

Marcos Paulo Faria Lima Barreto, “Projeto, Construção e Simulação de um Manipulador Robótico Paralelogramo”

Sumário

1	Introdução.....	3
2	Fundamentos Teóricos.....	4
2.1	Partes de um manipulador.....	4
2.2	Classificação	4
2.3	Extremidade Efetiva (End Efector).....	5
2.4	Cinemática Direta e Inversa.....	6
2.5	Jacobiana.....	7
2.6	Estática.....	8
2.7	Rigidez	9
2.8	Dinâmica	9
2.9	Controles	11
2.9.1	PID.....	12
2.9.2	Controle de Torque Computado	12
2.9.3	Robusto.....	13
2.10	Sensores e Sistemas de Atuação	13
2.10.1	Atuação.....	13
2.10.2	Fonte de Energia.....	14
2.10.3	Tranformadores de Potência Mecânica	16
2.10.4	Elementos de Atuação	17
2.10.5	Motores DC de Ímã Permanente	18
2.10.6	Sensores	20
2.10.7	Monitoramento do Ambiente	24
3	Modelo experimental.....	26
3.1	Maximização da destreza.....	27
3.2	Definição do comprimento dos elos	28
3.3	Maximização da área de trabalho x Inércia	29
3.4	Modelo Computacional.....	31
3.5	Fonte de Energia	32

3.6	Controlador	33
3.7	Amplificador	33
3.8	Atuadores	34
3.9	Sensores	35
3.9.1	Encoder do Manipulador Protótipo	35
3.9.2	Tacômetros do Manipulador Protótipo.....	36
3.9.3	Medição de Torque.....	38
3.10	Montagem	38
4	Simulação	40
4.1	Considerações Gerais.....	40
4.2	Ciclos de simulação	41
4.3	Resultados	42
4.3.1	PD	42
4.3.2	PD+G.....	44
4.3.3	PID.....	45
4.3.4	CTC	47
5	Conclusão	49
6	Referências	50

1 Introdução

Ao longo das últimas três décadas a robótica tem invadido cada vez mais a vida das pessoas e principalmente o ambiente industrial. É notável que o número de fábricas e ambientes industriais vem se automatizando cada vez mais. Esta estatística pode ser notada no Brasil pelo crescimento de empresas de automação industrial e projetos de automação como a Chemtech, Promon, IESA que mesmo com a crise econômica mundial no segundo semestre de 2008 mantiveram seu crescimento inalterado.

A robótica uma parte da automação industrial que realiza grande trabalho no crescimento e aceleração da economia. Robôs têm a função de substituir a força de trabalho humana em operações de risco, em ambientes inatingíveis pelo homem como, por exemplo, o fundo do mar e a superfície de reatores nucleares e ainda em tarefas onde possuem maior precisão e velocidade. Podem também executar diversas tarefas e serem reprogramados facilmente. Por essas e muitas outras vantagens a quantidade de robôs operando em fábricas tem crescido.

Um fator que atrapalha o aumento da utilização de robôs na indústria é o elevado custo de instalação deste, ou seja, a quantidade de capital inicial necessário para automatizar uma fábrica com robôs é muito grande. Isso ocorre não só pela produção de um robô ser cara, mas pelo seu custo de projeto também o ser.

O projeto de um manipulador, por exemplo, envolve a caracterização de materiais, visualização de posições possíveis, área de trabalho, carga a ser levantada. Para baratear o projeto, a programação e a validação do comportamento de robôs em fábricas é muito utilizada a simulação computacional. Esta possibilita prever o comportamento do sistema criado para as programações a serem feitas nele. Desta forma pode-se evitar que programações mal feitas causem acidentes. Pode-se também verificar a capacidade deste em diversas situações sem que se perca tempo. No computador não há custos com materiais para testar novas configurações nem risco de acidentes. O único gasto que se tem é com a mão de obra dos especialistas que o programam. Logo realizar a simulação numérica é o barateamento do projeto em diversas frentes materiais, tempo e risco.

Neste trabalho foi escolhida uma configuração industrial de manipuladores para ser criado um modelo de projeto e estudo para a universidade. Este modelo escolhido é o paralelogramo. Este modelo é muito utilizado em indústrias montadoras de carros possui em geral grande capacidade de carga e se mostra bem eficiente quanto à área de trabalho.

Desta forma este trabalho destina-se a mostrar o projeto mecânico, construção do sistema completo, utilizar de ferramentas computacionais para facilitar e tornar mais ágil este processo de projeto e por fim programar um simulador que dê suporte a operação deste manipulador para diversos tipos de controles. Assim, pode-se compreender melhor o projeto deste tipo de ferramenta crescente no mercado e fornecer a futuros estudantes contato mais próximo com a mesma. Além de facilitar estudos de aprimoramento de controles de manipuladores.

2 Fundamentos Teóricos

Para o entendimento do projeto, controle e simulação computacional de manipuladores robóticos é necessário a definição e aprofundamento de alguns tópicos. São descritos abaixo as nomenclaturas, tipo, modelagem matemática do comportamento dinâmico e métodos de controle de um braço robótico.

2.1 Partes de um manipulador

Essencialmente neste trabalho considera-se que o manipulador é feito de partes rígidas conectadas por juntas. Assim é necessário descrever quais são e como é a representação das propriedades físicas relevantes destas duas partes que compõem a descrição mecânica do manipulador.

Citados anteriormente os elos são corpos rígidos e, portanto possuem uma geometria espacial nas três dimensões, massa, centro de massa e inércia rotativa. As juntas são as conexões entre os elos que permitem o deslocamento relativo entre estes. Assim são classificadas de acordo com o tipo de movimento que permitem. De forma geral podem ser rotativas que permitem rotação em torno de um eixo ou mais ou lineares que permitem deslocamento linear entre elos. A Figura 2-1 mostra um elo com duas juntas rotativas.

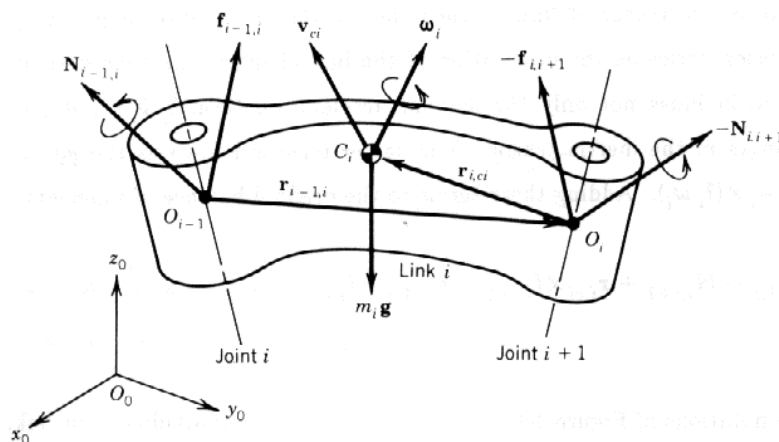


Figura 2-1

2.2 Classificação

A classificação dos manipuladores pode ser feita de diversas formas. A mais comum é feita a partir da forma como o conjunto de elos é ligada.

- Serial: Elos são ligados de forma seqüencial sem que seja fechado um laço entre eles topologicamente falando.
- Paralelo: As conexões dos elos formam laços entre si. Podem ser um ou mais.
- Híbridos: Tem partes ligadas em série e partes em paralelo.

A geometria da área de trabalho também pode ser uma forma de classificação. Novamente cito as formas mais comuns encontradas no mercado, mas outras combinações existem

- Cilíndricos
- Esféricos
- Paralelepípedos
- Plano

Nas Figuras a seguir Figura 2-2 Figura 2-3 é possível observar conjunto grande de configurações, tipos de elos e junta. A primeira contém apenas modelos do tipo série enquanto a segunda contém apenas paralelos.

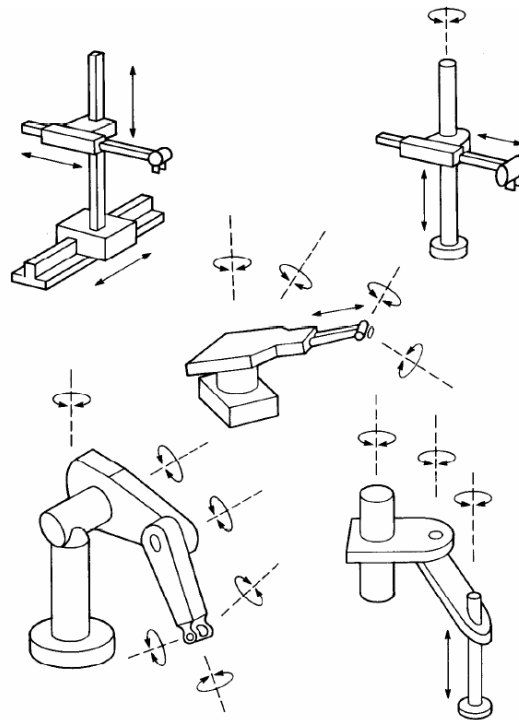


Figura 2-2

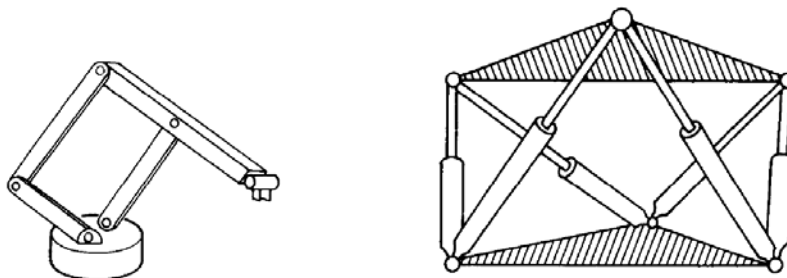


Figura 2-3

2.3 Extremidade Efetiva (End Effector)

A extremidade efetiva do manipulador nada mais é do que o ponto onde se pode atuar no meio de forma efetiva, ou seja, local onde se encontra a garra, o laser, ferro de

solda ou qualquer ferramenta que esteja sendo utilizada. Normalmente as trajetórias planejadas são pensadas para

2.4 Cinemática Direta e Inversa

A cinemática direta é equacionamento matemático que informa a posição da extremidade dados como referencia as posições dos motores que o controlam. Desta forma pode-se relacionar não só posição, mas também velocidade e aceleração entre estas partes. A posição do motor normalmente se confunde com a posição das juntas do manipulador podendo estar relacionadas também por uma redução. Assim, a Cinemática direta de um manipulador com k graus de liberdade pode ser resumida por:

$$\bar{X} = \bar{F}(q_1, q_2, \dots, q_k)$$

Onde \bar{X} é o vetor que contem as coordenadas cartesianas e os ângulos de direcionamento como, por exemplo, os ângulos de Euler extremidade efetiva:

$$\bar{X} = [x \ y \ z \ \phi \ \theta \ \psi]^T$$

A cinemática inversa faz a relação contrária, relaciona a a posição das juntas ou dos motores tendo como referencia a posição da extremidade. Esta relação é mais complicada de ser encontrada para manipuladores seriais do que para paralelos o inverso ocorre com a cinemática direta.

$$\bar{q} = F(\bar{X})$$

Exemplos:

Cinemática Direta:

$$\theta_1, \theta_2 \rightarrow x, y, \Phi$$

$$\begin{bmatrix} x \\ y \\ \Phi \end{bmatrix} = \begin{bmatrix} L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\ L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \\ \theta_1 + \theta_2 \end{bmatrix}$$

Cinemática Inversa:

$$x, y \rightarrow \theta_1, \theta_2$$

$$\tan(\alpha) = \frac{y}{x}$$

$$x^2 + y^2 = L_1^2 + L_2^2 + L_2 L_1 \cos(\theta_2) L_1 = L_1^2 + x^2 + y^2 - 2L_1 \sqrt{x^2 + y^2} \cos(\gamma)$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \alpha - \gamma = \tan^{-1}\left(\frac{y}{x}\right) \pm \cos^{-1}\left(\frac{L_1^2 + x^2 + y^2 - L_2^2}{2L_1 \sqrt{x^2 + y^2}}\right) \\ \pm \cos^{-1}\left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1 L_2}\right) \end{bmatrix}$$

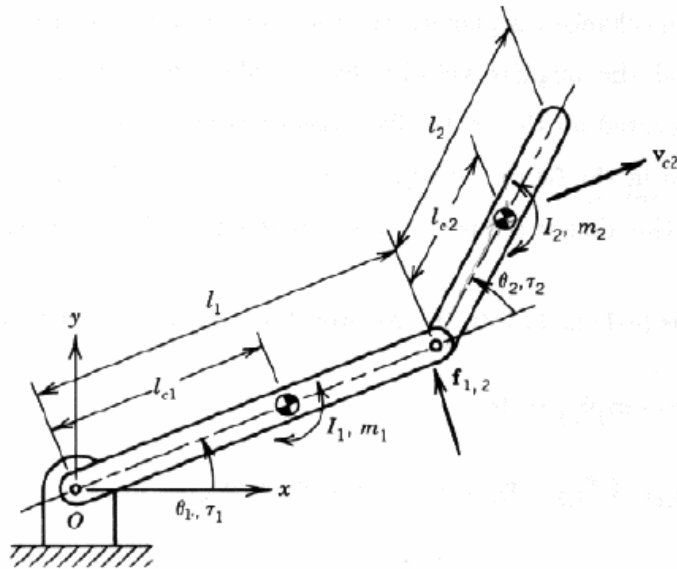


Figura 2-4

Em geral a cinemática direta é mais fácil de ser obtida para manipuladores do tipo serial. Existem algoritmos prontos pra isso e se torna muito simples quando usa a notação Denavit-Hartenberg. Já a cinemática inversa é mais complicada e exige tratamento particular para cada tipo de manipulador serial. Pode-se ver que o oposto ocorre com os manipuladores paralelos[14].

