



Escola Politécnica da Universidade de São Paulo
PMR – Departamento de Engenharia Mecatrônica e Sistemas Mecânicos
PMR2550 – Projeto de Conclusão de Curso 2

Projeto de Conclusão de Curso

Projeto e Construção de uma Perna Mecatrônica Relatório Final

Nome: Elias Davidsom Bardini Vaz

NUSP: 4939943

Orientador: Prof. Dr. Marcos Ribeiro Pereira Barreto

São Paulo, Fevereiro de 2009

Índice

Introdução	03
Objetivos	06
Metodologia	06
Análises Cinemática e Dinâmica	15
Construção	21
Resultados	23
Conclusão	23
Anexos	24
Referências Bibliográficas	42

Introdução

Hoje em dia existe muita atenção voltada à melhoria da qualidade de vida de pessoas deficientes, com ambientes adaptados, equipamentos de auxílio, et cetera. Tendo isso em vista, o desenvolvimento de uma máquina para simular e imitar o andar humano é de grande interesse atualmente. Além disso, há grande interesse em construir autômatos de forma humana, mais amistosos ao olhar humano, para, por exemplo, servirem de companhia para idosos e pessoas portadoras de deficiência. Existe também o interesse em desenvolver meios mecanizados de locomoção para inspeção e exploração de lugares inacessíveis e/ou inóspitos. Para esses fins, podem-se utilizar tanto robôs movimentados por rodas quanto movimentados por pernas. Entretanto, os primeiros não conseguem lidar com terrenos muito irregulares, dada sua baixa capacidade de adaptação, sendo necessário algo mais versátil, como pernas. E essa é a vantagem das pernas: sua versatilidade e adaptabilidade permitem que se tenha acesso aos mais variados lugares, sem q seja necessário prepara-los antes (aplainando o terreno, construindo trilhos, etc.).

Atualmente já existem alguns tipos de pernas mecatrônicas desenvolvidas para diferentes tipos de tarefas. As imagens a seguir mostram alguns tipos já desenvolvidos dessas pernas.

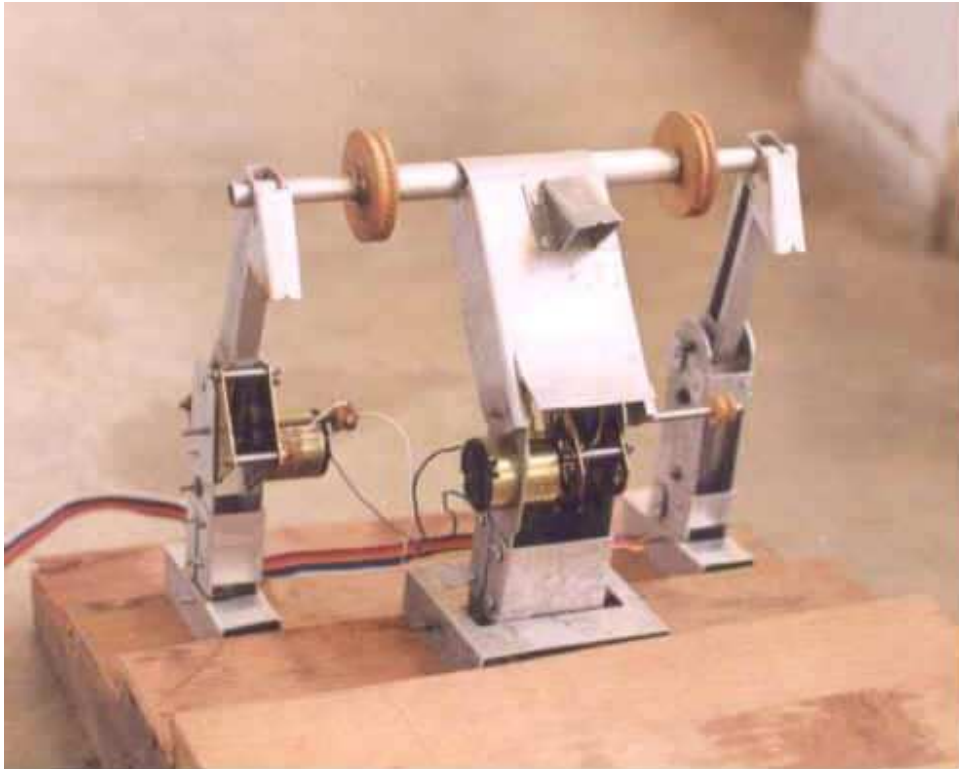


Figura 1: Mecanismo de perna para subir escadas



Figura2: Perna simples

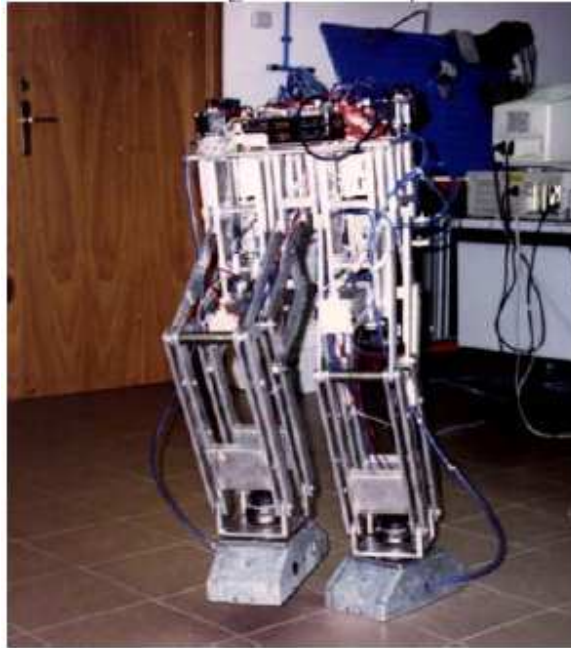


Figura 3: Perna sensorizada com associação de mecanismos



Figura 4: Robô dançante da Sony

Entretanto, o andar humano é algo muito difícil de copiar, especialmente considerando sua versatilidade (capacidade de se adaptar à irregularidade dos terrenos, flexibilidade nas mudanças de direção e posição) e complexidade (grande número de músculos, ossos e nervos envolvidos no controle e atuação do movimento).

Baseando-nos nos artigos italianos ¹ e ², tentaremos simplificar esse movimento, limitando-o para tarefas mais cotidianas como o andar simples (unidirecional, em solos regulares) e imaginando que para posterior aplicação, podemos utilizar este mecanismo em conjunto com as pernas humanas para melhor controle do movimento. Portanto, ao desenvolver algo mais simples, podemos abrir um grande leque para pessoas que perderam uma perna.

Objetivos

Este projeto consistirá no desenvolvimento de mecanismo semelhante a um par de pernas humanas e construção de um protótipo funcional. O movimento de andar será feito com articulações semelhantes às humanas, mas de maneira simplificada, de modo a adequar-se ao andar humano e eventualmente ser evoluído para uma prótese adaptada. Serão feitos a modelagem e um programa para simulação e um protótipo reduzido do mecanismo.

Metodologia

Para conseguirmos desenvolver a proposta acima utilizaremos a teoria de projeto de mecanismos. Para tanto, iremos rever alguns conceitos dessa teoria, para depois podermos aplicá-la sem maiores explicações.

Grau de liberdade de um sistema é o número de parâmetros independentes necessários para definir a posição do sistema em qualquer instante de tempo. Para um corpo rígido plano, por exemplo, sem restrições de movimento, teremos três graus de liberdade, pois são necessários dois comprimentos (distâncias de um certo ponto do corpo em relação

a eixos linearmente independentes de uma referencia, ou origem) além de um ângulo para sabermos a posição relativa de rotação do próprio corpo. Porém, também podemos encarar grau de liberdade como o número de entradas necessárias para podermos prever a saída, ou seja, quantos atuadores (motores) são necessários para conseguirmos certo resultado. Daqui temos que, conseguindo os mínimos requisitos necessários para termos a saída desejada, devemos buscar reduzir ao máximo o número de graus de liberdade, para necessitarmos do menor número possível de atuadores, simplificando assim o projeto.

Para o cálculo do grau de liberdade de qualquer mecanismo temos a equação de Gruebler (modificada por Kutzbach):

$$M = 3(L - 1) - 2J_1 - 2J_2$$

Na qual: M = grau de liberdade ou mobilidade

L = número de *links*

J_1 = número de junções de um grau de liberdade

J_2 = número de junções de dois grau de liberdade

Mas para utilizarmos essa equação temos que ao menos definir a forma geral do mecanismo a ser utilizado. Para isso vamos analisar o problema do andar. Como explicado acima, trataremos de um andar simplificado, em linha reta, com terreno regular, portanto cada uma das pernas pode ser tratada como um mecanismo plano. Além disso, conseguimos enxergar neste tipo de movimento uma repetição contínua, o que é ótimo, pois poderemos modelá-lo como um movimento cíclico, repetitivo. Conseqüentemente ao analisarmos lateralmente o movimento esperado do pé, com referencial no quadril, teremos uma curva fechada, com um lado reto (pé pisando, parado em relação ao chão) e outro curvo (pé no ar, se movendo em relação ao chão). Essa curva fechada é uma das muitas que pode ser feita acompanhando-se um ponto de um *coupler* (um *link* não conectado a uma junta fixa). O mecanismo mais simples que possui um *coupler* é o mecanismo de **quatro barras**, portanto ele será utilizado para modelarmos nossa perna.

Para termos esse tipo de curva fechada de maneira cíclica, precisaremos de um movimento repetitivo e sem inversão de fase. Para isso o melhor é termos um dos *links* como um *crank* (um *link* preso a uma junta fixa, que pode dar uma volta completa em torno

dela). Para conseguirmos isso devemos observar a condição de Grashof.

Sejam: S = comprimento do *link* maior
 L = comprimento do *link* menor
 P = comprimento de um dos *links* restantes
 Q = comprimento do outro *link* restante

Então: $S + L \leq P + Q$

Caso a condição acima seja satisfeita temos uma ligação de Grashof e ao menos um dos *links* consegue dar uma volta completa. Então essa condição deve ser satisfeita para obtermos o movimento desejado.

