units have integrated, **maintenance-free EEPROM user memory for up to 2000 PLC program steps** with a backup battery to protect memory contents against power failures. You can also configure password protection to prevent unauthorised access to your programs.



Integrated pulse outputs for frequencies from 10 to 2000Hz for controlling stepping motors and outputting **pulse-width modulated signals**.

External EPROM or EEPROM memory cassettes for PLC program storage. These memory cassettes are also available in versions with a real-time clock.

The MELSEC FX2N Series

Description

The MELSEC FX2N series has the most powerful CPUs in the MELSEC FX family. It combines the advantages of a compact PLC with the performance boost of modular PLC systems:

- One of the fastest PLC systems available, with a program cycle period of just 0.08 μ s per logical instruction
- Powerful basic instruction set with an additional 125 dedicated instructions for fast, efficient programming of complex tasks
- Simple handling
- Integrated real-time clock
- Integrated PID controller with auto-tuning facility
- Floating-point math, square root function
- Big memory capacity for up to 16000 PLC program steps

System structure

- Base unit with full PLC functionality
- Integrated power supply unit
- CPU
- Integrated digital inputs and outputs
- Supplementary add-in function boards for adapting the controller system to the required I/O ranges and functionality
- Integration as a master or slave station in peer-to-peer networks and as a slave station in 1:n networks
- Efficient, powerful user-friendly programming with hand-held programming units, MMIs and IEC 1131.3-compatible programming software
- Accessories

Equipment features

A basic MELSEC FX2N PLC system consists of a stand-alone base unit. Just like the modules in the other FX series these base units contain all the PLC components, including the CPU, memory and the I/O control circuitry.

All the base unit versions in the series have the same basic CPU and performance specifications.

A total of 20 different base units are available, with between 16 and 128 I/Os in their standard configuration. Versions are available with 230 V AC power supplies (24 V DC available from Summer 1998) and relay or transistor outputs. The digital inputs are powered by the integrated power supply unit. Removable terminal blocks make reconfiguration for new tasks very quick and easy.

A range of powerful expansion and special function modules enable you to configure your setup flexibly to provide the precise functionality and I/O specifications required by your application.

You can add I/Os to the base units by installing modular expansion units with 8 or 16 additional I/Os each. You can also add a range of compact expansion units and special function modules – for example for processing analog signals, for positioning tasks and to provide additional interfaces.

Integrated high-speed counter inputs for processing fast input signals. For example, you can configure two 60 kHz counters or four 10 kHz counters.

Interrupt processing is also handled via the inputs.

RAM/EEPROM memory for up to **16000 PLC program steps** gives you plenty of reserve, even for big, complex applications.

The base units can be expanded to provide configurations with up to 256 inputs and outputs with **modular and compact expansion units**.

Add-in function boards can be installed in the PLC to provide a **second RS-485 / RS-422 / RS-232 communications interface** for programming or network configurations.

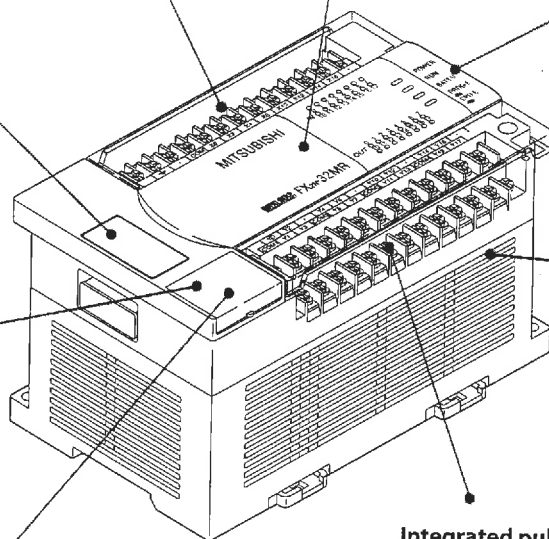
An add-in function board with 8 analog potentiometers is also available.