2.5 Jacobiana

A jacobiana é uma forma multidimensional de derivação. Se considerarmos que a posição é uma função apenas das posições, o que é verdade quando os elos do manipulador não são flexíveis e não há folgas na transmissão, e que as variáveis de posição são independentes entre si, para um manipulador com k graus de liberdade cuja cinemática direta é descrita, pode-se definir que [15]:

$$\delta \vec{X} = \frac{\partial \vec{F}}{\partial \vec{q}} \delta \vec{q} = J(\vec{q}) \delta \vec{q}$$

Por $F(\vec{q})$ ser independente para cada valor de \vec{X} , $J(\vec{q})$ pode ser escrita como uma matriz $k \times 6$ (três graus para a posição e o mesmo para orientação) podendo ser diminuída de acordo com a redução da quantidade de graus de liberdade do manipulador ou do problema em questão. Assim cada elemento da matriz jacobiana por ser escrito como:

$$J_{ij} = \frac{\partial F_j}{\partial q_i}$$

Assim, ao diferenciar os dois lados da equação com relação ao tempo pode-se obter a relação entre as velocidades das juntas e a velocidade da extremidade do manipulador.

$$\dot{\vec{X}} = J(\vec{q}) \dot{\vec{q}} = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$$

As três primeiras linhas da matriz relacionam o movimento das juntas ao movimento da posição linear da extremidade assim é chamada de matriz jacobiana

linear ou parte linear da jacobiana. Por analogia as três últimas são chamadas de matriz jacobiana angular ou parte angular da jacobiana.

Nota-se que a jacobiana muda para cada posição das juntas, sendo assim a relação é apenas pontualmente linear. Os autovalores desta matriz em qualquer ponto mostram a sensibilidade do deslocamento no ponto onde é calculada.

Exemplo:

$$J = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

Pensando-se no problema inverso ao discutido anteriormente, querendo através uma velocidade desejada à extremidade qual seria a velocidade nas juntas, pode-se utilizar a mesma equação para avaliá-lo.

$$\dot{q} = J^{-1}(q)\dot{X}$$

O que é importante revelar é que o núcleo da matriz jacobiana inversa mostra os pontos no espaço onde o manipulador perde graus de liberdade sabendo assim regiões limites de controle. Logo encontrar tais pontos é encontrar o limite de atuação. Esta posição pode ser encontrada procurando-se os pontos onde o determinante da matriz é nulo. Assim dá-se a estes o nome de pontos de singularidade, onde a jacobiana é singular.

2.6 Estática

A estática consiste em calcular qual o torque ou força necessária nos motores para que se possa manter o manipulador parado dadas forças e torques em algum ponto do manipulador. Assim, o manipulador deve estar em equilíbrio estático, o que significa que cada uma de seus elos deve estar na mesma situação. Mecanicamente o somatório de forças e momentos em cada um dos elos deve ser nulo. O que leva a equação:

$$\sum \vec{F}_{\text{externa}} + \sum \vec{F}_{\text{OutrosElos}} + \vec{g}M_{\text{Elo}} = 0$$

É válido lembrar que as componentes das forças perpendiculares ao vetor que do ponto de aplicação destas até o centro de gravidade também realizam momento com relação ao elo o que resulta na equação:

$$\sum \vec{M}_{\text{externa}} + \sum \vec{M}_{\text{OutrosElos}} + \sum \vec{F}_{\text{externa}} \times (\vec{r}_{\text{apolo}} - \vec{r}_{\text{cg}}) + \sum \vec{F}_{\text{OutrosElos}} = 0$$

Mas para objetivos de controle é apenas necessário saber qual deve ser a ação em cada atuador para que este sistema esteja em equilíbrio. Para tal pode-se utilizar o princípio dos trabalhos virtuais e provar que [14]:

$$\tau = J^T F$$

Esta relação não é válida apenas para forças, mas também para momentos já que a matriz jacobiana possui parte linear e angular. Onde:

$$F = [F_x \ F_y \ F_z \ M_x \ M_y \ M_z]^T$$

2.7 Rigidez

Pode-se relacionar também a força que um manipulador faz a um deslocamento na extremidade fazendo um controle proporcional de posição. Assim, dado um deslocamento o manipulador reage com uma força. Assim o braço se torna rígido a deslocamentos em sua extremidade como uma mola. Esta relação pode ser representada por [Asada cap4]:

$$F = K_e \Delta X$$

Como normalmente são medidos os deslocamentos nas juntas e não da extremidade e controlam-se os esforços dos atuadores e não as forças nessa deve-se traçar uma relação igualmente linear entre torque e deslocamentos das juntas representado por:

$$\begin{aligned} \tau &= K \Delta q = J^T F = J^T K_e \Delta X = J^T K_e J^{-1} \Delta q \\ \therefore K &= J^T K_e J^{-1} \end{aligned}$$

2.8 Dinâmica

A dinâmica de um corpo rígido começa basicamente pelas equações de Newton-Euler para descrição de movimentos.

Dinâmica Linear de Newton:

$$\sum F_{\text{externa}} + F_{i-1,i} - F_{i,i+1} + m_i g - m_i a_i = 0$$

Dinâmica Angular de Euler:

$$\sum \tau_{\text{ext}} + F_{\text{ext}} \times r_{\text{ext}} + \tau_{i-1,i} + \tau_{i,i+1} + F_{i-1,i} \times r_{i-1,ci} - F_{i,i+1} \times r_{i+1,ci} - I_i \dot{\omega} - \omega \times I \omega = 0$$

Estas equações descrevem o comportamento dinâmico em cada um dos elos do manipulador. Porém elas não são adequadas para se fazer uma análise geral do sistema já que não mostram relação direta dos esforços dos atuadores e o movimento produzido e são necessárias muitas equações de restrições e acoplamentos para chegar a esta. Para tal desenvolveu-se outra representação baseada em análise sistemática de entradas e saídas. Onde as entradas são os torques dos atuadores e as saídas a posição e orientação da extremidade.

A formulação lagrangeana pode fornecer esta noção global que se procura ao analisar um sistema dinâmico complexo. Além disso, já utiliza como coordenadas generalizadas as posições das juntas facilitando a formulação. Esta consiste na descrição do lagrangeano do sistema formado de duas parcelas, Energia Cinética e Potencial:

$$L = T - U$$

O comportamento dinâmico do sistema é descrito então como:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i$$

Onde Q_i é a força generalizada aplicada em q_i , T é a energia cinética do sistema e U é a energia potencial. Assim, a energia cinética ou potencial pode ser pensada como o somatório das energias correspondentes de cada um dos elos.

Para cada elo i pode-se escrever:

$$T_i = \frac{1}{2} m_i v_{ci}^T v_{ci} + \frac{1}{2} \omega_i^T I \omega_i$$

$$U = \sum_{i=0}^n m_i g^T r_{ci}^0$$

Para escrever a energia cinética em função das posições das juntas deve-se lembrar que a velocidade do centro de massa de um elo se relaciona com as velocidades das juntas através de uma matriz jacobiana como mostrado anteriormente. Logo, para cada elo i existirá $J^{ci} q = [v_{ci} \ \omega_i]^T$, tal que $J_E^{ci} q = v_{ci}$ e $J_A^{ci} q = \omega_i$. Assim, tem-se que:

$$T = \frac{1}{2} \sum_{i=1}^n (m_i q^T J_E^{ciT} J_E^{ci} q + q^T J_A^{ciT} I J_A^{ci} q) = \frac{1}{2} q^T H q$$

$$H = m_i J_E^{ciT} J_E^{ci} + J_A^{ciT} I J_A^{ci}$$

Escrevendo de forma escalar:

$$T = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n H_{ij} \dot{q}_i \dot{q}_j$$

Após estas considerações pode-se aplicá-las aos termos do lagrangeano e obter uma formulação mais abrangente e simplificada.

Para o primeiro termo do Lagrangeano $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i}$ não é considerada a parcela referente a energia potencial pois esta não depende diretamente da velocidade das juntas e portanto sua derivada com relação a velocidade destas é nula. Portanto o único termo restante é:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) = \sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \frac{dH_{ij}}{dt} \dot{q}_j = \sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n \frac{dH_{ij}}{dq_k} \dot{q}_j \dot{q}_k$$

Para o segundo termo são levados em consideração as duas parcelas, a energia cinética e a potencial. Calculando-as separadamente temos:

$$\frac{dT}{dq_i} = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{dH_{jk}}{dq_i} \dot{q}_j \dot{q}_k$$

$$\frac{dU}{dq_i} = \sum_{j=0}^n m_j g^T \frac{dr_{cj}^0}{dq_j} = \sum_{j=0}^n m_j g^T J_{L(cj)}^{ci} = G_i$$

Onde $J_{L(cj)}^{ci}$ é a coluna do jacobiano do centro de massa cj relativa ao deslocamento da junta q_i .

Para concluir o calculo dos termos em separados basta calcular o termo de força generalizada. Este termo resume todas as forças e momentos que realizariam trabalho no caso da movimentação infinitesimal do manipulador com a exceção da gravidade e

das forças de inércia do sistema. Considerando-se a separação entre o torque dos motores e as forças externas pode-se escrever este trabalho como sendo:

$$\delta \text{Trabalho} = \tau \delta q + \sum_{z=1}^m F_{extz} \delta p_{aplicação} = \left(\tau + \sum_{z=1}^m F_{extz} J_{p_{aplicação}} \right) \delta q = Q \delta q$$

As componentes do vetor $Q^T = [Q_1 \quad Q_2 \quad \dots \quad Q_n]$ são as forças generalizadas para as i equações do lagrangeano. Caso o ponto de aplicação de uma força seja a extremidade o Jacobiano do ponto de aplicação é o mesmo que o geral do manipulador.

Logo se pode reescrever o Lagrangeano como:

$$\sum_{j=1}^n H_{ij} \dot{q}_j + \sum_{j=1}^n \sum_{k=1}^n \frac{dH_{ij}}{dq_k} \dot{q}_j \dot{q}_k - \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{dH_{jk}}{dq_i} \dot{q}_j \dot{q}_k - \sum_{j=0}^n m_j g^T J_{L_{ij}}^c = Q_i$$

Resumindo:

$$h_{ijk} = \frac{dH_{ij}}{dq_k} - \frac{1}{2} \frac{dH_{jk}}{dq_i}$$

$$\sum_{j=1}^n H_{ij} \dot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k - G_i = Q_i$$

Logo a equação dinâmica do manipulador completo pode ser escrita como:

$$H\dot{q} + h - G = Q$$

Esta equação é a equação diferencial que rege o movimento do manipulador. É uma equação diferencial não linear com coeficientes não constantes no tempo e não homogênea. Sua solução não pode ser obtida de forma analítica e portanto deve ser simulada numericamente como será feito posteriormente neste trabalho.

2.9 Controles

Controle é uma reação ao erro entre o estado desejado do sistema e o estado encontrado. Esta pode levar em consideração apenas os distúrbios das variáveis de estado ou ser sensível a variações das características do sistema e ruídos dos sensores.

Existem controles que não são de reação, mas simplesmente de ação direta, mais conhecidos como controles de malha aberta. Estes são significativamente instáveis pois o sistema controlado deve ser extensamente conhecido para que saiba-se qual o comportamento obtido pelos comandos dados.