Analisando agora o formato da curva, de acordo com a tabela abaixo, podemos ter uma idéia da relação entre os tamanhos relativos entre os *links* a serem utilizados, ela será usada para comparação com o resultado analítico.

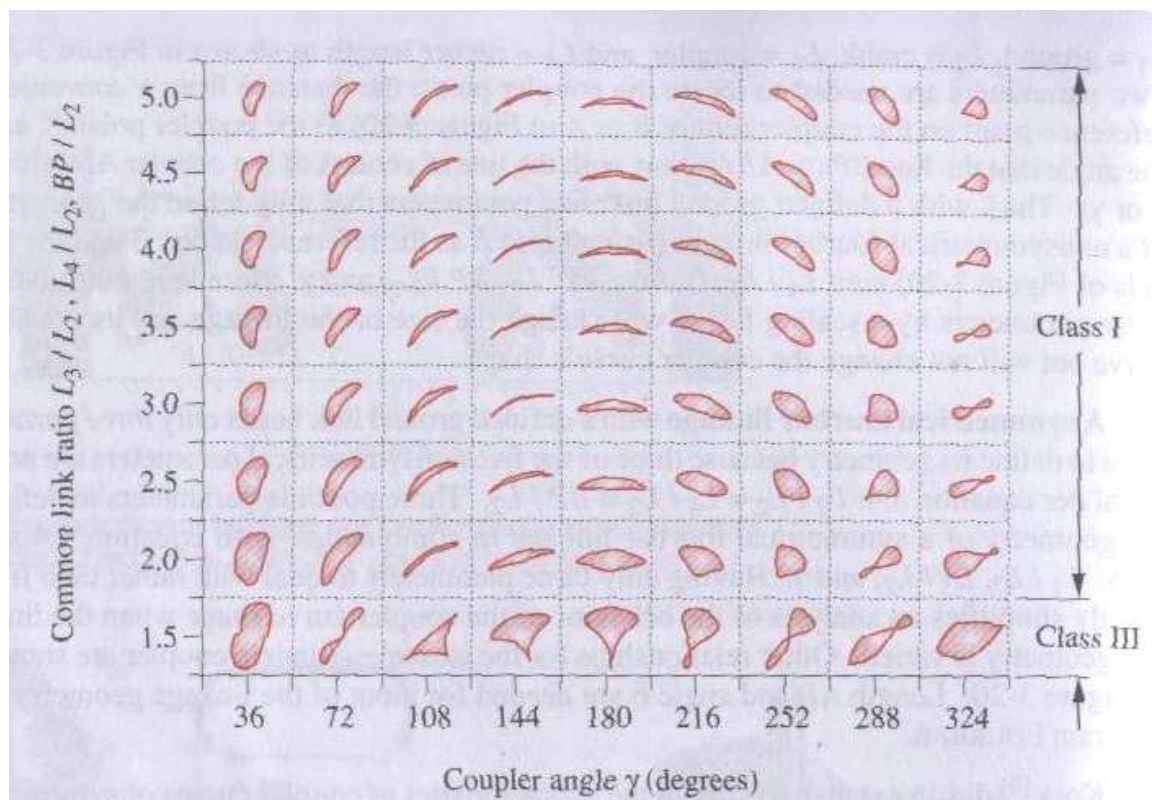


Figura 5: Tabela de curvas, de acordo com a relação entre links

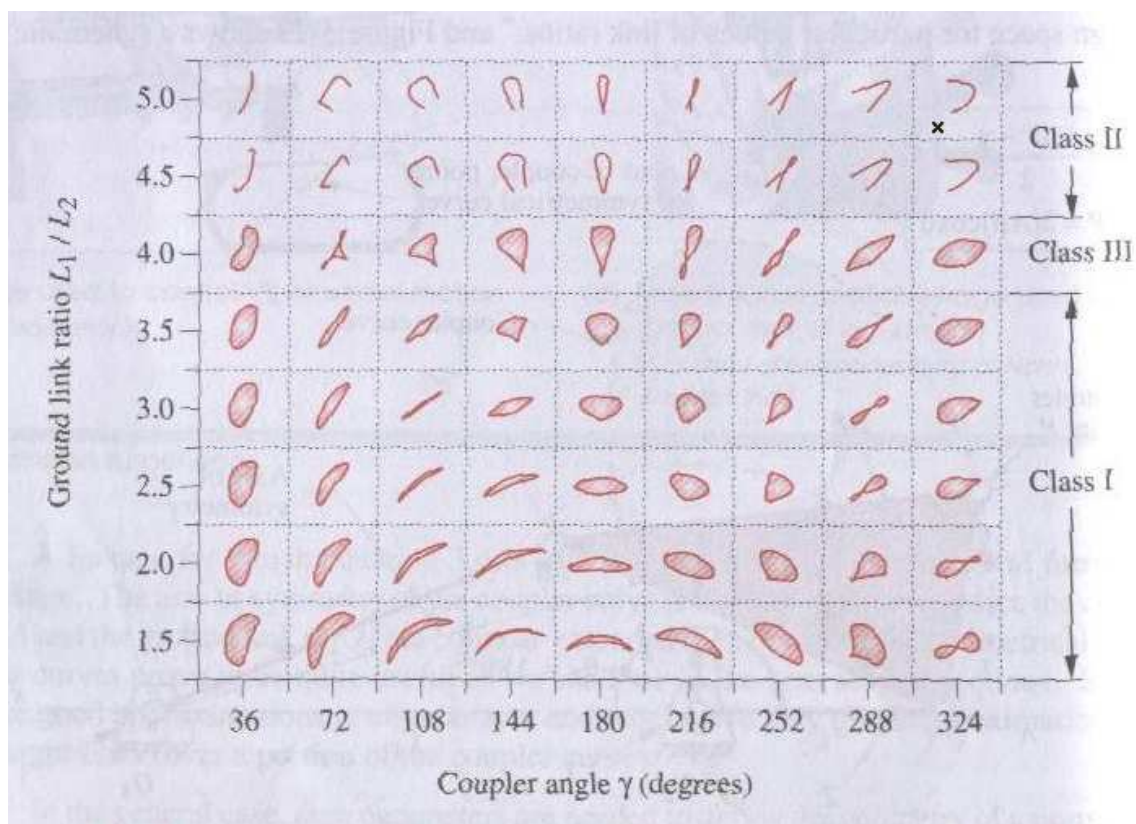


Figura 6: Tabela de curvas, de acordo com a relação entre ground e crank

Agora que definimos nosso mecanismo como um quatro-barras de Grashof, podemos fazer a síntese analítica do mesmo e calcular assim os parâmetros do mesmo.

Primeiramente faremos a solução literal de um mecanismo genérico de quatro barras e depois substituiremos nossos parâmetros para obtermos a nossa solução.

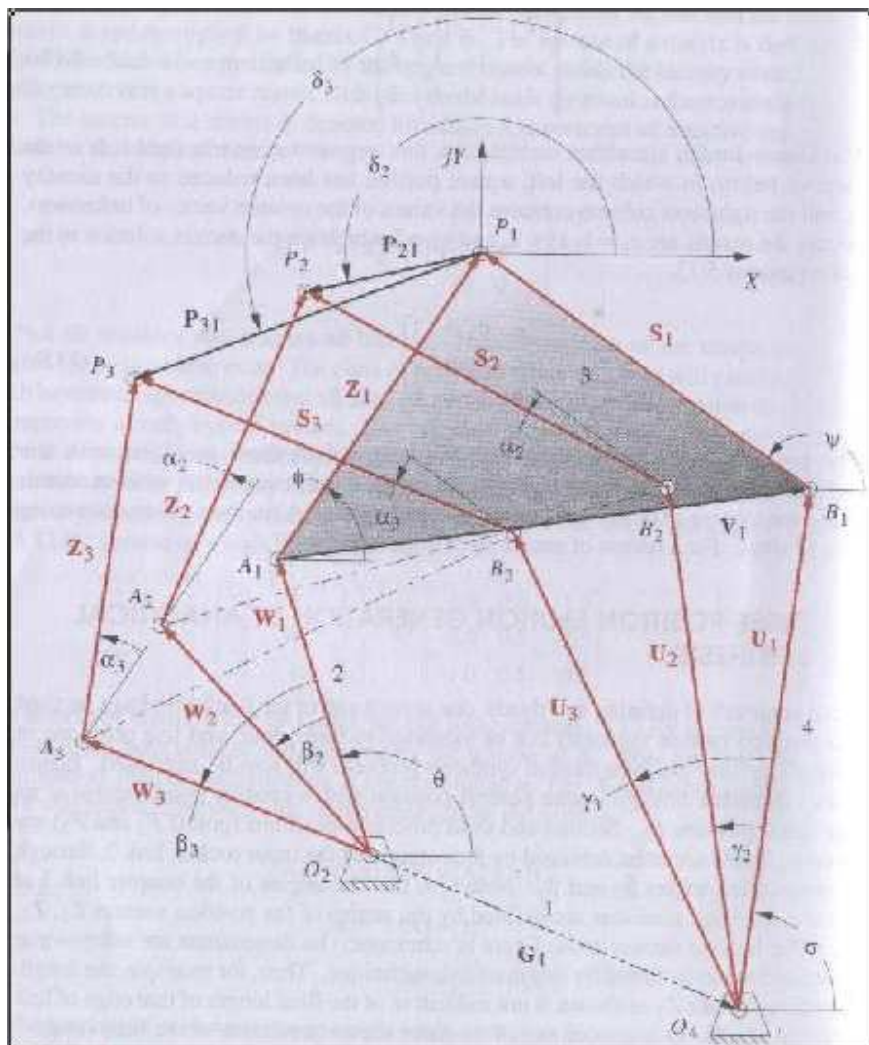


Figura 7: Esquema de um mecanismo genérico de quatro barras

Como mostrado na figura 7 modelaremos as três posições a partir da primeira, definida como origem do nosso sistema de coordenadas e as outras posições relativas à primeira.