Integrated serial interface for direct communication with computers

An **integrated RUN/STOP** switch is available

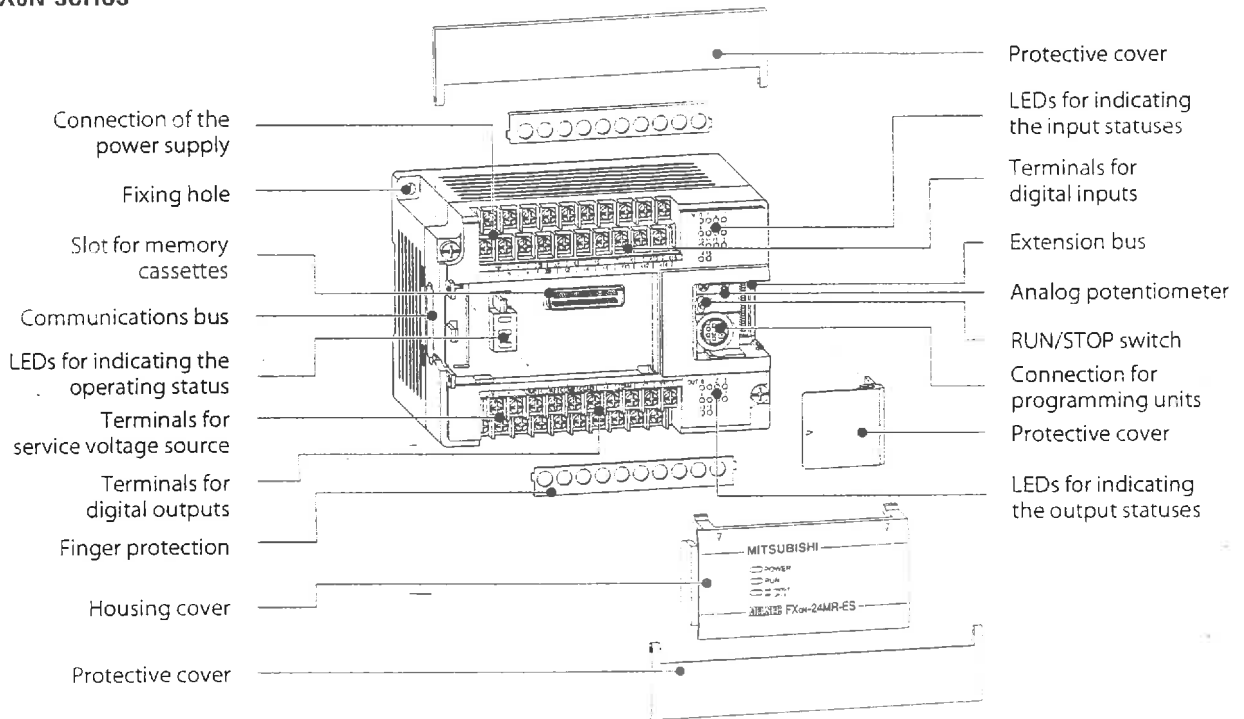
Integrated real-time clock with year, month and time

Integrated pulse outputs for frequencies from 10 to 20000 Hz with **deceleration and acceleration ramps** for controlling stepping motors and outputting **pulse-width modulated signals**.