O controle de malha fecha utiliza-se de sensores para estimar qual o estado do sistema controlado e assim reagir as diferenças entre o estado desejado e o atual tentando diminuir esta diferença medida. Estes são os controles utilizados neste trabalho, alguns tipos programados no simulador são detalhados a seguir.

2.9.1 PID

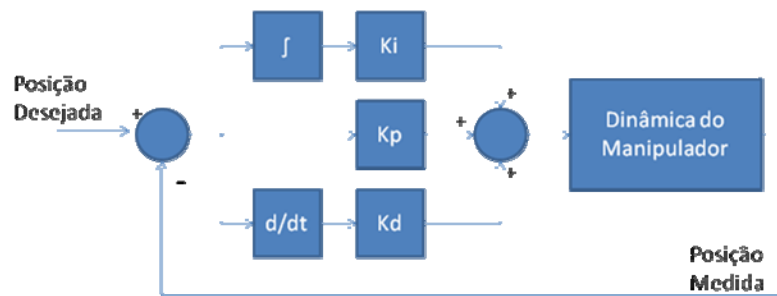


Figura 2-5

O controle PID é o controle que tenta diminuir o erro entre a variável medida e a variável desejada através de três fatores. Um proporcional a este erro, outro a derivada do respectivo erro e o ultimo referente à integral deste:

$$\tau_{PID} = K_p E + K_d \frac{dE}{dt} + K_i \int E dt$$

2.9.2 Controle de Torque Computado

Este controle calcula qual o torque necessário para anular todos os efeitos da dinâmica do manipulador: Inércia, Coriolis, efeito centrífugo e gravidade. Este torque é estimado pelas propriedades medidas e, portanto tenta transformar dinamicamente em massa unitárias pontuais. Pode ser falho caso os parâmetros estimados como massas, Inércias rotacionais e distancias de centros de massa e por conseqüência causar instabilidade. Sendo assim apenas necessário o acrescimo um controle PID. Na equação a baixo é usado o próprio controle PID usado sem o torque computado, para fazer a compensação é dividido pelo determinante da matriz de inércia.

$$H\ddot{q} + h - G = \tau$$

$$\tau = H(\ddot{q}_{desejada}) + h(q_{medida}, \dot{q}_{medida}) - G(q_{medida}) + \frac{\tau_{PID}}{\det(H)}$$

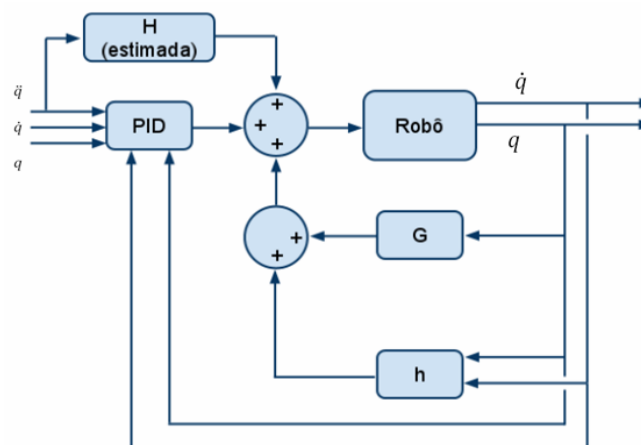


Figura 2-6

Pode-se falar apenas em compensação de gravidade, o que é uma particularização do torque computado, pois apenas leva em conta o termo gravitacional da realimentação da dinâmica.

2.9.3 Robusto

Este controle tem por objetivo eliminar instabilidades derivadas de estimativa erradas de propriedades mecânicas utilizadas em compensações como, por exemplo, compensação de gravidade ou torque computado. O termo por ser de compensação de erro só precisa ser utilizado quando o erro for maior que uma dada precisão do controlador.

Uma alternativa de controle robusto enunciada por [13] pode ser a de acrescentar um novo termo a equação de torque computado que se atualize a cada ciclo de controle. Este termo tem por objetivo diminuir o erro referente ao ciclo anterior de controle no ciclo atual levando em consideração este erro calculado para todas as variáveis de saída. O sistema referente deve obedecer ser diferencial de segunda ordem onde a matriz de inércia deve ser sempre positiva definida de coeficientes deriváveis o que é o caso da matriz de inércia do manipulador.

Assim, a lei de controle para uma entrada de controle $k+1$ é:

$$\tau_{k+1}(t) = \tau_k(t) + \Lambda e_k(t)$$

O erro $e_k(t)$ pode incluir não só a posição, mas também a velocidade multiplicada por outro termo de aprendizado. São sugerido termos de aprendizado diferentes entre a velocidade e a posição, mas que não é necessário. Com este acréscimo o controle fica:

$$\tau_{k+1}(t) = \tau_k(t) + \Lambda(\theta_d - \theta_k) + \psi(\dot{\theta}_d - \dot{\theta}_k)$$

Outros tipos de controle existem, muitos controles não lineares surgem para solucionar problemas particulares encontrados a cada nova aplicação. Alguns são: controle Fuzzy, controle preditivo por redes neurais, controle adaptativo.

2.10 Sensores e Sistemas de Atuação

Nesta secção serão descritos de forma geral dois componentes básicos de qualquer manipulador: os sensores e os sistemas de atuação. A atuação será descrita em quatro partes: em termos da fonte de energia necessária para mover o sistema; da amplificação responsável por converter os sinais de baixa potencia provenientes do controlador em sinais de potencia elevada a serem enviados aos atuadores; dos mecanismos de amplificação da potencia mecânica; dos elementos de atuação.

Os sensores são responsáveis por realizar a medição do estado do manipulador. Este estado pode ser definido por a imagem de sua posição as posições e velocidades das suas juntas, a força realizada em alguns pontos e outros. Portanto sensores destes tipos de variáveis de estado são apresentados a seguir.

2.10.1 Atuação

Dentre os tipos de sistemas de atuação podem ser destacados os elétricos e os hidráulicos, no entanto, por não fazerem parte do escopo deste texto e pelo fato do manipulador protótipo construído ser elétrico, os sistemas hidráulicos não serão abordados, informações para estes tipos podem ser obtidas em [1].

Segundo [2], o sistema de atuação das juntas de um manipulador pode ser constituído por:

- . Fonte de energia
- . Amplificador
- . Transmissões mecânicas
- . Elementos de atuação

A Figura 2-7 ilustra as conexões existentes entre os diversos componentes mencionados anteriormente.

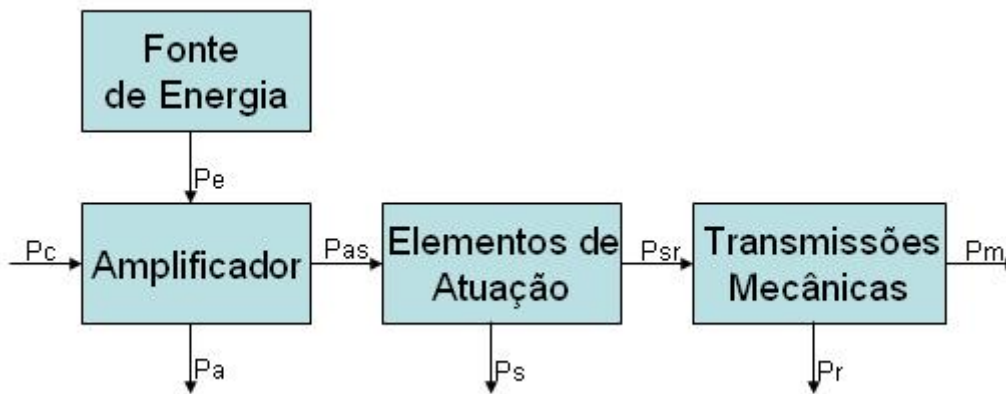


Figura 2-7

A Figura 2-7 representa um modelo genérico, fazendo referência apenas ao fluxo de potência existente entre os diversos constituintes do sistema. P_c está associado à potência do sinal de controle, P_e é a potência drenada da fonte de alimentação do sistema (elétrica ou hidráulica, por exemplo), P_{as} e P_{sr} são respectivamente a potência de entrada e de saída do elemento de atuação, observando que a segunda é de natureza mecânica, P_m é a potência efetiva enviada as juntas do manipulador e finalmente P_a , P_s e P_r representam as perdas por dissipação nos processos de conversão de energia respectivamente no amplificador, elemento de atuação e transmissões mecânicas.

2.10.2 Fonte de Energia

A fonte de energia tem como função prover a potência necessária ao amplificador, potência esta que deve ser de alguma forma proporcional ao sinal de controle enviado pelo controlador do sistema. A natureza da fonte de energia dependerá do tipo de elemento de atuação que está sendo utilizado, podendo ser de diferentes tipos como, por exemplo, elétrica ou hidráulica.

No caso de atuadores elétricos faz-se necessária uma fonte de alimentação elétrica constituída basicamente de um transformador e de uma ponte retificadora, capaz de converter a tensão alternada do sistema de distribuição de energia para os níveis contínuos requeridos pelo amplificador. Em caso de sistemas independentes ou embarcados são utilizadas baterias ou geradores. Em sistemas hidráulicos ou pneumáticos a energia de movimentação do sistema é provida por um fluido pressurizado.

Amplificador

O sinal proveniente do controlador é apenas uma referência que diz como deve ocorrer o movimento dos motores, não possuindo energia necessária para realizar o deslocamento dos mesmos. Faz-se necessário então um elemento que baseado no sinal de controle, obtenha energia suficiente da fonte e a transfira para os atuadores, a esse elemento dá-se o nome de amplificador.

Segundo [4], de maneira resumida considera-se que o amplificador deve ser constituído pela junção de circuitos eletrônicos com as seguintes funcionalidades: alterar o sentido de rotação e regular a entrada de energia elétrica dos motores.

Para realização do primeiro propósito, normalmente utiliza-se um circuito eletrônico denominado ponte H, circuito este que pode ser visto na Figura 2-8.

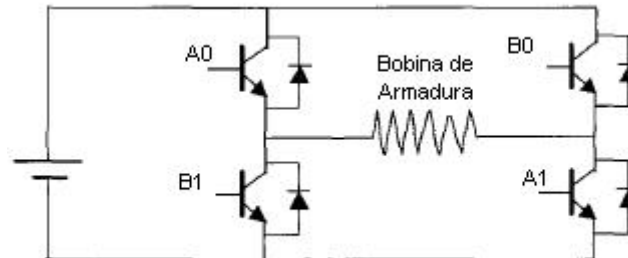


Figura 2-8

Sabe-se que para alterar o sentido de rotação de um motor elétrico de ímã permanente com escovas é necessário apenas inverter o sentido de circulação da corrente elétrica através de suas bobinas de armadura. A ponte H é constituída basicamente por quatro transistores, elementos que entre suas funcionalidades podem operar como chaves eletrônicas. Observando a figura x3, percebe-se que ao se ligarem apenas as chaves A0 e A1 o motor irá girar em uma dada direção e ao se ligarem apenas as chaves B0 e B1 o motor irá trocar seu sentido de rotação. Os diodos presentes no circuito servem para drenar a corrente reversa produzida quando as chaves são desligadas, impedindo assim que o circuito seja danificado.

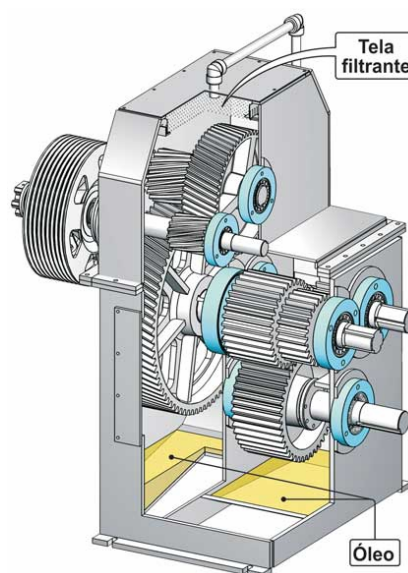
Para realizar a regulação da energia elétrica provida ao motor, pode-se utilizar o método de amplificação linear e o PWM. No primeiro, são produzidos níveis de tensão e corrente elétrica suficientes para movimentação dos motores, aplicados de

forma contínua e proporcionais aos sinais de controle. Neste caso, circuitos com amplificadores operacionais podem ser utilizados. No entanto, devido à baixa eficiência causada pela alta dissipação de calor, este método é usado apenas para sistemas de baixa potência. O método do PWM, modulação por largura de pulso, consiste em aplicar de forma descontínua, a tensão máxima ao motor. Estipula-se uma frequência na qual, pulsos de tensão são aplicados, faz-se variar o ciclo ativo dos mesmos proporcionalmente ao sinal de controle, conseguindo assim alterar o nível de energia elétrica efetiva aplicada aos motores. Este método apresenta maior eficiência quando comparado com o primeiro, uma vez que apresenta menor dissipação de calor nos componentes, devido ao fato da tensão não ser aplicada de forma contínua.