Para modelarmos este mecanismo analiticamente segundo as posições desejadas devemos dividi-lo em duas partes. Primeiramente modelaremos o lado esquerdo do mecanismo e seus vetores W e Z .

Fazendo as equações para irmos do ponto 1 para o ponto 2 e do ponto 1 para o ponto 3, temos:

$$W_2 + Z_2 - P_{21} - Z_1 - W_1 = 0$$

$$W_3 + Z_3 - P_{31} - Z_1 - W_1 = 0$$

Substituindo pelos equivalentes complexos:

$$we^{j(\theta+\beta_2)} + ze^{j(\phi+\alpha_2)} - p_{21}e^{j\delta_2} - ze^{j\phi} - we^{j\theta} = 0$$

$$we^{j(\theta+\beta_3)} + ze^{j(\phi+\alpha_3)} - p_{31}e^{j\delta_3} - ze^{j\phi} - we^{j\theta} = 0$$

Reescrevendo a soma dos ângulos nos expoentes, como o produto dos termos:

$$we^{j\theta}e^{j\beta_2} + ze^{j\phi}e^{j\alpha_2} - p_{21}e^{j\delta_2} - ze^{j\phi} - we^{j\theta} = 0$$

$$we^{j\theta}e^{j\beta_3} + ze^{j\phi}e^{j\alpha_3} - p_{31}e^{j\delta_3} - ze^{j\phi} - we^{j\theta} = 0$$

Simplificando e rearranjando:

$$we^{j\theta}(e^{j\beta_2} - 1) + ze^{j\phi}(e^{j\alpha_2} - 1) = p_{21}e^{j\delta_2}$$

$$we^{j\theta}(e^{j\beta_3} - 1) + ze^{j\phi}(e^{j\alpha_3} - 1) = p_{31}e^{j\delta_3}$$

A magnitude w dos vetores W_1 , W_2 e W_3 representam o comprimento de uma barra rígida, portanto têm o mesmo valor, assim como os vetores Z_1 , Z_2 e Z_3 que têm magnitude z .

Podemos separar as equações acima em partes real e imaginária, utilizando Euler.

Parte real:

$$w \cos \theta (\cos \beta_2 - 1) - w \sin \theta \sin \beta_2 + z \cos \phi (\cos \alpha_2 - 1) - z \sin \phi \sin \alpha_2 = p_{21} \cos \delta_2$$

$$w \cos \theta (\cos \beta_3 - 1) - w \sin \theta \sin \beta_3 + z \cos \phi (\cos \alpha_3 - 1) - z \sin \phi \sin \alpha_3 = p_{31} \cos \delta_3$$

Parte imaginária (já dividida por j):

$$w \sin \theta (\cos \beta_2 - 1) + w \cos \theta \sin \beta_2 + z \sin \phi (\cos \alpha_2 - 1) + z \cos \phi \sin \alpha_2 = p_{21} \sin \delta_2$$

$$w \sin \theta (\cos \beta_3 - 1) - w \cos \theta \sin \beta_3 + z \sin \phi (\cos \alpha_3 - 1) - z \cos \phi \sin \alpha_3 = p_{31} \sin \delta_3$$

Há doze variáveis nestas quatro equações: $w, \theta, \beta_2, \beta_3, z, \phi, \alpha_2, \alpha_3, p_{21}, p_{31}, \delta_2$ e δ_3 . Seis delas $\alpha_2, \alpha_3, p_{21}, p_{31}, \delta_2$ e δ_3 serão definidas de acordo com os valores escolhidos para o movimento. Das seis que restam $w, \theta, \beta_2, \beta_3, z, \phi$, serão assumidos valores de duas para podermos encontrar as soluções das outras quatro. Escolheremos os valores de β_2 e β_3 e a partir deles encontraremos w, θ, z e ϕ , os valores que definem os vetores W e Z .

Resta descobrir os ângulos e magnitudes dos vetores W e Z (w, θ, z, ϕ). Para simplificar, utilizaremos as componentes x e y dos vetores W e Z no lugar de suas coordenadas polares.

$$\begin{aligned} W_{1x} &= w \cos \theta & Z_{1x} &= z \cos \phi \\ W_{1y} &= w \sin \theta & Z_{1y} &= z \sin \phi \end{aligned}$$

Substituindo os valores acima nas equações anteriores:

$$\begin{aligned} W_{1x}(\cos \beta_2 - 1) - W_{1y} \sin \beta_2 + Z_{1x}(\cos \alpha_2 - 1) - Z_{1y} \sin \alpha_2 &= p_{21} \cos \delta_2 \\ W_{1x}(\cos \beta_3 - 1) - W_{1y} \sin \beta_3 + Z_{1x}(\cos \alpha_3 - 1) - Z_{1y} \sin \alpha_3 &= p_{31} \cos \delta_3 \\ W_{1y}(\cos \beta_2 - 1) - W_{1x} \sin \beta_2 + Z_{1y}(\cos \alpha_2 - 1) - Z_{1x} \sin \alpha_2 &= p_{21} \sin \delta_2 \\ W_{1y}(\cos \beta_3 - 1) - W_{1x} \sin \beta_3 + Z_{1y}(\cos \alpha_3 - 1) - Z_{1x} \sin \alpha_3 &= p_{31} \sin \delta_3 \end{aligned}$$

Na forma matricial:

$$\begin{bmatrix} \cos \beta_2 - 1 & -\sin \beta_2 & \cos \alpha_2 - 1 & -\sin \alpha_2 \\ \cos \beta_3 - 1 & -\sin \beta_3 & \cos \alpha_3 - 1 & -\sin \alpha_3 \\ \sin \beta_2 & \cos \beta_2 - 1 & \sin \alpha_2 & \cos \alpha_2 - 1 \\ \sin \beta_3 & \cos \beta_3 - 1 & \sin \alpha_3 & \cos \alpha_3 - 1 \end{bmatrix} \begin{bmatrix} W_{1x} \\ W_{1y} \\ Z_{1x} \\ Z_{1y} \end{bmatrix} = \begin{bmatrix} p_{21} \cos \delta_2 \\ p_{31} \cos \delta_3 \\ p_{21} \sin \delta_2 \\ p_{31} \sin \delta_3 \end{bmatrix}$$

Com isso temos a solução da síntese de três posições para o lado esquerdo do mecanismo, para quaisquer valores de β_2 e β_3 . Agora devemos repetir o processo para o lado direito do mecanismo e encontrar os vetores U e S . Da figura 7 temos as três posições e os ângulos $\sigma, \gamma_2, \gamma_3, \psi, \alpha_2, \alpha_3$. A solução para U e S será idêntica à para W e Z , alterando os nomes dos vetores e ângulos. As equações de laço dos vetores são:

$$\begin{aligned} U_2 + S_2 - P_{21} - S_1 - U_1 &= 0 \\ U_3 + S_3 - P_{31} - S_1 - U_1 &= 0 \end{aligned}$$

Substituindo, simplificando e rearranjando:

$$ue^{j\sigma}(e^{j\gamma_2} - 1) + se^{j\psi}(e^{j\alpha_2} - 1) = p_{21}e^{j\delta_2}$$

$$ue^{j\theta}(e^{j\gamma_3} - 1) + se^{j\psi}(e^{j\alpha_3} - 1) = p_{31}e^{j\delta_3}$$

Analogamente à solução para o lado esquerdo, nós assumiremos valores para os ângulos γ_2 e γ_3 . Os ângulos α_2 e α_3 são os mesmos que para o lado esquerdo. Significa que encontraremos os ângulos σ e ψ a partir das componentes x e y dos vetores U e Z . O equacionamento, na forma matricial, fica:

$$\begin{bmatrix} \cos \gamma_2 - 1 & -\sin \gamma_2 & \cos \alpha_2 - 1 & -\sin \alpha_2 \\ \cos \gamma_3 - 1 & -\sin \gamma_3 & \cos \alpha_3 - 1 & -\sin \alpha_3 \\ \sin \gamma_2 & \cos \gamma_2 - 1 & \sin \alpha_2 & \cos \alpha_2 - 1 \\ \sin \gamma_3 & \cos \gamma_3 - 1 & \sin \alpha_3 & \cos \alpha_3 - 1 \end{bmatrix} \begin{bmatrix} U_{1x} \\ U_{1y} \\ S_{1x} \\ S_{1y} \end{bmatrix} = \begin{bmatrix} p_{21} \cos \delta_2 \\ p_{31} \cos \delta_3 \\ p_{21} \sin \delta_2 \\ p_{31} \sin \delta_3 \end{bmatrix}$$

Assim, é só definirmos os valores das posições desejadas para que tenhamos suas respectivas soluções e possamos encontrar uma curva que passe pelos três pontos escolhidos.

Entretanto, a curva encontrada, apesar de satisfazer os três pontos escolhidos, pode não ser a melhor para desenvolver as passadas. Isso é percebido nas tabelas de curvas, nas quais vemos que o formato delas está, em sua maioria, de “cabeça para baixo” em relação ao movimento natural das passadas. Caso necessário, dependendo da curva apresentada pelas métricas finais, devido às características das curvas resultantes destes mecanismos, vamos considerar uma alternativa à montagem direta do mecanismo: um paralelogramo para a inversão, mas mantendo a semelhança, do movimento de rotação.

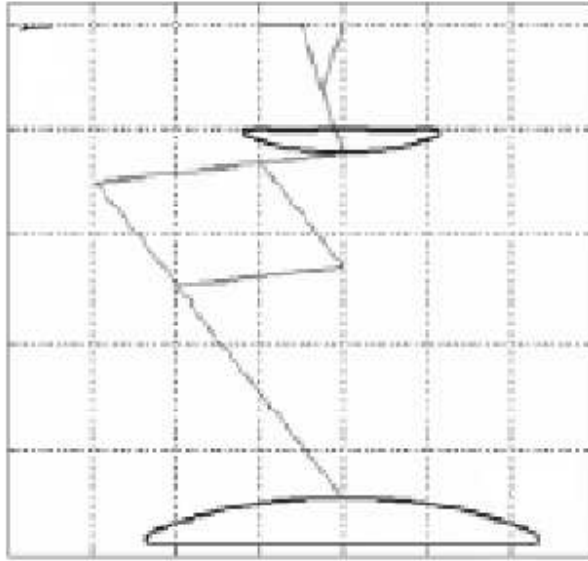


Figura 8: Inversão e amplificação do movimento com o uso de um paralelogramo

Utilizando este acréscimo no mecanismo, teremos uma curva mais semelhante ao passo pretendido (reta embaixo e convexa em cima), entretanto, ocorre uma sofisticação na montagem do sistema.