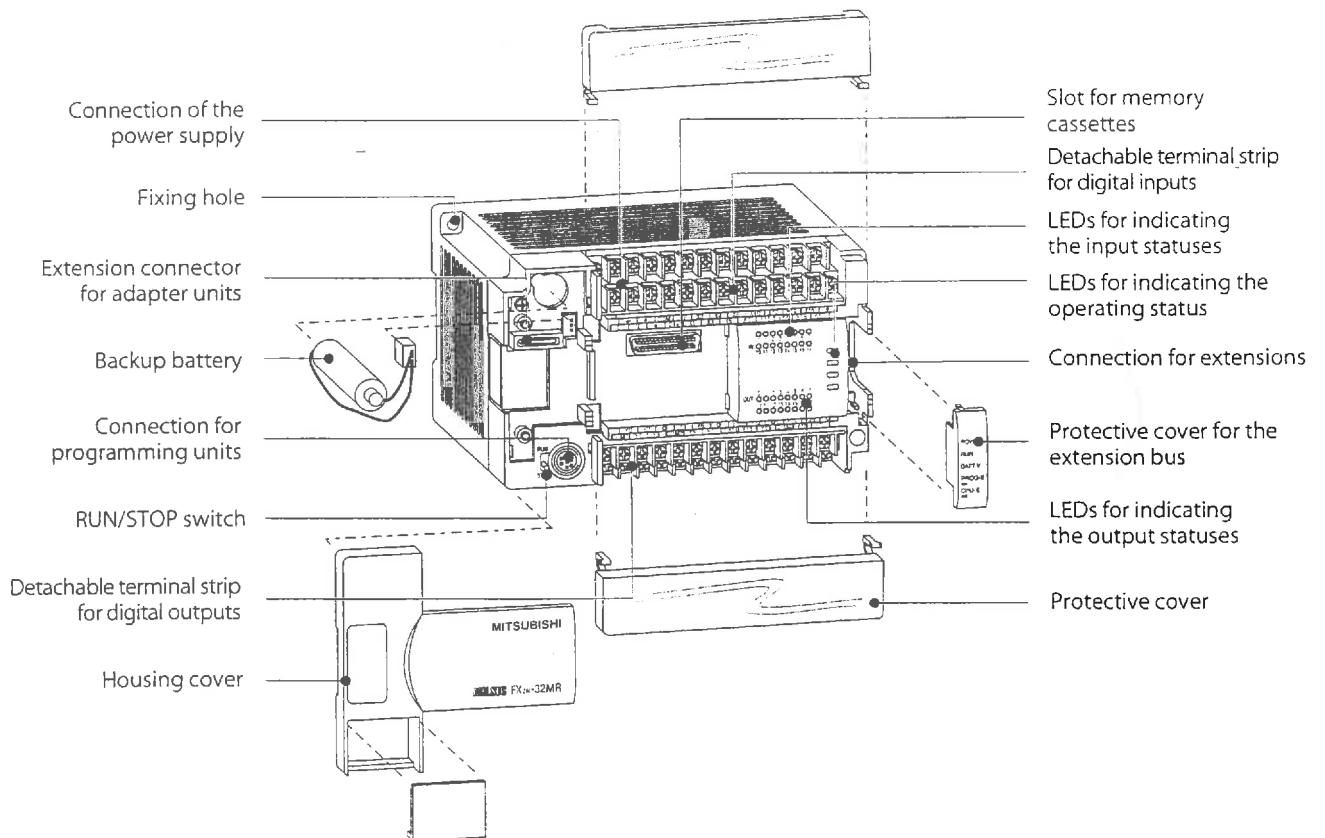


Description of Units

■ FX0N series



■ FX2N series



Combining Units from Different Series

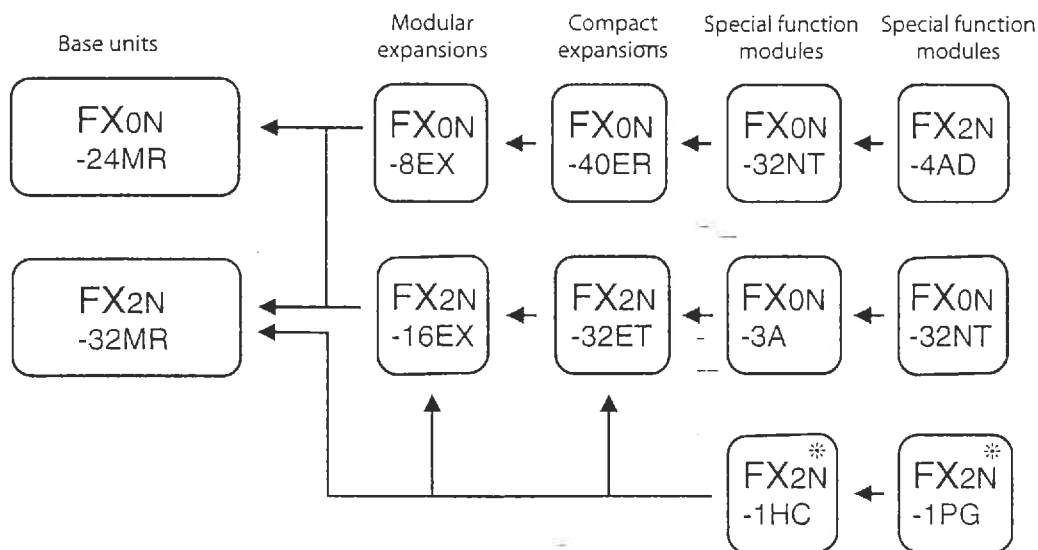
The I/O expansion modules and special function modules from the two series can be combined at will, subject to the restrictions imposed by the differences between the systems.

For example, you can use all the modules for the FX0N series in combination with a base unit from the FX2N series. Combined use of modules from both series is also possible.

A special conversion adapter is available (designation: FX2N-CNV-IF) for connecting modules from the old FX series to the base units of the FX2N series.

The tables and figures below show the restrictions and other special requirements that apply for combined use of modules from different systems.

Series	FX0N	FX2N
Restrictions	The FX2N-1HC and FX2N-1PG-E modules require CPU version 2.00 or higher	—
Special requirements	—	Modules FX0N-485ADP and FX0N-232ADP connect to the left of the CPU and require function board FX2N-CNV-BD for connection to the FX2N.



* Can also be used with MELSEC FX0N and CPU version 2.00 or higher

CPU version 2.0

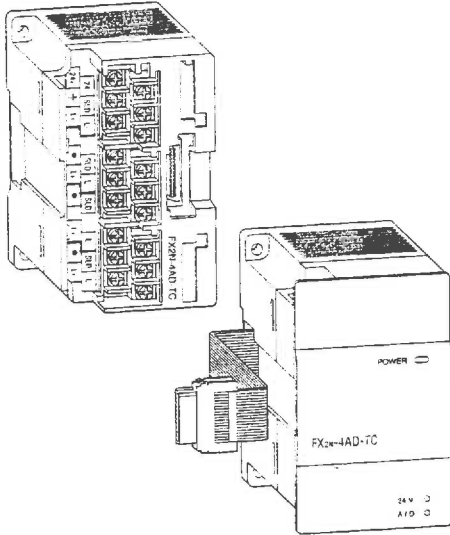
All base units of the FX0N series are now shipped with the 2.0 version of the CPU.

If you already have an FX0N base unit you can check the CPU version by comparing the serial number to the entries in the table on the right.

Version 1.30	Version 2.00	Remark
Up to no. 78□□□□	From no. 78□□□□	For all FX0N base units except FX0N-60MR-DS
Up to no. 79□□□□	From no. 7X□□□□	For FX0N-60MR-DS only

■ Analog Input Module for Thermo Elements FX2N-4AD-TC

☑ FX0N ☑ FX2N
CPU vers. 2.00



This special function module FX2N-4AD-TC is used for processing temperatures. It has 4 independent inputs for detecting signals from thermocouples of types J and K. The type of thermocouple can be chosen independently for each point.