2.10.3 Transformadores de Potência Mecânica

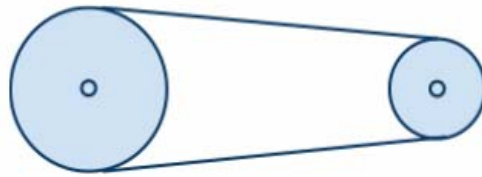
Muitas vezes a execução de movimentos nas juntas de um manipulador requer baixas velocidades e torques elevados, no entanto, os motores elétricos geralmente fornecem baixo torque a altas velocidades, deste modo, pode-se fazer necessária a utilização de mecanismos acoplados aos motores, capazes de produzir a relação velocidade torque necessária para uma dada aplicação. Conforme [2], esses mecanismos são normalmente chamados de transmissões e atuam de forma quantitativa (modificando relação torque velocidade) e qualitativa (por exemplo, conversão de movimento rotacional em linear.).

Existem alguns tipos de transmissões que podem ser utilizadas para transmitir a potência mecânica entre os motores e as juntas:



- Engrenagens

Figura 2-9



- Correias e Correntes

Figura 2-10

As engrenagens são as formas mais comuns de transmissão, podem ser encontradas desde mecanismos microscópicos (relógios, por exemplo) até sistemas que operam com milhares de HP de potência. Desde que perfeitamente montadas e lubrificadas, as engrenagens alcançam alta eficiência, podendo realizar tarefas como: mudança de velocidade de rotação; mudança na direção de rotação; multiplicação ou divisão do torque ou magnitude de rotação; conversão de movimento rotacional em linear e vice versa [6]. As correias e correntes são equivalentes sob o ponto de vista cinemático e podem ser empregadas quando o motor está localizado longe do eixo de atuação da junta do manipulador [2], as correias são capazes de absorver impactos e vibrações além de tolerarem algum desalinhamento entre as partes do sistema [3], já as correntes apesar de serem uma alternativa financeiramente atraente podem apresentar algumas desvantagens como: ruído e menor eficiência quando comparadas com as correias.

O manipulador protótipo construído não apresenta nenhum tipo de transmissão, uma vez que utiliza o sistema conhecido como Direct Drive, ao contrário do que é encontrado nos sistemas mecânicos convencionais, os eixos das juntas são diretamente acoplados aos motores resultando assim em características como: pequeno atrito, alta rigidez e ausência de folgas entre os dentes de engrenagens (backlash) [7]. A simplicidade deste tipo de mecanismo permite obter uma modelagem mais precisa da dinâmica do manipulador, essencial para realização de controle em tarefas de alta velocidade.

2.10.4 Elementos de Atuação

Normalmente, a atuação das juntas de um manipulador é realizada através de motores, referenciando a figura x1, percebe-se que de acordo com o tipo de energia de entrada P_{in} , os motores podem ser de diferentes tipos, como por exemplo, elétricos ou hidráulicos.

Segundo [2], o tipo de motor mais comumente empregado em robótica, são os motores elétricos, e entre estes os mais populares são os motores DC de ímã permanente e os motores DC sem escova (brushless).

Os motores DC de ímã permanente são constituídos por:

- Uma parte externa fixa denominada estator formado por ímãs que produzem um campo magnético constante.
- Uma parte interna chamada de armadura a qual é constituída por inúmeros enrolamentos de fio que transportam corrente elétrica ao redor de um núcleo ferromagnético rotativo chamado rotor.
- Um comutador que através de dispositivos denominados escovas permitem a conexão elétrica entre a armadura rotativa e a alimentação externa.

Os motores DC sem escova são constituídos por:

- Um núcleo rotativo (rotor) formado por um ímã permanente que gera fluxo magnético.
- Uma armadura estacionária (estator) constituída por enrolamentos.
- Um comutador estático que baseado em sinais provenientes de um sensor de posição localizado no eixo do motor, gera uma seqüência de alimentação elétrica para os enrolamentos do estator.

Os motores DC de ímã permanente apresentam algumas deficiências devido ao mecanismo de escovas necessário para a comutação de corrente nos enrolamentos da armadura, sendo que as principais são: aquecimento devido ao atrito das escovas, ruído excessivo, problemas de oxidação, criação de arcos elétricos [8]. Os motores DC sem escovas conseguem superar os problemas mencionados anteriormente, no entanto, necessitam de circuitos eletrônicos complexos capazes de criar a seqüência correta para acionamento das bobinas de estator.

Pelo fato do manipulador protótipo construído utilizar motores DC de ímã permanente, estes serão abordados com mais detalhes a seguir.

2.10.5 Motores DC de Ímã Permanente

Motor DC trabalha com o princípio de que uma corrente elétrica em um condutor quando submetida a um campo magnético experimenta uma força dada por:

$$F = i \times \theta$$

Onde θ é o campo magnético e i é a corrente no condutor. O motor consiste de um estator fixo e de uma parte girante chamada armadura, a qual é formada por inúmeros enrolamentos que circundam um núcleo denominado rotor. O estator produz um fluxo magnético radial θ que ao interagir com a corrente que circula nos enrolamentos da armadura produzem um torque que causa o giro do rotor. Como a armadura sofre rotação é necessária a utilização de um dispositivo denominado comutador, que através de escovas possibilitam a alimentação elétrica dos enrolamentos da armadura.

Os motores DC podem ser classificados pela maneira como o fluxo magnético é produzido e a armadura é montada. Os motores de ímã permanente possuem o fluxo θ constante e seu torque é controlado pela corrente de armadura [9].

A magnitude do torque produzido é dada por:

$$\tau = k_t \times i$$

Onde k_t é a constante de torque do motor e dependerá das características físicas do mesmo e i é a corrente que circula nos enrolamentos da armadura. Devido ao movimento do condutor (armadura) dentro do campo magnético, ocorre o aparecimento de uma tensão V_b nos terminais da armadura e essa é proporcional a velocidade de rotação do rotor, esta tensão é denominada força contra eletromotriz, podendo ser escrita por:

$$V_b = k_v \times \omega$$

Onde k_v é a constante de velocidade do motor e ω é a velocidade angular do rotor.

Como os motores apresentam fluxo magnético constante, é necessário apenas evidenciar o diagrama de blocos que representa a armadura, conforme visto na figura x5, e escrever sua respectiva equação:

$$V(t) = Ri + L \frac{di}{dt} + V_b$$

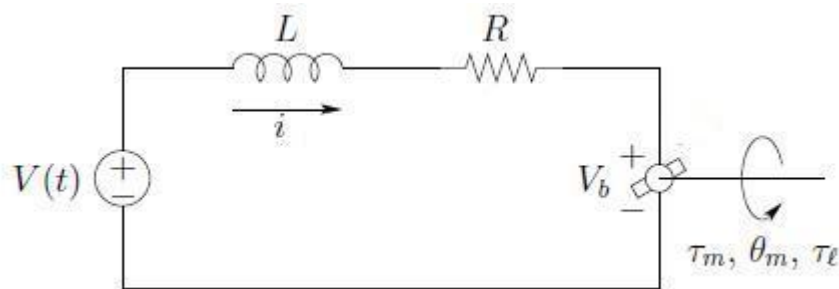


Figura 2-11

Onde $V(t)$, R e L representam respectivamente alimentação, resistência e indutância da armadura e V_b conforme mencionado anteriormente é a força contra eletromotriz.

Nota-se que a indutância primária ou indutância mútua nos enrolamentos da armadura está representada pela força contra eletromotriz, sendo assim a indutância L é geralmente negligenciada e representa parte do fluxo da armadura o qual não está interagindo com o estator e não é usado para geração de torque [8].

A equação mecânica do motor pode ser obtida aplicando a segunda lei de Newton ao rotor. Assumindo que o motor esta acoplado a uma carga que requer um torque T_L , e que o atrito na armadura possa ser modelado como viscoso [8], tem-se:

$$J_m \frac{d\omega_m}{dt} = T_m - T_L - b\omega_m$$

Onde J_m é o momento de inércia do eixo do rotor e b é a equivalente constante de amortecimento para o rotor.

2.10.6 Sensores

Em robótica, os sensores podem ser divididos em dois tipos: aqueles utilizados para monitoramento das características internas do manipulador e aqueles utilizados para o monitoramento do ambiente o qual o mesmo está interagindo. No presente trabalho serão abordados com mais profundidade aqueles pertencentes ao primeiro tipo, uma vez que são estes que estão presentes no manipulador protótipo. Os pertencentes ao segundo tipo serão apenas mencionados juntamente com referências onde poderão ser encontradas mais informações sobre os mesmos.

2.10.6.1 Monitoramento das características internas

A fim de se garantir que o manipulador esteja seguindo uma trajetória previamente definida, o algoritmo de controle exige o monitoramento em tempo real por meio de sensores, dos estados internos do sistema, ou seja, é necessário realizar medições das posições, velocidades e torques exercidos pelas juntas do manipulador.

2.10.6.2 Encoders Óticos

Encoders Óticos são usados para a medição de deslocamentos lineares e angulares. Estes dispositivos utilizam a luz como o meio para a transformação de movimento em sinais elétricos e são constituídos basicamente por uma ou mais máscaras perfuradas que permitem ou não a passagem de um feixe de luz infravermelha, gerado por um emissor que se encontra em um dos lados da máscara e captado por um receptor que se encontra do outro lado, e este, com apoio de um circuito eletrônico gera pulsos. Contando então o número de pulsos é possível se registrar o posicionamento e/ou velocidade. Para medições lineares a máscara é formada por um conjunto de uma ou mais linhas paralelas, já para medições angulares a máscara é formada por linhas radiais formando um disco [10]. Serão tratados aqui apenas os encoders para os sistemas rotacionais.

2.10.6.3 Encoders Incrementais

Os encoders óticos incrementais possuem uma máscara com furos igualmente espaçados. Com o movimento de rotação são produzidos dois sinais quadrados defasados de 90° um do outro, isto é conseguido através da montagem de dois sensores (fotodiodos) defasados, os quais recebem a luz produzida por um emissor (led) e que passa através da máscara. Estes dois sinais, conhecidos como sinais de quadratura, permitem determinar a direção de rotação, se no sentido horário ou anti-horário.

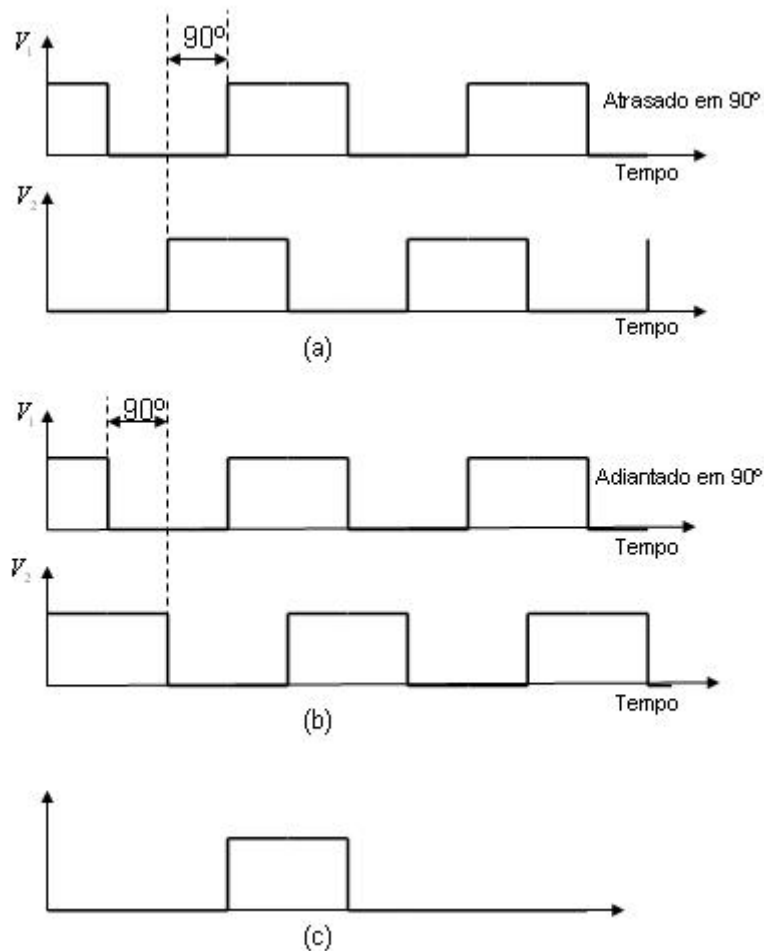


Figura 2-12

As Figura 2-12 a, b e c mostram as formas de onda de saída do encoder, quando o disco está girando em sentido horário e anti-horário respectivamente. Observando as figuras, percebe-se que para uma rotação no sentido horário V_1 está 90° atrasado em relação à V_2 e para uma rotação no sentido anti-horário V_1 está 90° adiantado em relação à V_2 . Deste modo, o sentido de rotação pode ser determinado a partir da diferença de fase entre os dois sinais. Segundo [8], um método para determinar a diferença de fase é temporizar os pulsos usando um sinal de clock de alta frequência. Por exemplo, se a operação de temporização é iniciada na borda de subida de V_1 e $n_1 =$ número de ciclos de clock até a borda de subida de V_2 e $n_2 =$ números de ciclos de clock até a próxima borda de subida de V_1 , então $n_1 > n_2 - n_1$ corresponde à rotação no sentido horário e $n_1 < n_2 - n_1$ corresponde à rotação no sentido anti-horário.