A modelagem para este tipo de montagem é a mesma da anterior, alterando-se apenas os pontos desejados para a curva de saída. De acordo com os resultados da análise cinemática, será definida a estrutura.

Para o cálculo dos esforços foi feito o seguinte equacionamento:

POSIÇÃO:

Q2: ângulo de L2 “crank”

$$E = \sqrt{L_2^2 + L_1^2 - L_1 * L_2 * \cos(Q_2)}$$

$$\beta = \frac{a \cos(E^2 + L_1^2 - L_2^2)}{2 * E * L_1}$$

$$\psi = \frac{a \cos(E^2 + L_3^2 - L_4^2)}{2 * E * L_3}$$

$$\lambda = \frac{a \cos(E^2 + L_4^2 - L_3^2)}{2 * E * L_4}$$

$$\theta_4 = \pi - \lambda - \beta$$

$$\theta_3 = \psi - \beta$$

$$\gamma = \pi - \psi - \lambda$$

VELOCIDADE

Equação vetorial: $V_A + V_{BA} = V_B$

$$L_2 * j * W_2 * e^{j*Q_2} + L_3 * j * W_3 * e^{j*\theta_3} = L_4 * j * W_4 * e^{j*\theta_4}$$

Em X:

$$-L_3 * \text{sen}(\theta_3) * W_3 + L_4 * \text{sen}(\theta_4) * W_4 = L_2 * W_2 * \text{sen}(Q_2)$$

Em Y:

$$L_3 * \cos(\theta_3) * W_3 - L_4 * \cos(\theta_4) * W_4 = -L_2 * W_2 * \cos(Q_2)$$

ACELERAÇÃO

Em X:

$$\begin{aligned} & -L_3 * \text{sen}(\theta_3) * \alpha_3 + L_4 * \text{sen}(\theta_4) * \alpha_4 = \\ & = -L_4 * W_4^2 * \cos(\theta_4) + L_2 * \alpha_2 * \text{sen}(Q_2) + L_2 * W_2^2 * \cos(Q_2) + L_3 * W_3^2 * \cos(\theta_3) \end{aligned}$$

Em Y:

$$\begin{aligned} & L_3 * \cos(\theta_3) * \alpha_3 + L_4 * \cos(\theta_4) * \alpha_4 = \\ & = -L_4 * W_4^2 * \text{sen}(\theta_4) - L_2 * \alpha_2 * \cos(Q_2) + L_2 * W_2^2 * \text{sen}(Q_2) + L_3 * W_3^2 * \text{sen}(\theta_3) \end{aligned}$$

Análises Cinemática e Dinâmica

Para analisarmos a curva de saída e as tensões no mecanismo das possíveis montagens, foi desenvolvido um programa, utilizando o software Matlab, para a plotagem de diversos pontos das curvas, além do cálculo dos esforços nas junções e do torque requerido do motor. O código do programa é apresentado abaixo após as curvas e respectivas análises.

Para a primeira modelagem (direta, sem o auxílio de um paralelogramo), obtivemos

as seguintes curvas de posição (L1=ground, L2=crank, L3=coupler, L4=rocker, P=distância do ponto P à junção L2-L3, Phi=ângulo entre PL2-L3 e L2-L3L3L4):

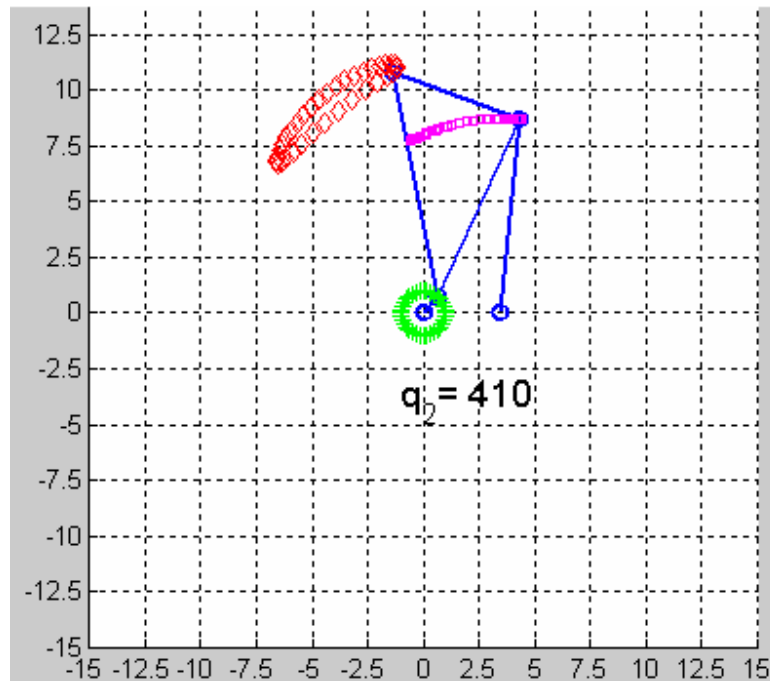


Figura 9: Curva (em vermelho) da saída do mecanismo

L3 = 8,75
L2 = 1
L4 = 8,75
L1 = 3,5
P = 10,29
Phi = 36°

Observamos na curva acima que a “barriga” da curva está na posição contrária ao esperado. A curva ainda atende os pontos de precisão desejados, entretanto o mecanismo pode não se comportar como o esperado em razão disso e das velocidades em cada parte do passo (pisando e no ar), pois notam-se no gráfico os pontos “no ar” têm uma velocidade maior, são mais espaçados entre si do que os pontos “do chão”. Isso pode tanto prejudicar quanto ser interessante ao funcionamento, faremos a simulação de posição para outros valores, a fim de termos uma opção melhor ao fazer a análise dinâmica.

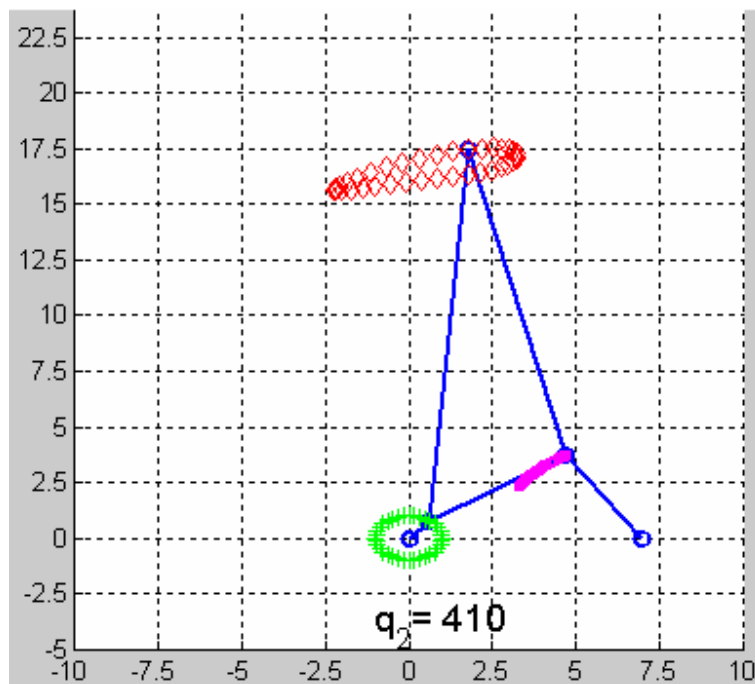


Figura 10: Curva (em vermelho) da saída do mecanismo

$L3 = 5$
 $L2 = 1$
 $L4 = 4,375$
 $L1 = 7$
 $P=16,75$
 $\Phi=50^\circ$

Nesta curva, conseguimos, apesar do passo mais achatado, uma curva mais próxima às características desejadas para um passo, em termos de velocidade e da forma geral da curva.

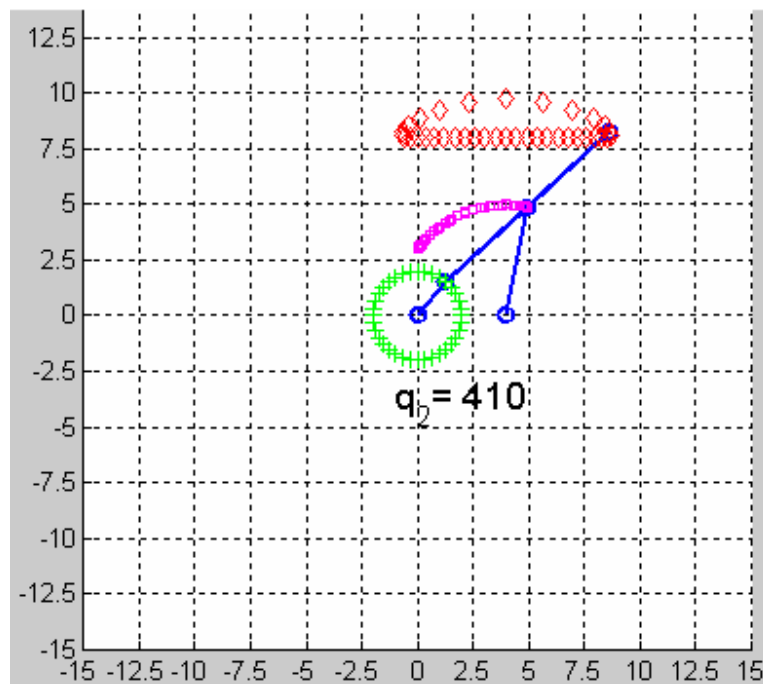


Figura 11: Curva (em vermelho) da saída do mecanismo

$L3 = 5$
 $L2 = 2$
 $L4 = 5$
 $L1 = 4$
 $P = 10$
 $\Phi = 0$

Para a segunda modelagem (com auxílio de paralelogramo), obtivemos a curva de posição acima para a saída do mecanismo principal (ainda sem a participação do paralelogramo, que amplificará a curva e alterará o sentido da rotação). Percebemos que o espaçamento dos pontos é muito diferente entre as duas partes da curva. Isso pode levar a trancos indesejados, devido à diferença entre as velocidades em cada parte.