The electrical magnitude at an input is converted into a digital numerical value with a sign. The converted value is stored by the PLC in a memory address, so-called buffer memory, in the module and then converted. It is also possible to calculate

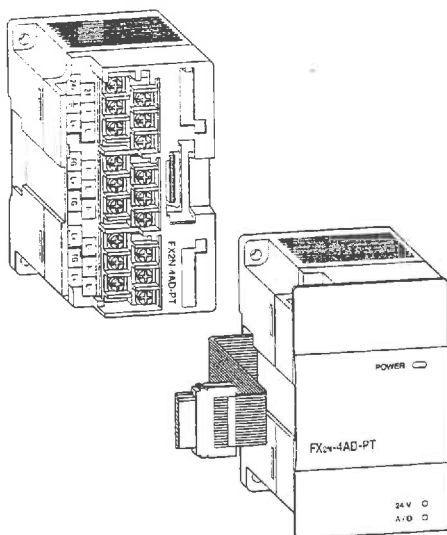
a mean value from a predetermined number of measurements in order to obtain stable digital results.

The number of measurements must be transferred by the PLC program to a buffer memory of the special function module. The value determined is available in another memory address.

Specifications	FX2N-4AD-TC
General specifications	Conforms to FX2N base units
Power supply	5 V DC / 30 mA (from base unit), 24 V DC / 50 mA
Analog inputs	4 (K or J type)
Compensated temperature range	°C -100 – +600 (J type) / -100 – +1200 (K type)
Digital output	-1000 – +6000 (J type) / -1000 – +12000 (K type)
Resolution	°C 0.3 (J type) / 0.4 (K type)
Overall accuracy	±0.5 %
Conversion speed	ms 240 per channel (±2 %)
Related I/O points	8
Weight	kg 0.3
Dimensions (W x H x D)	mm 55 x 90 x 87
Order information	Art. no. 65588

■ Analog Input Module for Pt100 Inputs FX2N-4AD-PT

☑ FX0N ☑ FX2N
CPU vers. 2.00



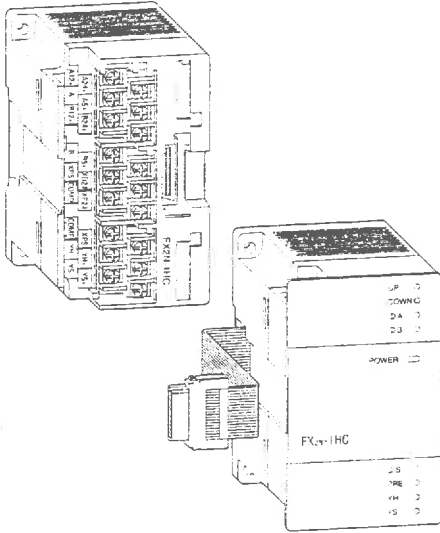
The analog input module FX2N-4AD-PT permits the connection of four Pt100 sensors to the FX controller.

The respective temperatures can be read out either in °C or °F.

Specifications	FX2N-4AD-PT
General specifications	Conforms to FX2N base units
Power supply	5 V DC / 30 mA (from base unit), 24 V DC / 50 mA
Analog inputs	4 (Pt100 sensors)
Compensated temperature ranges	°C -100 – +600
Digital output	-1000 – 6000 (12 bit conversion)
Resolution	°C 0.2
Overall accuracy	±1 % over full linear range
Conversion speed	ms 15 for 4 channels
Related I/O points	8
Weight	kg 0.3
Dimensions (W x H x D)	mm 55 x 90 x 87
Order information	Art. no. 65587

High-Speed Counter FX2N-1HC

FX0N FX2N
CPU vers. 2.00



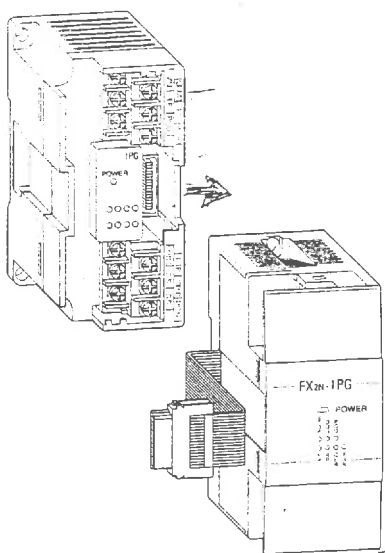
In addition to the internal high-speed MELSEC FX counters, the high-speed counter module FX2N-1HC provides the user with an external hardware counter. It counts 1- or 2-phase pulses up to a frequency of 50 kHz. The counting range covers either 16 or 32 bit.

The two integrated transistor outputs can be switched independently of one another by means of internal comparison functions. Hence, simple positioning tasks can also be realized economically. In addition, the FX2N-1HC can be used as a ring counter.

Specifications	FX2N-1HC
General specifications	Conforms to FX2N base units
Signal level	5, 12, 24 V DC / 7 mA
Power supply	5 V DC / 90 mA from base unit
Counter inputs	2 (1 phase) oder 1 (2 phase)
Max. counting frequency	kHz 50
Input format	Bit 16, 32
Type of counter	Up/down counter, ring counter
Counting range	16 Bit -2147483648 -- -2147483647
	32 Bit 0 - 65535
Output type	2 x transistor (5 - 24 V DC; 0.5 A)
Related I/O points	8
Weight	kg 0.3
Dimensions (W x H x D)	mm 55 x 90 x 87
Order information	Art. no. 65584

Single Axes Positioning Module FX2N-1PG-E

FX0N FX2N
CPU vers. 2.00



The positioning module FX2N-1PG-E is an extremely efficient single-axis positioning module for controlling either step drives or servo drives (by external regulator) with a pulse chain. It is very suitable for achieving accurate positioning in combination with the MELSEC FX series. The configuration and allocation of the position data are carried out directly via the PLC program. A very wide range of manual and automatic functions are available to the user.