Os encoders incrementais necessitam de uma posição inicial a partir da qual iniciam a contagem e conseqüentemente determinação da posição angular, para este propósito, muitos encoders possuem além dos dois sinais de saída já mencionados, um

terceiro responsável por fornecer uma referência, a partir da qual se pode iniciar o sensoriamento, este sinal de referencia pode ser visto na figura x7c.

A resolução de um encoder representa a menor mudança que pode ser medida, sendo definida pelo número N de janelas existentes na máscara. Se apenas um sinal de pulsos é utilizado (sem detecção de sentido) e as bordas de subida dos mesmos são detectadas a resolução será dada por $(360/N)^0$. No entanto, se ambos os sinais de pulso (quadraturas) estiverem disponíveis e for possível realizar a detecção de bordas de subida e descida, pode-se utilizar o artifício de somar os sinais e assim conseguir que a resolução do encoder aumente em quatro vezes $(360/4N)^0$.

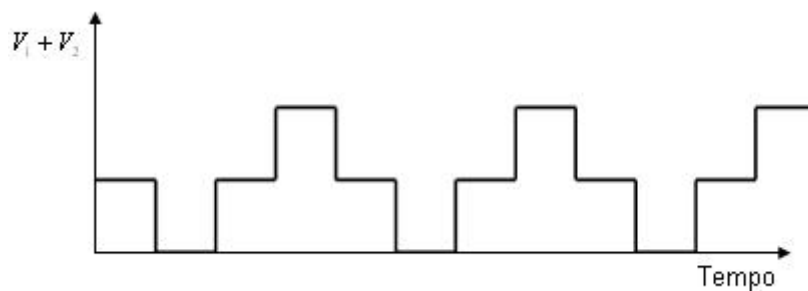


Figura 2-13

Figura x8 – Adição dos sinais de quadratura

A Figura 2-13 representa a soma dos sinais (Figura 2-12), percebe-se então que o sinal resultante apresenta quatro transições para cada ciclo do encoder. Pela detecção de cada transição, quatro pulsos podem ser contados dentro de cada ciclo principal.

Os encoders também podem ser utilizados para medição da velocidade de rotação de um eixo e para este propósito dois métodos são possíveis [8]: Método de contagem de pulso e Método de temporização de pulsos. Para determinação da velocidade angular ω usando o primeiro método, conta-se o número de pulsos n durante um dado período de amostragem T , deste modo, o tempo médio para um pulso será dado por T/n , se existem N janelas na máscara do encoder, o tempo médio para uma revolução completa será de NT/n , e assim $\omega = 2\pi n / NT$. Para determinação da velocidade usando o segundo método, estipula-se uma frequência de clock f Hz, contando m ciclos de clock durante um período do encoder (intervalo entre duas janelas consecutivas), o tempo médio será de dado por m/f , com o total de N janelas na máscara, o tempo médio por revolução será de Nm/f , e assim $\omega = 2\pi f / Nm$. O primeiro método possui a desvantagem de apresentar alguma imprecisão para medição de baixas velocidades, o que pode ser superado pelo segundo método com a utilização de uma alta frequência de clock.

2.10.6.4 Encoders Absolutos

O encoder absoluto se diferencia do encoder incremental por possuir vários sensores óticos que combinados entre si geram um código correspondente a dada posição. Seu elemento básico é um disco formado por um padrão de trilhas concêntricas, vários feixes de luz atravessam cada trilha para iluminar fotosensores individuais.

A grande vantagem dos encoders absolutos é o fato de que a posição é determinada pela leitura de um código e este é único para cada posição do seu curso, diferentemente dos incrementais, onde a posição é determinada a partir de um marco inicial. Para a codificação de saída dos encoders absolutos prefere-se a utilização do código Gray ao binário, já que esse por apresentar a característica de mudar o estado de apenas um bit quando ocorre incremento ou decremento de valor, impede que aconteçam leituras ambíguas no momento de transição de valores [8]. No entanto, são requeridos circuitos adicionais para realizar a conversão do código Gray para o binário.

A desvantagem de se utilizar encoders absolutos é o fato de que podem custar até duas vezes mais que os incrementais, já que apresentam mais complexidade de construção.

2.10.6.5 Sensores de Velocidade

Velocidades podem ser medidas indiretamente a partir de sensores de posição, no entanto, muitas vezes é preferível se medir diretamente utilizando sensores destinados a este propósito. Os transdutores de velocidade normalmente encontrados em aplicações robóticas são os tacômetros, os quais apresentam o funcionamento baseado nos princípios das máquinas elétricas. Os dois tipos básicos são os tacômetros DC e AC.

2.10.6.6 Tacômetros DC

O tacômetro DC é o mais encontrado em aplicações. Nada mais é do que um pequeno gerador DC, no qual o campo magnético é provido por um ímã permanente. Quando o rotor é colocado em rotação ocorre o aparecimento de uma voltagem DC proporcional a velocidade angular do mesmo. Devido à presença de comutadores, a tensão de saída apresenta uma oscilação residual, a qual não pode ser eliminada por filtros, já que sua frequência dependerá da velocidade de rotação do rotor, essa oscilação esta em torno de 2 a 5% do valor médio do sinal de saída [2]. O sentido da velocidade é determinado observando se a tensão gerada apresenta valor positivo ou negativo com relação ao referencial.

2.10.6.7 Tacômetros AC

Estes dispositivos apresentam duas bobinas separadas no estator. Uma destas é energizada com uma voltagem AC de referência. A tensão induzida na outra bobina é o sinal de saída do tacômetro. Quando o rotor está parado ou se movendo muito lentamente, a tensão induzida na segunda bobina é muito semelhante à tensão de referência. Quando o rotor se move a uma dada velocidade, uma tensão adicional

proporcional a velocidade do rotor é induzida na segunda bobina [8]. O sentido da velocidade é determinado baseando-se no fato de que a fase do sinal induzido na segunda bobina será igual ou inversa aquela da bobina de referência, dependendo da direção de rotação do rotor.

2.10.6.8 Sensores de Torque

De modo a conseguir compensar as não linearidades e os acoplamentos dinâmicos existentes em manipuladores, se torna necessário controlar o torque realizado pelos motores que constituem o sistema de atuação, sendo assim, são requeridos sensores que monitorem esta grandeza, seja de forma direta ou indireta. A forma direta faz-se uso de strain gages montados no eixo do rotor, de maneira a se obter diretamente o torque exercido pelo mesmo. A medição de forma indireta baseia-se no fato de que o torque exercido pelo motor é diretamente proporcional a corrente que circula em sua armadura, sendo assim ao se inserir um componente capaz de medir esta corrente é possível através da equação $\tau = k_t \times i$ se obter o torque. Em [11] encontrar-se um estudo detalhado destes dois modos de medição em um manipulador direct drive.

2.10.7 Monitoramento do Ambiente

O objetivo dos sensores utilizados para o monitoramento externo é obter características da interação entre os robôs e o ambiente, de forma a aumentar o grau de autonomia do sistema. Esta seção pretende mencionar de forma superficial alguns destes sensores, já que os mesmos não estão presentes no manipulador protótipo.

2.10.7.1 Sensores de proximidade

Sensores de proximidade podem ser utilizados para que o robô consiga detectar obstáculos e evitar que ocorram colisões, um tipo básico de sensor de proximidade são os ultrasônicos. Estes são dispositivos de baixo custo, capazes de medir a distancia para objetos que se encontram tipicamente entre 0,3 e 10 metros do robô [4]. O funcionamento desses sensores baseia-se no envio de ondas de ultra-som, que ao encontrarem algum obstáculo são refletidas e assim captadas pelo sensor. Sabendo então que a velocidade do som é uma constante e contando o tempo de envio e recepção dos sinais de ultra-som é possível calcular as distancias para os obstáculos.

2.10.7.2 Sensores de Força

Muitas aplicações na área de robótica requerem que se controle a força que o robô está exercendo sobre um determinado objeto presente no ambiente, como por exemplo, tarefas onde o robô deva executar operações de corte e perfuração, deste modo, o sensoriamento dessa grandeza se torna importante. Muitos sensores de força são baseados em medições de strain gages. Estes elementos são basicamente resistores de valor conhecido que ao serem submetidos a uma tensão mecânica tem seu valor de resistência modificado, utilizando-se então de circuitos eletrônicos específicos é

possível se obter medições que podem ser relacionadas diretamente com a força aplicada [8].

2.10.7.3 Sensores de visão

Sensores de visão permitem que o robô obtenha informações geométricas e qualitativas do ambiente que o circunda de modo a executar controle e planejamento de movimentos [2]. As câmeras se caracterizam como sensores de visão e sua função é medir a quantidade de luz refletida por objetos que se encontram no ambiente, para tal fim são empregados elementos fotosensíveis denominados pixels, os quais convertem energia luminosa em energia elétrica [2].

Vasta é a bibliografia relacionada a sensores para monitoramento do ambiente, existem inúmeros tipos e várias aplicações relacionadas. Para maiores informações a respeito do assunto pode-se consultar [12], no qual se encontram sistemas autônomos dotados visão. [8] é uma boa referência para sensores de força, já que apresenta toda teoria necessária pra utilização de strain gages.

3 Modelo experimental

Modelagem matemática do experimento

O modelo experimental proposto para este trabalho é o modelo de manipulador paralelogramo. Este modelo, como já dito antes é bastante utilizado na indústria. Ele possui diversas formas de ser representado matematicamente [14][2][9] e a representação adotada foi:

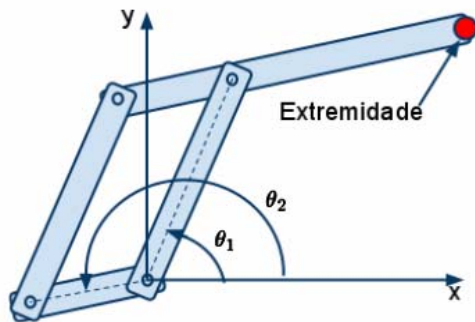


Figura 3-1

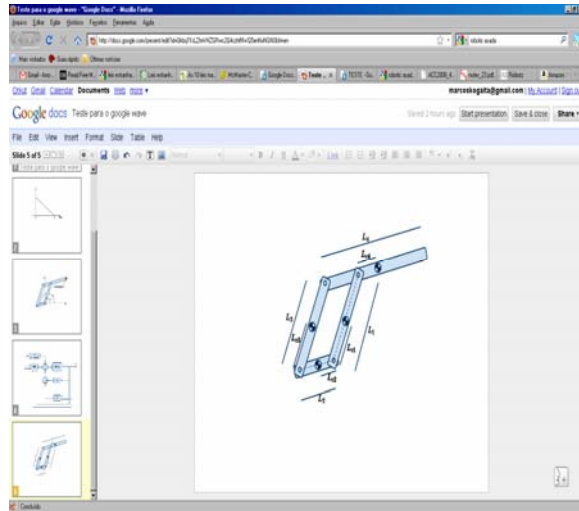


Figura 3-2

As figuras Figura 3-1 e Figura 3-2 definem o que representam as variáveis utilizadas ao longo do projeto descrito. Logo, processo análogo ao descrito no capítulo teórico deve ser feito pra calcular-se e definir-se as características físicas do manipulador de acordo com alguns objetivos.