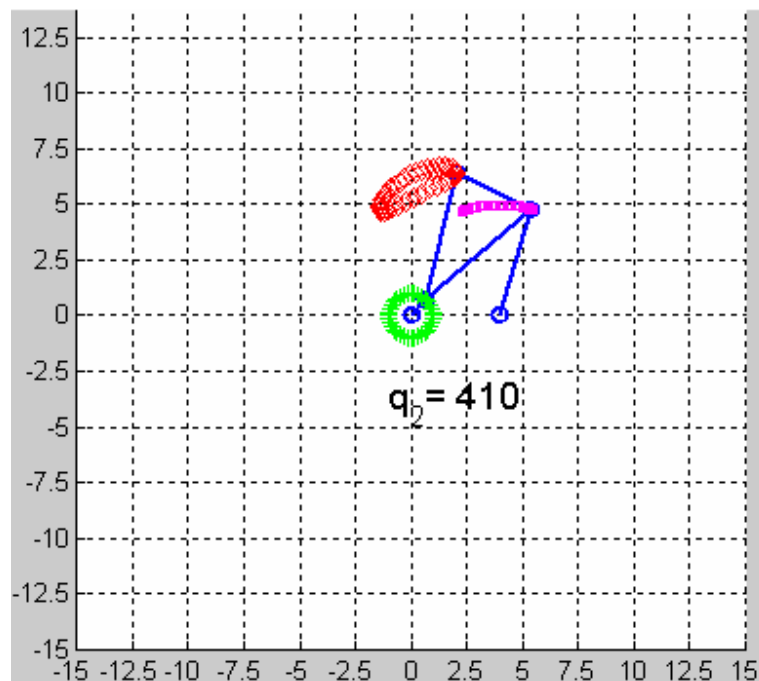


Figura 12: Curva (em vermelho) da saída do mecanismo

$L3 = 6,25$
 $L2 = 1$
 $L4 = 5$
 $L1 = 4$
 $P=5,88$
 $\Phi=36^\circ$

Nota-se que nestas curvas a “barriga” permanece ao contrário, isso faz com que possa ser necessária a inversão da curva na construção, além de ter uma amplitude pequena (que implica num passo menor). Mas vemos que a velocidade possui menor variação, o que contribui para um movimento mais suave.

Apesar de a terceira curva apresentar uma parte reta maior, as velocidades são extremamente diferentes. Já na segunda curva, temos velocidades mais próximas, o que é fundamental para um passo sem trancos, mas pode forçar o desequilíbrio do protótipo enquanto um dos pés estiver no ar. A última montagem supõe o auxílio posterior de um paralelogramo, que pode ser usado para amplificar o movimento, além de invertê-lo. Na primeira, a relação das medidas desfavorece a construção (se usarmos um *crank* de 4 cm, por exemplo, teríamos barras de mais de 40cm).

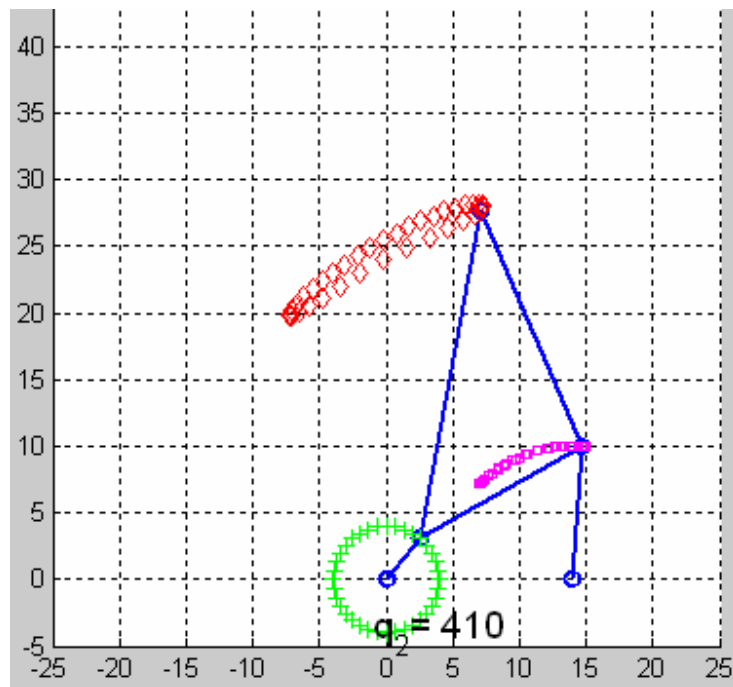


Figura 13: Curva (em vermelho) da saída do mecanismo

$L3 = 14$
 $L2 = 4$
 $L4 = 10$
 $L1 = 14$
 $P = 25$
 $\Phi = 50^\circ$

Dos exemplos analisados, já pensando na construção do protótipo, foi escolhida esta última configuração, pois ela é a que melhor atende aos requisitos de posição, construção e estabilidade, apresenta um formato de curva adequado pelo desnível entre as posições do passo, além de uma velocidade levemente maior enquanto o pé está no ar, reduzindo a possibilidade de tombamento lateral. Por fim, ela atende aos requisitos dinâmicos calculados, cujos valores seguem abaixo.

Em anexo ao relatório seguem os códigos para a simulação. O primeiro com a simulação cinemática e o segundo com simulação cinemática e dinâmica. Os valores são os da curva escolhida.

Parte da simplificação do movimento de andar, nos fez adotar um mecanismo estável estaticamente, para que não houvesse a necessidade de equilibrá-lo durante o

movimento. Para isso construímos os pés de maneira que o centro de massa ficasse sobre a área de cada pé, enquanto este permanecesse no chão.

Construção

Devido a suas propriedades mecânicas, além da versatilidade ao trabalhar com o material, será utilizado o alumínio para a construção do mecanismo. Os desenhos de construção, bem como da montagem do mecanismo estão em anexo, em arquivos do formato Solid Works.

Com relação ao motor, será utilizado um motor de passo para cada perna, controlados por computador através de porta paralela e CI ULN2803, com montagem conforme a figura 14. Este circuito permitirá, o controle independente de cada perna, de acordo com a programação. A linguagem de programação escolhida para isso é Visual Basic, o programa foi adaptado para controlar dois motores ao mesmo tempo, em vez de apenas um.

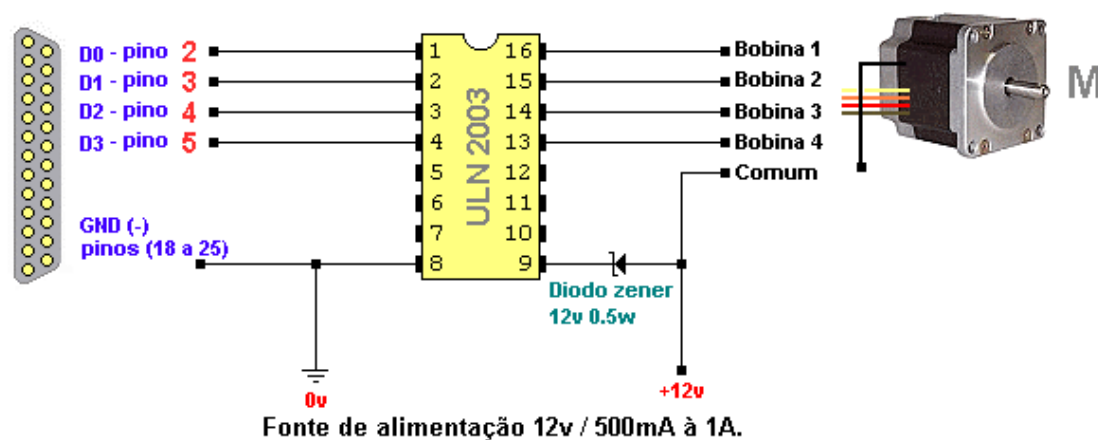


Figura 14: Pinagem para controle de um motor utilizando o ULN2803

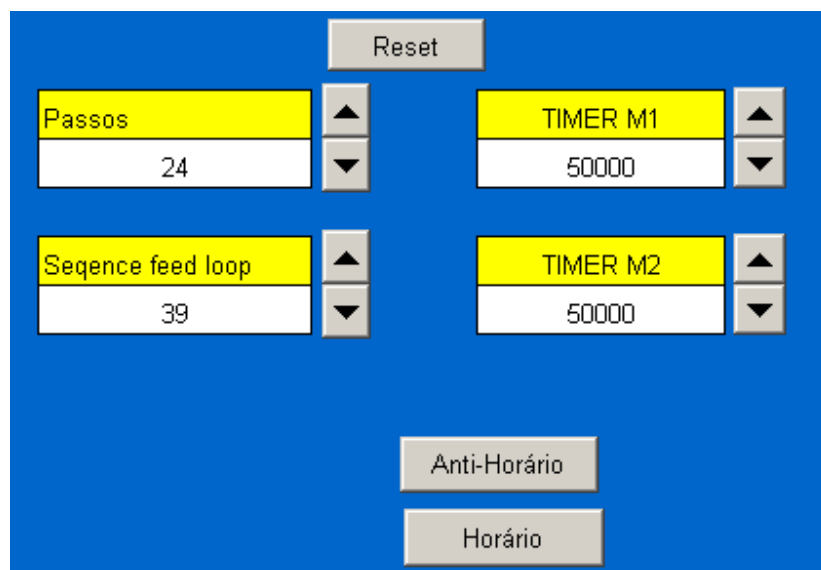


Figura 15: Tela de controle dos motores

Será utilizado passo completo com energização de duas bobinas por passo, que fornece um torque maior que o passo completo de uma bobina e tem consumo de energia menor que o meio passo. Essas características são importantes, pois não necessitamos de uma velocidade muito grande, além de termos restrição no consumo da energia dos motores pelo circuito eletrônico ULN2803 (no qual a corrente deve ser limitada a 500mA).