Further special features are:

- Possibility of absolute or relative positioning
- 7 different operation functions, such as jog mode, zeroing, variable speeds, etc.
- Separate programming units and operator panels are not required.
- The speed increase or decrease can be set either automatically or manually.

Specifications	FX2N-1PG-E
General specifications	Conforms to FX2N base units
Signal level for digital inputs	24 V DC / 7-40 mA
Power supply	5 - 24 V DC / 60 mA
Accessible axes	1
Output frequency	pulse/s 10 - 100 000
Related I/O points	8
Weight	kg 0.3
Dimensions (W x H x D)	mm 43 x 90 x 87
Order information	Art. no. 65583

Rack mounting stepper drives with AC or DC power input

SD Series drives are full/half and ministepping modules that offer high standards of performance for low to medium power applications. Delivering nominal currents of 2A, 3A and 5A, they provide the benefits of bipolar chopper operation at an economic price.

These rack-mounting modular drives are extremely flexible in terms of power supply requirements and motor compatibility. They may be powered directly from the secondary winding of an isolating transformer or from unregulated DC supplies. Motor current is programmable using simple jumper links on the card, with selectable automatic current reduction at standstill.

On standard drives the resolution is selectable between 200 and 400 steps/rev, whilst the SD15M offers four selectable resolutions up to 4000 steps/rev. This makes it the perfect choice for applications requiring smooth rotation over a wide speed range.

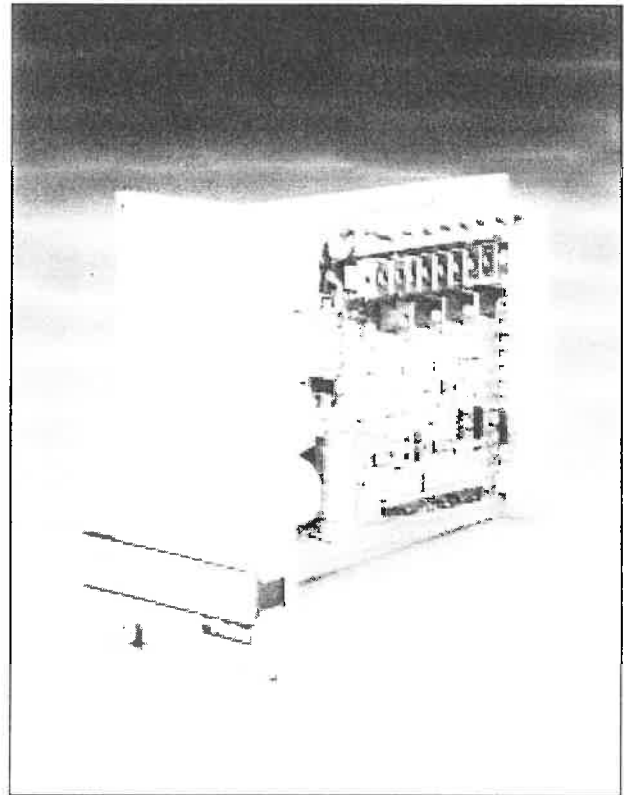
Step and direction pulses may be derived from a user-supplied pulse source, a rack-mounted indexer card, a standalone Parker indexer or an integral clock generator on the drive. The SD15M clock generator compensates automatically for resolution changes so that the preset shaft speed remains constant regardless of the selected resolution.

Features common to the entire SD range include short circuit protection across and between phases, overvoltage protection and a remote de-energise facility. Versions with an integral power dump are available - please see below for further details. A range of pre-wired rack systems is available to house SD Series drives together with optional indexers. Please refer to the information on rack systems later in this catalogue.

Regenerative power dump option

Applications which involve rapid deceleration of high-inertia loads may require that the drive is fitted with a power dump circuit. The SD15MD and WD versions have the same electrical specification as the SD15 but incorporate a power dump with a continuous rating of 15 watts (170 watts peak).

The MD or WD will be required if the deceleration time in seconds is less than $\{J\omega^2 - 0.3\}$ where ω is the maximum speed in revolutions per second and J is the total system inertia in Kg-m². If the expression in brackets is negative, no power dump is required. The power dump is strongly recommended if a 106-size stepper motor is being used.