Cinemática direta:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 \cos(\theta_1) - (L_4 - L_2) \cos(\theta_2) \\ L_1 \sin(\theta_1) - (L_4 - L_2) \sin(\theta_2) \end{bmatrix}$$

Cinemática inversa:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{x}{y}\right) - \frac{\cos^{-1}\left(-\frac{x^2 + y^2}{4L_1^2 + 2}\right)}{4L_1^2 + 2} \\ \tan^{-1}\left(\frac{x}{y}\right) + \frac{\cos^{-1}\left(-\frac{x^2 + y^2}{4L_1^2 + 2}\right)}{4L_1^2 + 2} \end{bmatrix}$$

Jacobianos:

$$J = \begin{bmatrix} -L_1 \sin(\theta_1) & (L_4 - L_2) \sin(\theta_2) \\ L_1 \cos(\theta_1) & (L_4 - L_2) \cos(\theta_2) \end{bmatrix}$$

$$J_{C_1} = \begin{bmatrix} -L_{C_1} \sin(\theta_1) & 0 \\ L_{C_1} \cos(\theta_1) & 0 \end{bmatrix}$$

$$J_{C_2} = \begin{bmatrix} 0 & L_{C_1} \sin(\theta_2) \\ 0 & L_{C_2} \cos(\theta_2) \end{bmatrix}$$

$$J_{C_3} = \begin{bmatrix} -L_{C_1} \sin(\theta_1) & -L_{C_2} \sin(\theta_2) \\ L_{C_1} \cos(\theta_1) & L_{C_2} \cos(\theta_2) \end{bmatrix}$$

$$J_{C_4} = \begin{bmatrix} -L_{C_1} \sin(\theta_1) & -L_{C_4} \sin(\theta_2) \\ L_{C_1} \cos(\theta_1) & L_{C_4} \cos(\theta_2) \end{bmatrix}$$

Matriz de Inércia:

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$$

$$H_{11} = I_1 + m_1 L_{C_1}^2 + I_3 + m_3 L_{C_3}^2 + m_4 L_1^2$$

$$H_{22} = I_2 + m_2 L_{C_2}^2 + I_4 + m_4 L_{C_4}^2 + m_3 L_2^2$$

Para descobrir os termos $H_{12} = H_{21}$ utiliza-se a energia cinética.

$$T = \sum_{i=0}^n m_i \dot{q}^T J_{C_i}^T J_{C_i} \dot{q}$$

$$H_{12} = H_{21} = (m_3 L_2 L_{C_3} - m_4 L_1 L_{C_4}) \cos(\theta_1 - \theta_2)$$

As características do manipulador foram calculadas para que este termo de acoplamento da inércia das juntas fosse nulo e assim, facilitasse o controle das mesmas. Logo, para que esta propriedade não dependa da posição das juntas é necessário que:

$$\frac{m_3 L_{C_3}}{m_4 L_{C_4}} = \frac{L_1}{L_2}$$

Termos de Kristofel:

$$h_{ijk} = \frac{dH_{ij}}{dq_k} - \frac{1}{2} \frac{dH_{jk}}{dq_i}$$

$$h_{111} = h_{222} = 0$$

$$h_{112} = h_{212} = -h_{121} = -h_{221} = \frac{\alpha}{2} \sin(\theta_1 - \theta_2)$$

$$h_{122} = -h_{211} = \alpha \sin(\theta_1 - \theta_2)$$

Onde :

$$\alpha = (m_3 L_2 L_{C_3} - m_4 L_1 L_{C_4})$$

Além de respeitar a equação de desacoplamento das inércias, outros objetivos foram estipulados para dimensionamento do manipulador. Capacidade de deslocamento, maximização da área de trabalho e maximização da destreza. Com estes três objetivos foram estipuladas as medidas.

3.1 Maximização da destreza

A destreza é um conceito abstrato para a língua portuguesa, mas bem definido na robótica como a capacidade de se movimentar em todas as direções dados graus de liberdade dos atuadores. Assim se em uma dada posição o manipulador não puder se movimentar em uma certa direção quando em alguma posição, esta posição é uma

posição de singularidade e falta destreza ao manipulador para se movimentar nesta direção singular. Para maximizar esta característica é preciso pensar além das posições de singularidade.

O jacobiano como visto antes relaciona o deslocamento das juntas ao deslocamento da extremidade. Assim, se encarar-se a matriz jacobiana como a transformação linear entre o espaço das velocidades das juntas para o espaço velocidade da extremidade, percebe-se que os autovetores da matriz formam uma base ortogonal que define o espaço de velocidades e seus autovalores definem o quanto este autovetor contribui para o deslocamento. As posições de singularidade nada mais são do que posições onde um dos autovalores é nulo. Nestas o manipulador não pode se mover na direção do autovetor associado ao autovalor nulo.

Logo, ao ajustarem-se os autovalores do jacobiano para serem iguais em ambas as direções, está se ajustando a igualdade de esforço que os motores devem fazer para movimentar o manipulador para qualquer direção. Assim, cada motor fica responsável por uma direção não sobrecarregando estes em direção alguma enquanto outra fica folgada. Esta condição pode ser atingida quando:

Autovalores:

$$J - \lambda I = 0$$

$$\lambda = \frac{-(L_1 s_1 + (L_4 - L_2) c_2) \pm \sqrt{(L_1 s_1 + (L_4 - L_2) c_2)^2 - 4(L_1(L_4 - L_2)(s_1 c_2 - c_1 s_2))}}{2}$$

Para igualar os autovalores a igualdade deve ser estabelecida:

$$\Delta = 0$$

$$\therefore L_1 = (L_4 - L_2)$$

3.2 Definição do comprimento dos elos

Os elos nesta fase podem ter seu comprimento estimado uma vez que o único elo que já existem três relações que definem as características dos elos. Como próximo passo leva-se em conta a capacidade de sustentação do manipulador. Esta é definida pelos torques máximos nos motores, que são, de acordo com suas folhas de dados no Anexo B, iguais a 4,05Nm. Somando-se os torques nos motores, o torque máximo realizado sobre o eixo compartilhado destes é de 8Nm.

Deseja-se então avaliar qual o torque dos motores na posição onde os elos estão totalmente esticados e na horizontal, posição de maior torque estático. Desta forma pode-se traçar um gráfico variando-se L_1 e avaliando este torque necessário.

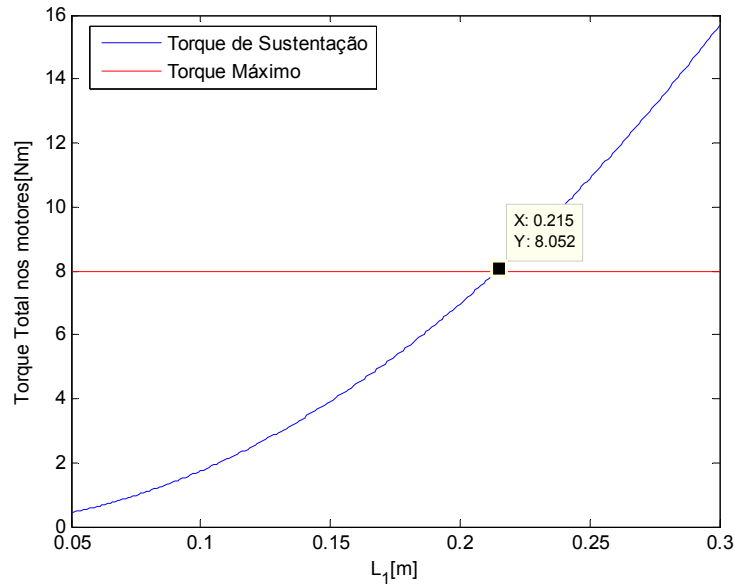


Figura 3-3

O gráfico Figura 3-3 mostra que o tamanho para o elo 1 deve ser menor que 21 cm dadas as considerações acima. Arredondando-se para baixo para garantir a alguma folga, o elo é estipulado em 20 cm. O elo 3 deve ter o mesmo comprimento do elo 1 por definição do modelo de manipulador utilizado. Logo com a relação entre os elos 1,2 e 4 definida, basta definir ou elo 2 ou 4 para concluir a definição dos elos.

3.3 Maximização da área de trabalho x Inércia

Os próximos objetivos para caracterização das medidas do manipulador são minimizar a inércia do manipulador para que este possa alcançar velocidades maiores e ao mesmo tempo é desejado que a área de trabalho seja a maior possível. Como só existe neste ponto uma variável a ser ajustada L_2 é configura-se um problema de multiobjetivo, situação onde dois ou mais objetivos são antagônicos. Isso quer dizer que para minimizar a inércia diminui-se a área de trabalho (não desejado) e para aumentar a área de trabalho aumenta-se a inércia. Esta relação pode ser claramente vista no gráfico abaixo. São representados a área de trabalho e o valor do máximo autovalor da matriz de inércia.

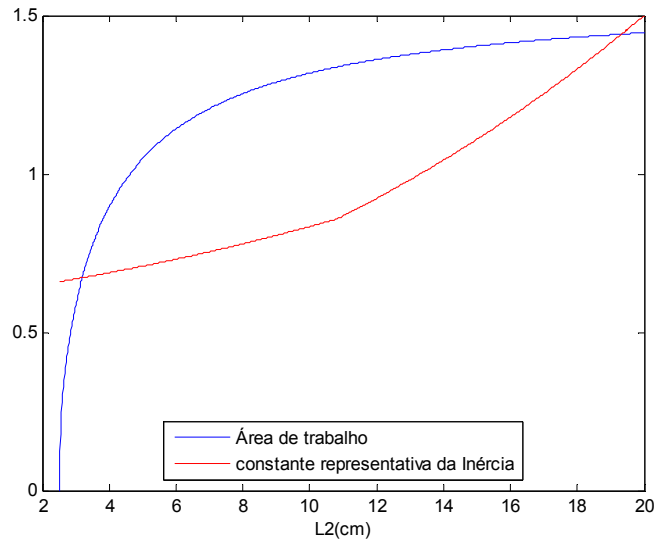


Figura 3-4

Para eliminar o problema de multiobjetivo a forma tradicionalmente adotada é traçar a curva de Ótimos de Pareto. Esta técnica busca a região do espaço onde não há como estabelecer ótimos absolutos através de dominância. O conceito de dominância inicialmente descrito por Pareto fala que uma solução domina a outra se sua avaliação em pelo menos um objetivo é melhor que a da solução dominada e as restantes são iguais [Siarry]. Como as funções de avaliação são monotônicas, possuem apenas uma variável e esta é comum, a fronteira de Ótimos de Pareto é o próprio espaço das funções de avaliação.

Para resolver o dilema multiobjetivo, outra solução muito utilizada é a criação de uma única função objetivo que englobe as anteriores [Siarry]. Na literatura são encontradas muitas formas de fazê-lo: Combinação linear, Produto, Minimização das energias, Distância ao alvo, etc. Para este trabalho foi utilizado o Produto das funções. Porém uma pequena adaptação é necessária, pois as funções têm prerrogativas distintas, uma quer ser minimizada e a outra maximizada. Uma forma de resolver este problema é inverter uma das funções e a escolhida foi a área de trabalho. Desta forma o valor ótimo encontrado foi de $L_2 = 8\text{cm}$ para minimizar a função:

$$F = \frac{\max(\text{etg}(H))}{\text{Área}_{\text{Trabalho}}}$$

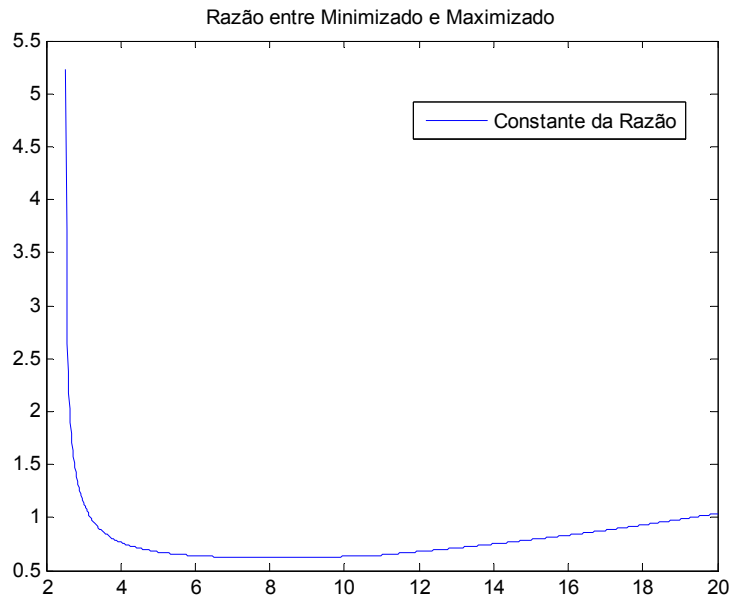


Figura 3-5

Esta abordagem se mostrou eficiente, pois foi de rápida implementação e comparando os dados com os de manipuladores comerciais razões entre as métricas encontradas são recorrentes. A tabela abaixo mostra algumas comparações feitas.