Nº do passo	B3	B2	B1	B0	Motor 1 (saída da porta)	Motor 2 (saída da porta)
1	1	1	0	0	12	192
2	0	1	1	0	6	96
3	0	0	1	1	3	48
4	1	0	0	1	9	144

Figura 16: Seqüência de bits/passos para cada motor

Resultados

Apesar de o controle dos motores ter funcionado perfeitamente e o mecanismo, quando movido manualmente descrever o movimento esperado, os motores não conseguiram fornecer torque suficiente para que obtivéssemos um movimento completo de acordo com o esperado.

Conclusão

Certas considerações, como não levar em conta o atrito nas juntas e desconsiderar a deformação mecânica, além de um motor fraco, cujo holding torque não suportava a carga da estrutura foram as principais razões de o mecanismo não ter funcionado conforme o esperado.

Anexos

Anexo 1: Código-fonte do programa para simulação cinemática

```
%
clear
close
figure(3);
V=2*2.5;%coupler
W=1;%crank
U=1.75*2.5;%rocker
O2O4=7;%ground
P=5.7*2.5*2*sin(36*pi/180);%triang
Phi=50*pi/180; %angulo do ponto P do triangulo com o lado base.
q2_0=pi/3; %q2 inicial= 60graus
q2=q2_0;
I=36; %I=numero de iteracoes desejadas
for k=1:I %variacoes
    q2=(k-1)*(2*pi/I)+q2_0; %calculo do angulo theta2 da sequencia de
frames
    r=sqrt(O2O4^2 + W^2 -2*O2O4*W*cos(q2));
    Beta=acos((O2O4^2+r^2-W^2)/(2*r*O2O4));
    Psi=acos((V^2+r^2-U^2)/(2*r*V));
    Lambda=acos((U^2+r^2-V^2)/(2*r*U));
    if sin(q2)>=0%(q2>=0) & (q2<=pi)
        T4=pi-Lambda-Beta;
        T3=Psi-Beta;
    else
        T4=pi-Lambda+Beta;
        T3=Psi+Beta;
    end
    hold on
    axis([-10 10 -5 25]) %limites dos eixos x e y
    axis square %transforma a area de plotagem em quadrado
    set(gca, 'XTick', -10:2.5:10)
    set(gca, 'YTick', -5:2.5:25)
    grid on %ativa as grades
    Lx=[0,W*cos(q2),O2O4+U*cos(T4),O2O4];
    Ly=[0,W*sin(q2),U*sin(T4),0];
    plot(Lx,Ly, '-ob', 'LineWidth',2);
    hold on
    Tx=[W*cos(q2),W*cos(q2)+P*cos(T3+Phi),O2O4+U*cos(T4),W*cos(q2)];
    Ty=[W*sin(q2),W*sin(q2)+P*sin(T3+Phi),U*sin(T4),W*sin(q2)];
    plot(Tx,Ty, '-ob', 'LineWidth',2);
    Ax(k)=W*cos(q2);
    Ay(k)=W*sin(q2);
    plot(Ax,Ay, '+g');
    Bx(k)=O2O4+U*cos(T4);
    By(k)=U*sin(T4);
    plot(Bx,By, 'sm', 'MarkerSize',4);
    Px(k)=W*cos(q2)+P*cos(T3+Phi);
    Py(k)=W*sin(q2)+P*sin(T3+Phi);
    plot(Px,Py, 'dr');
    xlabel('Eixo X');
    ylabel('Eixo Y');
```

```

q2g=q2*180/pi; %converte o angulo T2 em graus
q2gs=num2str(q2g); %converte o numero T2g em string
t3=text(-1,-4,['q_2= ',q2gs],...'BackgroundColor','w',
        ...'EdgeColor','b',...
        'FontSize',14);
M(k+1) = getframe(gcf,[0 0 550 400]); %comando getframe para
transformar a
%figura atual num frame (quadro) para animacao posterior.
clf
end
M(1)=M(I+1);
movie(M,1,3);
%plot(0,0)
movie2avi(M,'palt.avi','quality',100,'fps',3) %o comando movie2avi gera
um video
%formato avi com os frames M gerados.

```

Anexo 2: Código-fonte do programa para simulação dinâmica

```

%
clear
close
figure(3);
V=5;%coupler
W=1;%crank
U=4.375;%rocker
O204=7;%ground
P=28.5*sin(36*pi/180);%triang
Phi=50*pi/180; %angulo do ponto P do triangulo com o lado base.
q2_0=pi/3; %q2 inicial= 60graus
q2=q2_0;
I=36; %I=numero de iteracoes desejadas

L1=O204;%ground
L2=W;%crank
L3=V;%coupler
L4=U;%rocker
BP=P/sin(Phi);%lado esq tri
AP=P/cos(Phi);%lado dir tri

W2=2; %velocidade angular 2 em rad/s
ALFA2=0; %aceleracao angular 0 em rad/s2
F=0; %Forca aplicada no ponto P (negativo para baixo)
FX=real(F);
FY=imag(F);
g=10; %aceleracao da gravidade

%Calculo da massa das barras
RH0=750; %densidade do mdf =750kg/m3
WW=25e-3; %lado da barra
T=5e-3; %espessura da barra

AT= WW*T; %area da seccao transversal das barras
V2=L2*AT;
V3=AP*AT;
V4=L4*AT;

```

```

M2=RHO*V2;
M3=RHO*V3;
M4=RHO*V4;
%Calculo dos momentos de inercia de massa
%Formula: Iz=M*(W*W+L*L)/12
IZG2=M2*(W*W+L2*L2)/12;
IZG3=M3*(W*W+AP*AP)/12;
IZG4=M4*(W*W+L4*L4)/12;
%

%
for k=1:I %variacoess
    q2=(k-1)*(2*pi/I)+q2_0; %calculo do angulo theta2 da sequencia de
frames
    r=sqrt(0204^2 + W^2 -2*0204*W*cos(q2));
    Beta=acos((0204^2+r^2-W^2)/(2*r*0204));
    Psi=acos((V^2+r^2-U^2)/(2*r*V));
    Lambda=acos((U^2+r^2-V^2)/(2*r*U));
    if sin(q2)>=0%(q2>=0) & (q2<=pi)
        T4=pi-Lambda-Beta;
        T3=Psi-Beta;
    else
        T4=pi-Lambda+Beta;
        T3=Psi+Beta;
    end
    hold on
    axis([-10 10 -5 25]) %limites dos eixos x e y
    axis square %transforma a area de plotagem em quadrado
    set(gca, 'XTick', -10:2.5:10)
    set(gca, 'YTick', -5:2.5:25)
    grid on %ativa as grades
    Lx=[0,W*cos(q2),0204+U*cos(T4),0204];
    Ly=[0,W*sin(q2),U*sin(T4),0];
    plot(Lx,Ly, '-ob', 'LineWidth', 2);
    hold on
    Tx=[W*cos(q2),W*cos(q2)+P*cos(T3+Phi),0204+U*cos(T4),W*cos(q2)];
    Ty=[W*sin(q2),W*sin(q2)+P*sin(T3+Phi),U*sin(T4),W*sin(q2)];
    plot(Tx,Ty, '-ob', 'LineWidth', 2);
    Ax(k)=W*cos(q2);
    Ay(k)=W*sin(q2);
    plot(Ax,Ay, '+g');
    Bx(k)=0204+U*cos(T4);
    By(k)=U*sin(T4);
    plot(Bx,By, 'sm', 'MarkerSize', 4);
    Px(k)=W*cos(q2)+P*cos(T3+Phi);
    Py(k)=W*sin(q2)+P*sin(T3+Phi);
    plot(Px,Py, 'dr');
    xlabel('Eixo X');
    ylabel('Eixo Y');
    q2g=q2*180/pi; %converte o angulo T2 em graus
    q2gs=num2str(q2g); %converte o numero T2g em string
    t3=text(-1,-4,['q_2= ',q2gs],... 'BackgroundColor','w',
        ... 'EdgeColor','b',...