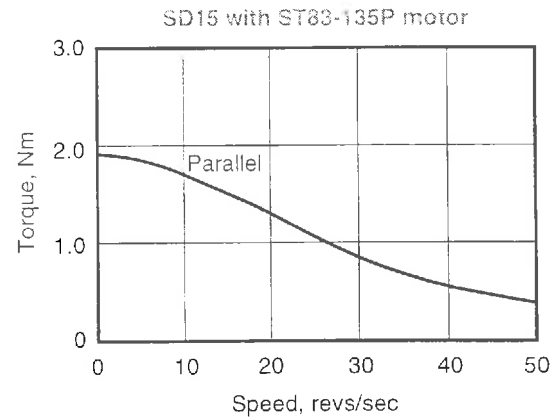
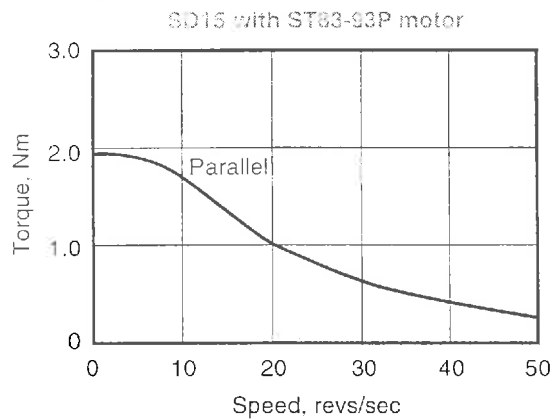
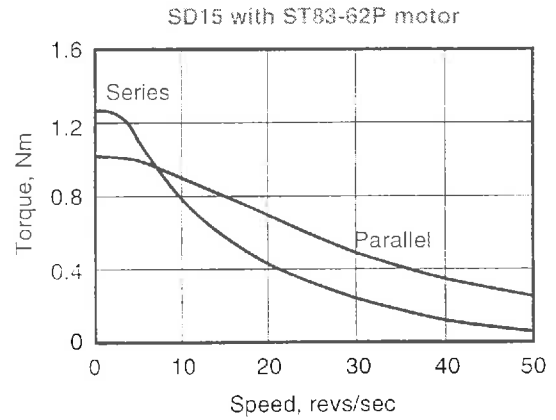
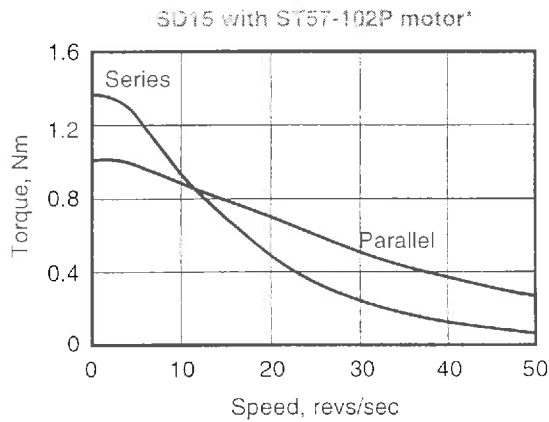
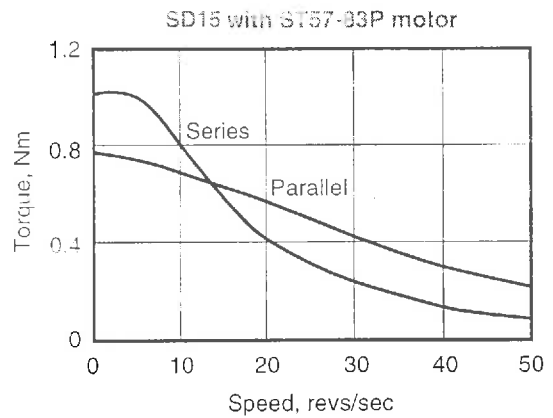
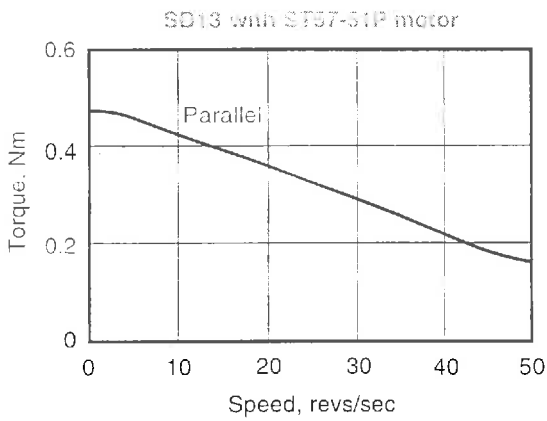
SD series features

- Bipolar switch-mode drive with integral oscillator
- High-reliability, surface mount design
- 60VDC motor supply for improved high-speed torque
- Selectable resolutions up to 4000 steps/rev
- Output current up to 5A per phase
- Operation from AC or DC supplies
- Phase-to-phase short circuit protection
- Motor current programmable by jumper links
- Selectable automatic current reduction at standby
- Regenerative power dump option
- Standard 3U rack mounting
- Compatible with standard 2/4 phase stepper motors