Tabela 1

Manipuladores	$\frac{L_2}{L_1}$	$\frac{L_1}{L_4}$
Este Trabalho	0,4	1
IRB 4400 (ABB)	0,37	0,95
IRB 660 (ABB)	0,36	0,97
KR 100-2 (KUKA)	0,35	0,92

3.4 Modelo Computacional

Neste estágio do projeto do manipulador já é possível criar um modelo computacional do manipulador. Este modelo ajuda a prever as propriedades físicas e construtivas do projeto. Foi utilizado o software Solidworks® onde é possível modelar cada uma das peças separadamente e verificar como a junção destas se comporta.

Assim pode-se definir que para juntar os elos seriam utilizados eixos de aço. Estes, se acoplados a motores seriam fixados aos elos por chavetas. Quando não, estes são fixados dois elos e seu deslocamento é restrito por colares de alumínio. Para diminuir o atrito entre os elos são colocados rolamentos axiais de rolete. Quando um eixo passa por dois elos um destes recebe uma bucha de bronze para que o atrito seja diminuído entre o eixo e o elo. No outro elo não há necessidade, pois o eixo desliza sobre o elo com menos atrito e se mantém preso ao outro.

Uma placa de Alumínio também foi providenciada para que tanto o manipulador pudesse ser acoplado independente de onde seria instalado fisicamente. Foi pensado também que nesta plataforma deveriam ficar o controlador CompactRio, o Amplificador

Roboteq , a bateria, motores e sensores. Alguns cortes na placa foram feitos para que a passagem do manipulador não fosse restrita para rotações de $270^\circ \geq \theta_2 > 25^\circ$.

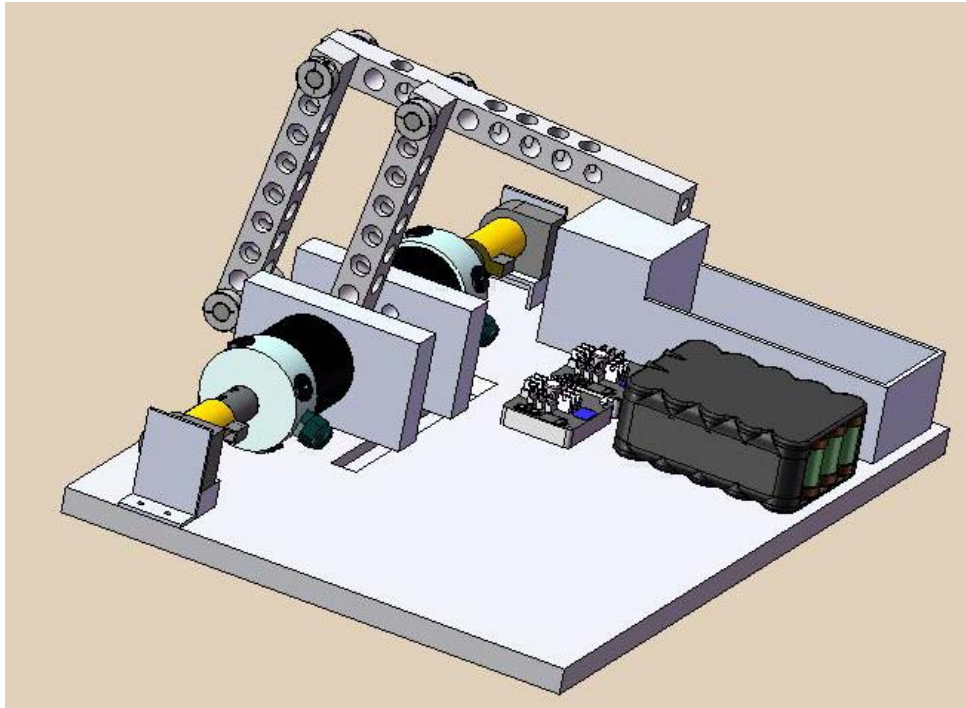


Figura 3-6

3.5 Fonte de Energia

Os elementos de atuação do manipulador protótipo construído são motores elétricos, deste modo, uma fonte de energia elétrica foi requerida, devido à dificuldade de se obter uma fonte capaz de produzir elevados níveis de corrente contínua requeridos pelos motores, optou-se por utilizar baterias. O tipo de bateria utilizada foi níquel-cádmio (NiCd) que conforme mencionado em [3] é capaz de fornecer altos níveis de corrente, além de ser recarregável, a tensão nominal desta bateria é de 24Volts e sua capacidade de carga é de 3600mAh. A figura x2 mostra a bateria utilizada.



Figura 3-7

3.6 Controlador

O CompactRio® controlador da National Instruments é utilizado como controlador do manipulador. Ele possui um módulo FPGA para configuração e gerenciamento de entradas e saídas de dados que possibilita um ganho muito grande na velocidade de pré-processamento das entradas e pós-processamento das saídas. Tanto que o controle de corrente nos motores é feito na sua FPGA.



Figura 3-8

O controle de posição/trajetória é calculado no módulo de controle onde é estipulada quais as corrente necessárias de se mandar para os motores através das leituras dos sensores e objetivos preestabelecidos de controle ou estabelecido de simultaneamente. Este módulo permite comunicação com computador supervisor para aquisição e apresentação de dados através de uma rede ethernet. Através desta rede também é possível dar download de novas configurações e programas de controle.

3.7 Amplificador

O manipulador construído apresenta dois motores elétricos, deste modo, foi especificado o amplificador de dois canais independentes RoboteQ AX2550, este tem capacidade de operar de 12 a 48 Volts contínuos até 120 Ampères por canal, utiliza o método de modulação por largura de pulso (PWM) com uma frequência de 16khz [5]. É possível ver uma foto do amplificador na Figura 3-9.



Figura 3-9

No anexo A, podem ser encontradas as configurações realizadas no amplificador para que este pudesse ser utilizado no desenvolvimento do manipulador protótipo, dentre as mais importantes pode ser citado que esse foi configurado para operar com sinais de controle analógico de 0 a 5 Volts contínuos para ambos os canais, isto significa que, para uma entrada de controle de 2,5Volts, o respectivo motor encontram-se parado, para uma entrada de 0 Volts, máxima tensão da bateria é aplicada em um sentido de rotação e finalmente ao se aplicar 5 Volts de controle o motor muda seu sentido de rotação com a máxima tensão aplicada. No anexo A, também pode ser encontrada uma vista do amplificador, na qual, evidenciam-se as ligações dos motores e das fontes de energia (baterias).

3.8 Atuadores

Para o manipulador protótipo foram especificados dois motores MagMotors® S28 150, estes são motores DC com escova e de ímã permanente, na Figura 3-10 pode ser vista uma foto do atuador especificado.



Figura 3-10

No anexo B são encontradas as características mecânicas e elétricas deste motor.

3.9 Sensores

3.9.1 Encoder do Manipulador Protótipo

No projeto do manipulador protótipo utilizaram-se dois encoders incrementais, um para cada junta do sistema, estes foram diretamente acoplados aos eixos dos motores. Estes encoders apresentam uma máscara com 512 furos o que resultaria em uma resolução de $0,7031^\circ$, no entanto foi utilizada a técnica descrita anteriormente, na qual realiza-se a soma dos sinais de quadratura e faz-se a detecção das bordas de subida e descida, com isto foi possível obter uma resolução de $0,1758^\circ$. Na Figura 3-11, pode-se ver o disco perfurado do encoder utilizado.



Figura 3-11

Observando a Figura 3-11 percebe-se que a periferia do disco é formada por ranhuras que deixam passar luz em pontos igualmente espaçados, possibilitando assim realizar a contagem de pulsos e determinação da posição angular.

O encoder utilizado não apresenta pulso de referência, sendo assim, o programa de controle apresenta inicialmente uma rotina que aguarda que o operador coloque manualmente os elos do sistema em uma posição conhecida, a partir da qual os pulsos são contados e a posição angular estabelecida.

Para detecção do sentido de rotação, baseia-se no fato de que ao se somar os sinais das Figura 3-12 a e b, para obtenção dos sinais de soma de quadratura para rotação nos sentidos horário e anti-horário respectivamente, resulta-se as formas de onda vistas na figura x10.

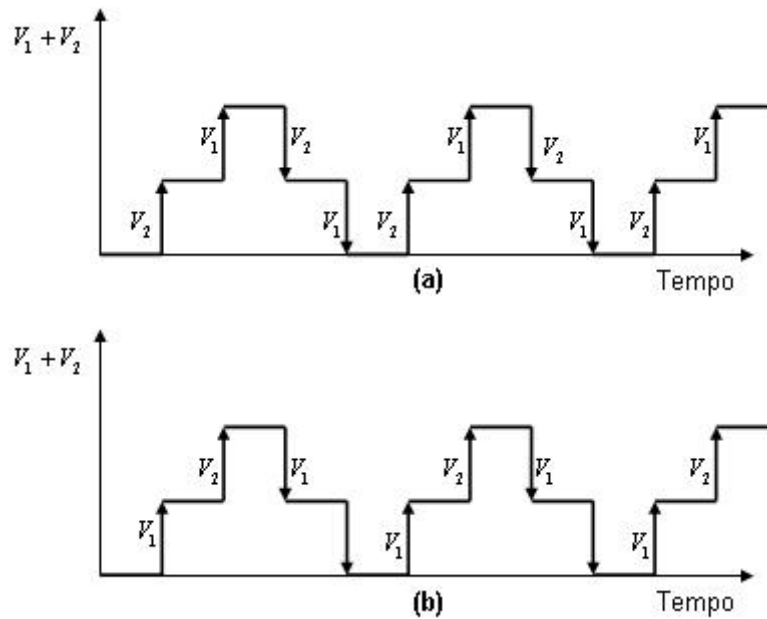


Figura 3-12

Observando então a Figura 3-12, criou-se uma lógica booleana responsável por indicar o sentido de rotação, essa tem como entradas a existência ou não de transições positivas ou negativas dos sinais V_1 , V_2 e seus respectivos níveis: alto ou baixo. Um pulso no sentido horário deve ser contado se alguma das seguintes condições ocorrer: Transição positiva de V_2 e sinal V_1 em estado baixo; transição positiva de V_1 e V_2 em estado alto; transição negativa de V_2 e sinal V_1 em estado alto; transição negativa de V_1 e V_2 em estado baixo. Se por acaso nenhuma das condições anteriores ocorrer, sabe-se que o pulso deve ser contado no sentido anti-horário. Por convenção, adotou-se que a partir do referencial que significa ângulo zero grau, a rotação no sentido anti-horário é positiva e no sentido horário é negativa.

No projeto do manipulador, o encoder foi utilizado apenas como sensor de posição, deixando que tacômetros fossem responsáveis pela aquisição das velocidades das juntas.

3.9.2 Tacômetros do Manipulador Protótipo

O manipulador protótipo apresenta tacômetros DC acoplados aos rotores dos atuadores elétricos, possibilitando assim obter um nível de tensão DC proporcional a velocidade de rotação dos motores.

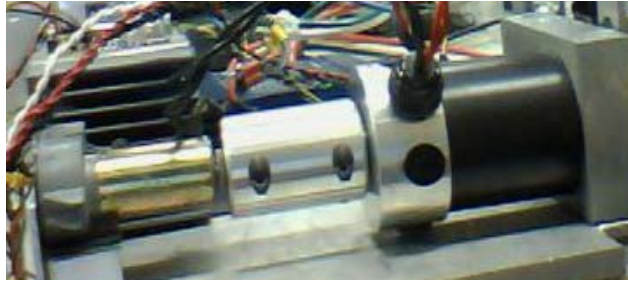


Figura 3-13

Foi realizada uma calibração dos tacômetros DC, de modo a se obter uma relação entre tensão de saída dos mesmos e velocidade em rotações por minuto dos atuadores. Para tal fim foi utilizado um tacômetro a laser.