```

```

        'FontSize',14);
    MM(k+1) = getframe(gcf,[0 0 550 400]); %comando getframe para
transformar a
    %figura atual num frame (quadro) para animacao posterior.
    clf

%Problema de Posicao
Q2=q2; %angulo Q2
E=sqrt(L2*L2 + L1*L1 - 2*L2*L1*cos(Q2));
BETA=acos((E*E + L1*L1 - L2*L2)/(2*E*L1));
PSI=acos((E*E + L3*L3 - L4*L4)/(2*E*L3));
LAMBDA=acos((E*E + L4*L4 - L3*L3)/(2*E*L4));
TETA4=pi-LAMBDA-BETA;
TETA3=PSI-BETA;
GAMA=pi-PSI-LAMBDA;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problema de velocidade
%Equacao vetorial: VA + VBA = VB
% L2*j*W2*exp(j*Q2) + L3*j*W3*exp(j*TETA3) = L4*j*W4*exp(j*TETA4)
% REAL: -L3*sin(TETA3)*W3 + L4*sin(TETA4)*W4 = L2*W2*sin(Q2)
% IMAG: L3*cos(TETA3)*W3 - L4*cos(TETA4)*W4 = -L2*W2*cos(Q2)
A=[-L3*sin(TETA3) , L4*sin(TETA4) ; L3*cos(TETA3) , -L4*cos(TETA4)];
C=[L2*W2*sin(Q2) ; -L2*W2*cos(Q2)];
B=A\C;
W3=B(1); %ou B(1,1)
W4=B(2); %ou B(2,1)
VA=L2*j*W2*exp(j*Q2);
VAX=real(VA);
VAY=imag(VA);
VB=L4*j*W4*exp(j*TETA4);
VBX=real(VB);
VBY=imag(VB);
VP=VA + AP*j*W3*exp(j*TETA3);
VPX=real(VP);
VPY=imag(VP);
VG2=VA/2;
VG4=VB/2;
VG3=VA + (AP/2)*j*W3*exp(j*TETA3);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problema Aceleracao
%Equacao vetorial: AAT + AAN + ABAT + ABAN = ABT + ABN
%L2*j*ALFA2*exp(j*Q2) - L2*(W2^2)*exp(j*Q2) + L3*j*ALFA3*exp(j*TETA3)
- L3*(W3^2)*exp(j*TETA3)=
% = L4*j*ALFA4*exp(j*TETA4) - L4*(W4^2)*exp(j*TETA4)
%
%Real: -L3*sin(TETA3)*ALFA3 + L4*sin(TETA4)*ALFA4 =
% = -L4*(W4^2)*cos(TETA4) + L2*ALFA2*sin(Q2) + L2*(W2^2)*cos(Q2) +
L3*(W3^2)*cos(TETA3)
%
%Imag: L3*cos(TETA3)*ALFA3 - L4*cos(TETA4)*ALFA4 =
% = -L4*(W4^2)*sin(TETA4) - L2*ALFA2*cos(Q2) + L2*(W2^2)*sin(Q2) +
L3*(W3^2)*sin(TETA3)
R=[-L3*sin(TETA3) , L4*sin(TETA4) ; L3*cos(TETA3) , -L4*cos(TETA4)];
T=[-L4*(W4^2)*cos(TETA4) + L2*ALFA2*sin(Q2) + L2*(W2^2)*cos(Q2) +
L3*(W3^2)*cos(TETA3)];

```

```

T=[T; -L4*(W4^2)*sin(TETA4) - L2*ALFA2*cos(Q2) + L2*(W2^2)*sin(Q2) +
L3*(W3^2)*sin(TETA3)];
S=R\T;
ALFA3=S(1);
ALFA4=S(2);
AA=L2*j*ALFA2*exp(j*Q2) - L2*(W2^2)*exp(j*Q2); %aceleracao ponto A
AB=L4*j*ALFA4*exp(j*TETA4) - L4*(W4^2)*exp(j*TETA4); %aceleracao
ponto B
AG2=AA/2; %aceleracao CG2
AG2X=real(AG2);
AG2Y=imag(AG2);
AG4=AB/2; %aceleracao CG4
AG4X=real(AG4);
AG4Y=imag(AG4);
AG3A= (AP/2)*j*ALFA3*exp(j*TETA3) - (AP/2)*(W3^2)*exp(j*TETA3);
%diferenca aceleracao G3A
AG3= AA + AG3A; %aceleracao CG3
AG3X=real(AG3);
AG3Y=imag(AG3);
%
%%%%%%%%%%%%%%
%Analise dinamica
%Elo 2:
%vetorial: F32 + F12 + P2 = M2*AG2
%F32X + F12X = M2*AG2X;
%F32Y + F12Y = M2*AG2Y + M2*g;
%vetorial: T12 + RAG2xF32 + R02G2xF12 = IZG2*ALFA2
%T12 + RAG2X*F32Y - RAG2Y*F32X + R02G2X*F12Y - R02G2Y*F12X =
IZG2*ALFA2
RAG2=(L2/2)*exp(j*Q2);
RAG2X=real(RAG2);
RAG2Y=imag(RAG2);
R02G2=(L2/2)*exp(j*(Q2+pi));
R02G2X=real(R02G2);
R02G2Y=imag(R02G2);
%
%Elo 3:
%vetorial: -F32 + F43 + P3 + F = M3*AG3
% -F32X + F43X = M3*AG3X - FX;
% -F32Y + F43Y = M3*AG3Y + M3*g - FY;
%vetorial: -RAG3xF32 + RBG3xF43 + RPG3xF = IZG3*ALFA3
% -RAG3X*F32Y + RAG3Y*F32X + RBG3X*F43Y - RBG3Y*F43X = IZG3*ALFA3 -
RPG3X*FY + RPG3Y*FX;
RAG3=(AP/2)*exp(j*(TETA3+pi));
RAG3X=real(RAG3);
RAG3Y=imag(RAG3);
RBG3=(AP/2 - L3)*exp(j*(TETA3+pi));
RBG3X=real(RBG3);
RBG3Y=imag(RBG3);
RPG3=(AP/2)*exp(j*TETA3);
RPG3X=real(RPG3);
RPG3Y=imag(RPG3);
%
%Elo 4:
%vetorial: -F43 + F14 + P4 = M4*AG4
% -F43X + F14X = M4*AG4X;
% -F43Y + F14Y = M4*AG4Y + M4*g;

```

```

%vetorial: -RBG4xF43 + R04G4xF14 = IZG4*ALFA4;
% -RBG4X*F43Y + RBG4Y*F43X + R04G4X*F14Y - R04G4Y*F14X = IZG4*ALFA4;
RBG4=(L4/2)*exp(j*TETA4);
RBG4X=real(RBG4);
RBG4Y=imag(RBG4);
R04G4=(L4/2)*exp(j*(TETA4+pi));
R04G4X=real(R04G4);
R04G4Y=imag(R04G4);
%
%L=[F32X F32Y F12X F12Y T12 F43X F43Y F14X F14Y]'

K(1,:)=[1,0,1,0,0,0,0,0,0];
M(1,1)=[M2*AG2X];
K(2,:)=[0,1,0,1,0,0,0,0,0];
M(2,1)=[M2*AG2Y + M2*g];
K(3,:)=[-RAG2Y,RAG2X,-R02G2Y,R02G2X,1,0,0,0,0];
M(3,1)=[IZG2*ALFA2];
K(4,:)=[-1,0,0,0,0,0,1,0,0];
M(4,1)=[M3*AG3X - FX];
K(5,:)=[0,-1,0,0,0,0,0,1,0];
M(5,1)=[M3*AG3Y + M3*g - FY];
K(6,:)=[RAG3Y,-RAG3X,0,0,0,-RBG3Y,RBG3X,0,0];
M(6,1)=[IZG3*ALFA3 - RPG3X*FY + RPG3Y*FX];
K(7,:)=[0,0,0,0,0,-1,0,1,0];
M(7,1)=[M4*AG4X];
K(8,:)=[0,0,0,0,0,0,-1,0,1];
M(8,1)=[M4*AG4Y + M4*g];
K(9,:)=[0,0,0,0,0,0,RBG4Y,-RBG4X,-R04G4Y,R04G4X];
M(9,1)=[IZG4*ALFA4];
%
L=K\M;
L

end
%fim iteracao
MM(1)=MM(I+1);
movie(MM,1,3);
%plot(0,0)
movie2avi(MM,'palt.avi','quality',100,'fps',3) %o comando movie2avi gera
um video
%formato avi com os frames M gerados.

```

Anexo 3: Código-fonte do programa em Visual Basic para o controle dos motores (este código é uma adaptação do código de Bhaskar Gupta¹¹)

```
Private Sub CommandButton1_Click()
countit = 0
myTimer1 = 0
myTimer2 = 0
countit = Sheet1.Cells(4, 1)
countit2 = Sheet1.Cells(4, 7)
'MsgBox "Before Starting...Please be Patience as it may take few minutes"
Do While countit > 0
    myTimer1 = Sheet1.Cells(4, 3)
    myTimer2 = Sheet1.Cells(7, 3)

    'Primeiro passo
    Out 888, 3
    Do While myTimer1 > 0
        myTimer1 = myTimer1 - 1
    Loop

    Out 888, 48
    Do While myTimer2 > 0
        myTimer2 = myTimer2 - 1
    Loop

    myTimer1 = Sheet1.Cells(4, 3)
    myTimer2 = Sheet1.Cells(7, 3)

    'Segundo passo
    Out 888, 6
    Do While myTimer1 > 0
```

```
myTimer1 = myTimer1 - 1
```

```
Loop
```

Out 888, 96

```
Do While myTimer2 > 0
```

```
myTimer2 = myTimer2 - 1
```

```
Loop
```

```
myTimer1 = Sheet1.Cells(4, 3)
```

```
myTimer2 = Sheet1.Cells(7, 3)
```

'Terceiro passo

Out 888, 12

```
myTimer = Sheet1.Cells(4, 3)
```

```
Do While myTimer1 > 0
```

```
myTimer1 = myTimer1 - 1
```

```
Loop
```

Out 888, 192

```
myTimer = Sheet1.Cells(4, 7)
```

```
Do While myTimer2 > 0
```

```
myTimer2 = myTimer2 - 1
```

```
Loop
```

```
myTimer1 = Sheet1.Cells(4, 3)
```

```
myTimer2 = Sheet1.Cells(7, 3)
```

'Quarto passo

Out 888, 9

```
myTimer = Sheet1.Cells(4, 3)
```

```
Do While myTimer1 > 0
```

```
    myTimer1 = myTimer1 - 1
    Loop
Out 888, 144
    myTimer = Sheet1.Cells(4, 7)
    Do While myTimer2 > 0
        myTimer2 = myTimer2 - 1
    Loop
```

```
countit = countit - 1
```

```
Loop
```

```
Out 888, 0
```

```
'MsgBox "Done"
```

```
End Sub
```

```
'Private Sub SpinButton1_SpinDown()
```

```
'myTimer = Sheet1.Cells(4, 3)
```

```
,
```

```
'If myTimer = 0 Then
```

```
'myTimer = 0
```

```
'Else
```

```
'myTimer = myTimer - 1
```

```
'End If
```

```
,
```

```
'Sheet1.Cells(4, 3) = myTimer
```

```
,
```

```
'End Sub
```

```
'Private Sub SpinButton1_SpinUp()
```

```
'myTimer = Sheet1.Cells(4, 3)
```

```
'myTimer = myTimer + 1
'Sheet1.Cells(4, 3) = myTimer
'End Sub
```

```
"
```

```
'Private Sub SpinButton2_SpinDown()
'countit = Sheet1.Cells(4, 1)
'
'If countit = 0 Then
'countit = 0
'Else
'countit = countit - 1
'End If
'
'Sheet1.Cells(4, 1) = countit
'End Sub
"
```

```
'Private Sub SpinButton2_SpinUp()
'countit = Sheet1.Cells(4, 1)
'countit = countit + 1
'Sheet1.Cells(4, 1) = countit
'End Sub
Private Sub CommandButton2_Click()
Out 888, 0
End Sub
```

```
Private Sub CommandButton3_Click()
countit = 0
myTimer = 0
countit = Sheet1.Cells(4, 1)
```