$$t_{dec} < \frac{J\omega^2 - 0.3}{P}$$

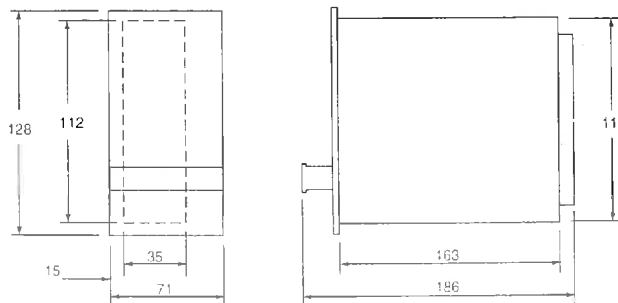
Parameter	Value	
Power		
Motor Supply Voltage	18-0-18 to 44-0-44 VAC or 24 to 30 VDC	
Motor Supply Current	SD12 - 1.5A, SD13 - 2.5A, SD15 & SD15M - 5A	
Motor Supply Torque	18-0-18 VAC or -24 VDC at 150 mA max.	
Amplifier		
Type	Bipolar chopper	
Motor Resolution (steps/rev)	SD12, SD13 & SD15 - 200 or 400 steps/rev SD15M - 400, 1000, 2000 or 4000 steps/rev	
Reverse Circuit Protection	Phase-to-phase	
Nominal Output Current	SD12 2A/phase, SD13 3A/phase, SD15 & SD15M 5A/phase Current may be set to lower values by jumper	
Maximum Stepping Rates	10 kHz @ 200 step/rev, 20 kHz @ 400 step/rev (200 kHz @ 4000 step/rev for SD15M)	
Nominal Chopping Frequency	20 kHz	
Edge Connector	32-way DIN 41612 Type D	
Physical		
Dimensions	See diagram	
Rack mounting data	Front panel 14HP wide, 3U high (see optional panels below)	
Weight	200g	
Command Interface		
SD Drive only		via SD0 (Octo-MUX) board
Input Impedance-current	-K7 to +12VDC	20mA max.
Input Logic Low Level	0.3V	TTL +0.3V
Input Logic High Level	10V to 12V or open circuit	TTL +3.3V
Clock (Step) Input	Low-going pulse, 1.2µS min. (2.5µS for SD15M)	Differential TTL pulse, 10µS min. (1µS for SD15M)
Output Circuit	NPN open-collector, 25V OFF max, 15mA ON max.	isolated NPN open-collector, 12V OFF max, 5mA ON max.
Internal oscillator speed ranges		
Slow (not ramped)	SD12, SD13, SD15	SD15M
Fast (ramped)	30-1000 steps/sec	3-130 rpm (independent of resolution)
Acceleration Time	600-20,000 steps/sec	30-3,000 rpm (independent of resolution)
Deceleration Time	60 mS	
Deceleration Time	30 mS - acceleration and deceleration times may be increased by additional capacitor	
Environmental		
Operating Temperature	0° to 50° C	
Storage Temperature	-40° to 85° C	
Humidity	0 to 95% non-condensing	
Motors		
Type	2 phase hybrid; 4, 6 or 8 leads	
Minimum Inductance	1 mH	
Recommended induct. range	1 - 10mH	
Optional front panels	FP36 for SD12; FP37 for SD13; FP38 for SD15; FP48 for SD15M	
Transformers	T0193 (300VA); T0194 (600VA)	

Typical performance data



* Series connection available only on flying lead version of this motor

SD dimensions (mm)



A range of pre-assembled rack systems for SD and CD series stepper drives

To simplify the installation and commissioning of rack-mounted drive modules, Parker offer a range of pre-assembled rack systems to house up to six drives. The system is based on individual motherboards for each drive, with additional motherboards catering for power supplies and control cards. Each rack system is fitted with the appropriate number of motherboards according to the module complement, allowing all external connections to be made via screw terminals or plug-in connectors.

All assemblies are based on the standard Euro-rack system and are 3U high (132mm). Module and panel widths are always quoted in horizontal pitches (HP), the width of a standard rack being 84HP. For example, SD drive front panels are 14HP wide, so six panels will occupy the full rack width of 84HP.

All drive motherboards used in SC and CN series racks incorporate opto-isolation of the control signals, making them directly compatible with the Parker indexers listed later in this catalogue.

SC series racks for SD drives

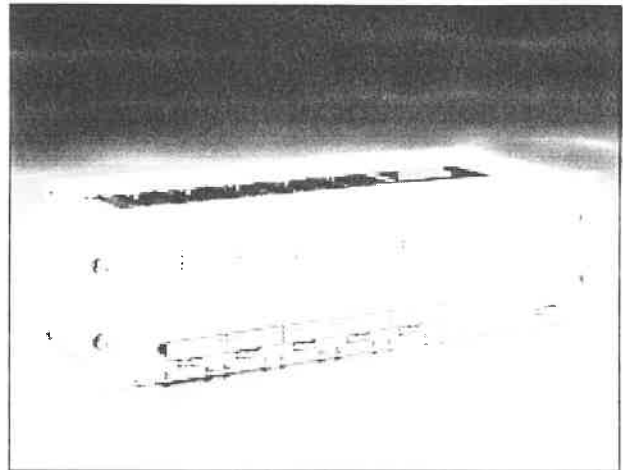
The SD drive operates directly from an isolating transformer and therefore requires no separate power supply module. This allows up to six SD drives to be housed in a single SC series rack. When combined with IFX indexer cards, the maximum number of axes per rack is three.

CN series racks for CD drives

CD60M and CD80M drives require a separate PM2000CN power supply module, which is 24HP wide. This still allows room for six drives in a CN series rack since each drive front panel is only 10HP wide. When combined with IFX indexer cards, up to three axes can be accommodated in a single rack.

PM2000CN power supply module

The PM2000CN offers a convenient and economic method of powering CD series drives. All necessary components are included with the exception of the mains transformer which is mounted separately from the rack. The power module has an output current rating of 16A DC at a bus voltage of 120V. A regenerative power dump circuit is included, and the module comes complete with a 24HP front panel. The AC input to the PM2000CN may be single or three phase.