'MsgBox "Before Starting...Please be Patience as it may take few minutes"

Do While countit > 0

myTimer = Sheet1.Cells(4, 3)

Out 888, 1

Do While myTimer > 0

myTimer = myTimer - 1

Loop

myTimer = Sheet1.Cells(4, 3)

Out 888, 2

Do While myTimer > 0

myTimer = myTimer - 1

Loop

Out 888, 4

myTimer = Sheet1.Cells(4, 3)

Do While myTimer > 0

myTimer = myTimer - 1

Loop

Out 888, 8

myTimer = Sheet1.Cells(4, 3)

Do While myTimer > 0

myTimer = myTimer - 1

Loop

countit = countit - 1

Loop

Out 888, 0

End Sub

Private Sub CommandButton4_Click()

countit = 0

myTimer = 0

countit = Sheet1.Cells(4, 1)

'MsgBox "Before Starting...Please be Patience as it may take few minutes"

Do While countit > 0

myTimer = Sheet1.Cells(4, 3)

Out 888, 1

Do While myTimer > 0

myTimer = myTimer - 1

Loop

myTimer = Sheet1.Cells(4, 3)

Out 888, 3

Do While myTimer > 0

myTimer = myTimer - 1

Loop

Out 888, 2

myTimer = Sheet1.Cells(4, 3)

Do While myTimer > 0

myTimer = myTimer - 1

Loop

Out 888, 6

myTimer = Sheet1.Cells(4, 3)

Do While myTimer > 0

myTimer = myTimer - 1

Loop

myTimer = Sheet1.Cells(4, 3)

Out 888, 4

Do While myTimer > 0

myTimer = myTimer - 1

Loop

```
myTimer = Sheet1.Cells(4, 3)
```

```
Out 888, 12
```

```
Do While myTimer > 0
```

```
myTimer = myTimer - 1
```

```
Loop
```

```
Out 888, 8
```

```
myTimer = Sheet1.Cells(4, 3)
```

```
Do While myTimer > 0
```

```
myTimer = myTimer - 1
```

```
Loop
```

```
Out 888, 9
```

```
myTimer = Sheet1.Cells(4, 3)
```

```
Do While myTimer > 0
```

```
myTimer = myTimer - 1
```

```
Loop
```

```
countit = countit - 1
```

```
Loop
```

```
Out 888, 0
```

```
End Sub
```

```
Private Sub CommandButton5_Click()
```

```
countit = 0
```

```
myTimer = 0
```

```
countit = Sheet1.Cells(4, 1)
```

```
'MsgBox "Before Starting...Please be Patience as it may take few minutes"
```

```
Do While countit > 0
```

```
myTimer = Sheet1.Cells(4, 3)
```

```
Out 888, 8
```

```
Do While myTimer > 0
```

```

    myTimer = myTimer - 1
    Loop
    myTimer = Sheet1.Cells(4, 3)
Out 888, 4
    Do While myTimer > 0
    myTimer = myTimer - 1
    Loop
Out 888, 2
    myTimer = Sheet1.Cells(4, 3)
    Do While myTimer > 0
    myTimer = myTimer - 1
    Loop
Out 888, 1
    myTimer = Sheet1.Cells(4, 3)
    Do While myTimer > 0
    myTimer = myTimer - 1
    Loop
countit = countit - 1
Loop
Out 888, 0
End Sub

```

```

Private Sub CommandButton6_Click()
countit = 0
myTimer = 0
countit = Sheet1.Cells(4, 1)
'MsgBox "Before Starting...Please be Patience as it may take few minutes"
Do While countit > 0
    myTimer1 = Sheet1.Cells(4, 3)
    myTimer2 = Sheet1.Cells(7, 3)

```

'Primeiro passo

Out 888, 9

```
Do While myTimer1 > 0  
myTimer1 = myTimer1 - 1  
Loop
```

Out 888, 144

```
Do While myTimer2 > 0  
myTimer2 = myTimer2 - 1  
Loop
```

```
myTimer1 = Sheet1.Cells(4, 3)  
myTimer2 = Sheet1.Cells(7, 3)
```

'Segundo passo

Out 888, 12

```
Do While myTimer1 > 0  
myTimer1 = myTimer1 - 1  
Loop
```

Out 888, 192

```
Do While myTimer2 > 0  
myTimer2 = myTimer2 - 1  
Loop
```

```
myTimer1 = Sheet1.Cells(4, 3)  
myTimer2 = Sheet1.Cells(7, 3)
```

'Terceiro passo

Out 888, 6

```
myTimer = Sheet1.Cells(4, 3)
```

```
Do While myTimer1 > 0
```

```
myTimer1 = myTimer1 - 1
```

```
Loop
```

Out 888, 96

```
myTimer = Sheet1.Cells(4, 7)
```

```
Do While myTimer2 > 0
```

```
myTimer2 = myTimer2 - 1
```

```
Loop
```

```
myTimer1 = Sheet1.Cells(4, 3)
```

```
myTimer2 = Sheet1.Cells(7, 3)
```

'Quarto passo

Out 888, 3

```
myTimer = Sheet1.Cells(4, 3)
```

```
Do While myTimer1 > 0
```

```
myTimer1 = myTimer1 - 1
```

```
Loop
```

Out 888, 48

```
myTimer = Sheet1.Cells(4, 7)
```

```
Do While myTimer2 > 0
```

```
myTimer2 = myTimer2 - 1
```

```
Loop
```

```
countit = countit - 1
```

```
Loop
```

Out 888, 0

```
'MsgBox "Done"
```

```
End Sub
```

```
Private Sub CommandButton7_Click()
```

```
countit = 0
```

```
myTimer = 0
```

```
countit = Sheet1.Cells(4, 1)
```

```
'MsgBox "Before Starting...Please be Patience as it may take few minutes"
```

```
Do While countit > 0
```

```
    myTimer = Sheet1.Cells(4, 3)
```

```
Out 888, 9
```

```
    Do While myTimer > 0
```

```
        myTimer = myTimer - 1
```

```
    Loop
```

```
    myTimer = Sheet1.Cells(4, 3)
```

```
Out 888, 8
```

```
    Do While myTimer > 0
```

```
        myTimer = myTimer - 1
```

```
    Loop
```

```
Out 888, 12
```

```
myTimer = Sheet1.Cells(4, 3)
```

```
    Do While myTimer > 0
```

```
        myTimer = myTimer - 1
```

```
    Loop
```

```
Out 888, 4
```

```
myTimer = Sheet1.Cells(4, 3)
```

```
    Do While myTimer > 0
```

```
        myTimer = myTimer - 1
```

```
    Loop
```

```
    myTimer = Sheet1.Cells(4, 3)
```

Out 888, 6

Do While myTimer > 0

myTimer = myTimer - 1

Loop

myTimer = Sheet1.Cells(4, 3)

Out 888, 2

Do While myTimer > 0

myTimer = myTimer - 1

Loop

Out 888, 3

myTimer = Sheet1.Cells(4, 3)

Do While myTimer > 0

myTimer = myTimer - 1

Loop

Out 888, 1

myTimer = Sheet1.Cells(4, 3)

Do While myTimer > 0

myTimer = myTimer - 1

Loop

countit = countit - 1

Loop

Out 888, 0

End Sub

Private Sub SpinButton1_Change()

End Sub

Private Sub SpinButton2_Change()

End Sub

Private Sub SpinButton3_Change()

End Sub

Private Sub SpinButton4_Change()

End Sub

Private Sub SpinButton5_Change()

End Sub

Private Sub SpinButton6_Change()

End Sub

Referências Bibliográficas

1) OTTAVIANO, Erika; LANNI, Chiara; CECCARELLI, Marco. *Numerical and Experimental Analysis of a Pantograph-Leg with a Fully-Rotative Actuating Mechanism*. University of Cassino.

2) TAVOLIERI, Cristina; OTTAVIANO, Erika; CECCARELLI, Marco; DI RIENZO, Alessandro. *Analisis and Design of a 1-DOF Leg for Walking Machines*. University of Cassino.

3) AWTAR, Shorya; SINGH, Saket Kumar; TEWARI, Praveen; *Stair Climbing Mechanism: Senior Year Project*. Indian Institute of Technology, Kanpur

4) <http://groups.csail.mit.edu/locomotion/pubs.html> (acessado em 29/05/2007 às 13:32);

5) <http://ruina.tam.cornell.edu/hplab/pdw.html> (acessado em 29/05/2007 às 13:44).

6) NORTON, Robert L.; *Design of Machinery: Na Introduction to the Synthesis and Analysis of Mechanisms and Machines*. McGraw-Hill, 2001.

7) <http://ww1.ucmss.com/books/LFS/CSREA2006/ESA7302.pdf>(acessado em 29/05/2007 às 13:55).

8) <http://msm.guc.edu.eg/Graduation%20Projects%20topics/Ayman%20El-Badawy%20BSc%20projects.pdf> (acessado em 29/05/2007 às 14:05)

9) <http://www.cns.atr.jp/~gordon/papers/clawar2001.pdf> (acessado em 29/05/2007 às 14:10)

10) <http://www.guiadomarceneiro.com/madeira/?gdm=mdfs> (acessado em 15/05/2008)

11) GUPTA, Bhaskar; *Understanding various types of Stepper Motors And Controlling it through Parallel Port* (acessado em <http://collections.sharewith.us> em 20/10/2008)