Transformers

There is a range of standard transformers available to power SD or CD series drives. The required transformer size depends partly on motor size, shaft power demand and duty cycle, however the quantities listed in the table form a useful guide. These are based on no more than half the axes being in motion at any one time.

All single-phase transformers may be operated from 115V and 230V supplies. Models TO181 and TO182 have four-winding primaries which also permit line-to-line connection on 400V 3-phase supplies.

Ordering information

SC and CN racks can be supplied as drive-only versions or for drives with indexer cards. Please refer to the order codes and select the appropriate rack according to the number of axes required. Standard rack systems do not allow for a mix of drives with and without indexer cards, however other combinations are available to special order. The rack order code refers to the rack assembly only; all plug-in drive modules, power supplies, front panels and transformers are ordered separately. When all parts are ordered at the same time, the rack will be supplied with all modules and front panels fitted.

Order codes

SC: for SD drives
CN: for CD drives
Number of axes

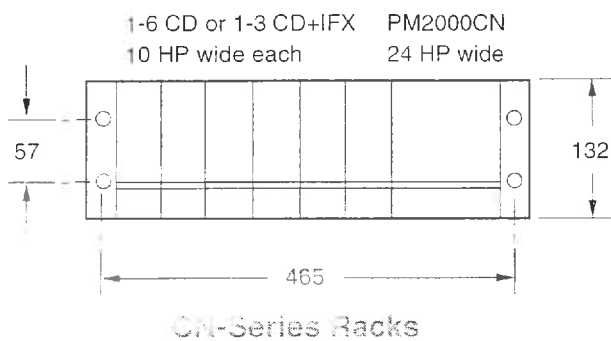
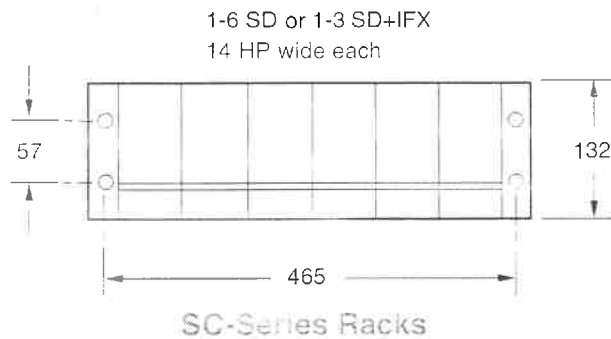
SC 1 0
0: drives only
2X/RS232C: drives with IFX indexer cards

Example - SC30: three SD drives

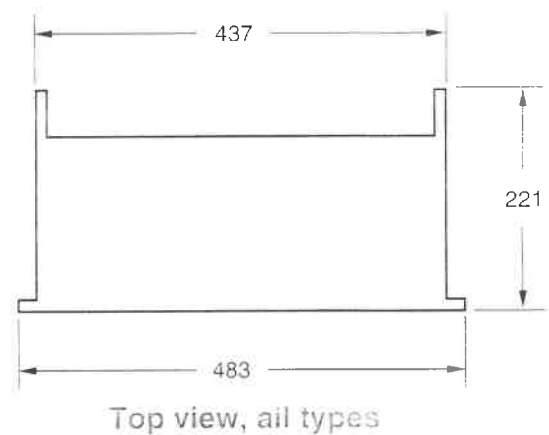
CN22X/RS232C: two CD drives with IFX cards

SC/CN Series

rack systems & power supplies



SC & CN rack dimensions (mm)



Ordering front panels

Optional front panels are available for all drive modules as well as blank panels to cover control cards or unused spaces in the rack. Please note that these panels are not included with the drive and must be ordered separately if required.

Panel type	Width	Part No.
SD12	14HP	FP36
SD13	14HP	FP37
SD15	14HP	FP38
SD15M	14HP	FP48
CD60M	10HP	FP52
CD80M	10HP	FP53
Blank	10HP	FP55
Blank	14HP	FP39
Blank	20HP	FP54

Completing the rack

The table below shows the number of blank panels required for each rack type in order to cover control cards or empty slots in the rack.

Rack type	Number of blank panels required		
	FP55	FP39	FP54
SC10, SC12X		5	
SC20, SC22X		4	
SC30, SC32X		3	
SC40		2	
SC50		1	
CN10, CN12X	1		2
CN20			2
CN30	1		1
CN40			1
CN50	1		
CN22X	2		1
CN32X	3		

Transformers for rack-mounting drives

	Models for SD drives		Models for CD drives			
	TO193	TO194	TO181	TO182	TO185	TO186
AC input range	110-240v	110-240v	90-480v	90-480v	180-480v	180-480v
Single/three phase	1	1	1	1	3	3
DC bus voltage	60	60	120	120	120	120
VA rating	300	600	2600	1200	2500	5000
Max. number of drives*	3 x SD12 2 x SD13 1 x SD15	6 x SD12 4 x SD13 2 x SD15	8 x CD60M 6 x CD80M	4 x CD60M 3 x CD80M	8 x CD60M 6 x CD80M	11 x CD60M 8 x CD80M
Dimensions w x d x h	117x117x181	126x168x212	172x215x261	135x195x234	302x132x319	380x140x359
Weight kg	3.5	7.5	25	16	34	50

* Assumes 50% duty cycle. Larger numbers of drives may be accommodated at lower duty cycles