Figura 3-14

O seguinte procedimento foi então utilizado: Os motores foram colocados em rotação com diferentes velocidades, onde para cada uma, foi realizada a medição do sinal de saída dos tacômetros, em volts, e a medição direta da velocidade utilizando o tacômetro de referência, deste modo, foi possível a obtenção dos gráficos com as respectivas relações.

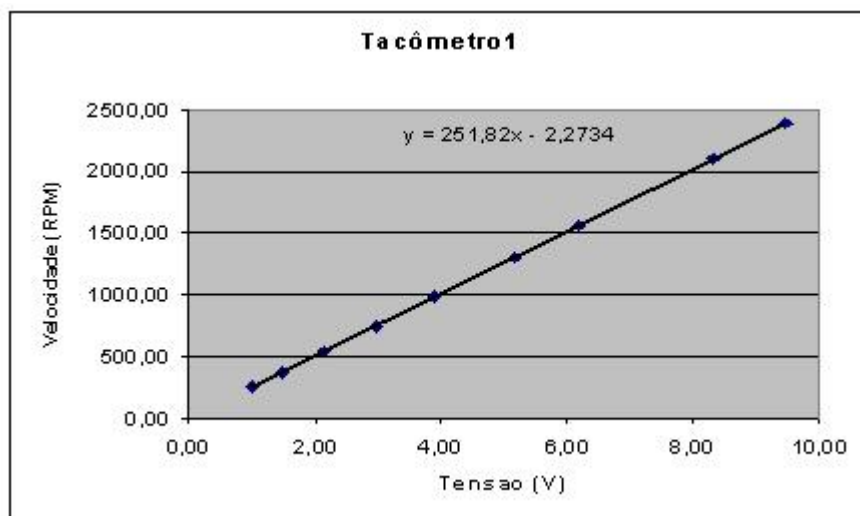


Figura 3-15

Gráfico x1 - Relação Tacômetro 1

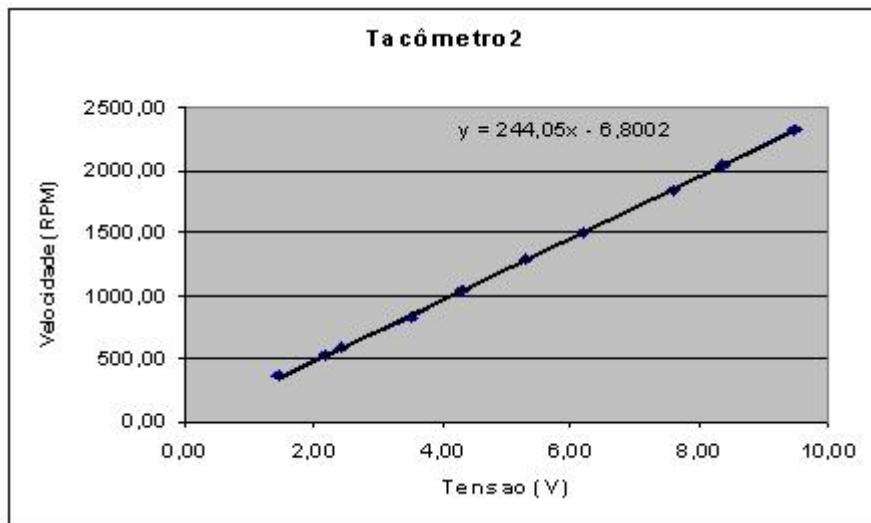


Figura 3-16

Gráfico x1 - Relação Tacômetro 2

Os gráficos Figura 3-15 e Figura 3-166 representam as relações entre tensão e velocidade para os tacômetros utilizados, é possível verificar também as equações de primeiro grau geradas a partir destes gráficos e evidenciadas abaixo, onde as equações representam respectivamente o tacômetro 1 e o tacômetro 2.

$$RPM = (251,82 \times \text{tensão}) - 2,2734$$

$$RPM = (244,05 \times \text{tensão}) - 6,8002$$

3.9.3 Medição de Torque

O manipulador protótipo construído possui o método de medição indireta, e para este fim, são utilizados resistores shunt ligados em série com a armadura do motor. Os resistores foram obtidos a partir de uma associação em paralelo de dez resistores de $100\text{ m}\Omega$ com $\pm 1\%$ cada um e potência de 10 W , o que resulta num resistor equivalente de $10\text{ m}\Omega$ capaz de dissipar até 100 W e sensibilidade de 10 mV/A , deste modo, conseguir-se-ia medir correntes de até 100 A , no entanto, estabeleceu-se no controle um limite máximo para os valores de corrente bem abaixo do máximo de 100 A .

3.10 Montagem

A montagem do manipulador sobre a plataforma consumiu 4 meses do projeto pois foi necessário fazer a compra de todos os materiais, alguns deles não encontrados no Brasil como é o caso dos colares. Além de fazer toda a construção mecânica da plataforma e dos elos. Estes cortes e furos foram feitos de acordo com as plantas dos elos, apresentadas no anexo D.

Algumas fotos da montagem são mostradas

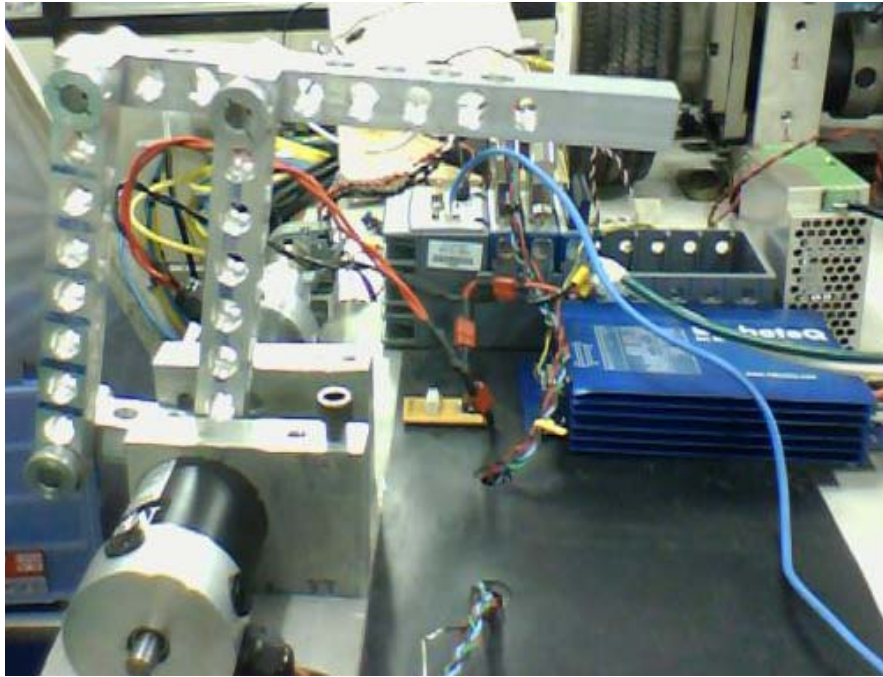


Figura 3-17

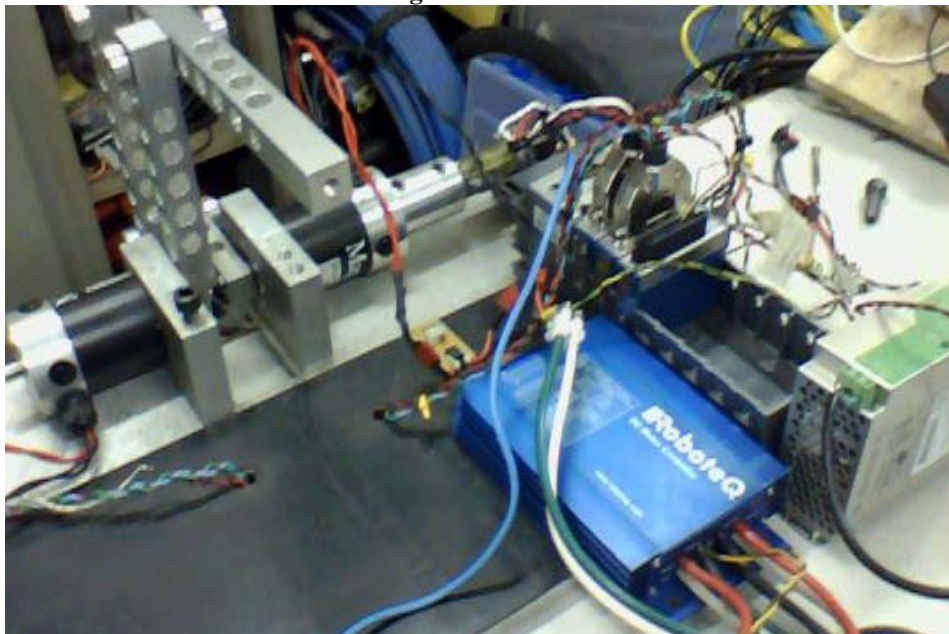


Figura 3-18

4 Simulação

4.1 Considerações Gerais

Naturalmente como a simulação é computacional todas as variáveis devem ser discretizadas, assim o tempo de simulação também o é. É estabelecido que a condição de parada é atingir o tempo total de simulação estabelecido inicialmente. Assim, este tempo deve ser discretizado para tanto é estabelecido um passo fixo para a contagem de tempo da solução numérica da dinâmica a ser simulada. O passo determinado de forma que existam pelo menos 10 passos temporais para cada atuação do controle. Como o controle real atua a uma frequência de 1MHz é de 0.1 μ s.

São feitas então as considerações iniciais a respeito das características físicas do manipulador.

Elos são considerados blocos paralelepípedos.

Massa, centro de gravidade e Tensor de Inércia com relação ao centro de gravidade são estabelecidas a partir do modelo no software SolidWorks. Mas podendo ser calculados com densidade uniforme e inércias de paralelepípedos.

É estabelecida uma matriz de atrito para as juntas a ser estabelecida segundo modelo de atrito viscoso.

Características do ambiente são fixadas:

O Campo gravitacional estabelecido como constante e igual a 9.79 m/s² como estabelecido pelo IMETRO dentro dos laboratórios de física da PUC-Rio;

O solo é considerado com reações do tipo mola e amortecimento na direção normal. Atrito viscoso é considerado também na direção paralela ao solo.

Torque máximo e constantes são iguais aos descritos pela configuração dos motores no anexo B.

Seguidamente as características de controle PID são descritas por Heurística clássica mas podendo ser modificadas:

- Proporcional: Kp exige torque máximo caso erro seja maior que 10% de curso máximo;
- Derivativo: é estimado como um décimo do ganho proporcional;
- Integral: considerado como igual ao proporcional.

4.2 Ciclos de simulação

O ciclo de simulação estima o estado do manipulador a cada Δt segundos. Neste passo são calculadas a cada intervalo de tempo todas as matrizes características do manipulador citadas no capítulo anterior. São também calculadas as posições desejadas das juntas de acordo com o plano de trajetória passado utilizando a função de cinemática inversa. Seguidamente a interação com o ambiente é verificada e calculada.

No próximo passo seguindo o tempo de ciclo do controlador, que é mais lento que a simulação do estado, são recalculados os torques nos motores. Estes são mantidos fixos para o intervalo de tempo em que o controlador não modifica estes valores. O controle também pode ser ajustado, para validação do modelo, através da corrente lida nos motores. Assim, pode-se simular uma ação de controle ocorrida e verificar se os valores das posições das juntas são correspondentes.

Alguns tipos de controles foram programados para o simulador e outros não foram finalizados. Assim, existe a opção de utilizar os controles:

- Sem controle
- Proporcional Derivativo (PD)
- Proporcional Derivativo e Integral (PID)
- PD com Compensação de gravidade (PD+G)
- Controle de Torque Computado (CTC)
- Controle de Torque Computado com parâmetros estimado para o controle (CTC). Este serve para testar o comportamento do controle caso os parâmetros estimados não estejam corretos
- Controle Robusto com PD+G

Espaço foi proposto para a inclusão de outros dois controles: Adaptativo e por algoritmos genéticos.

É importante notar que todo controle deve respeitar uma restrição imposta pelo método de integração que é a de isolar os fatores de torque que são proporcionais às variáveis de posição e velocidade das juntas do manipulador. Desta forma todo e qualquer controle deve registrar duas parcelas uma contendo somente fatores relacionados às variáveis medidas das juntas e outro contendo os demais fatores de controle. Tratamento dos torques é feito para limitar o torque total ao torque saturado.

O estado do manipulador é mostrado em tempo real para acompanhamento visual de seu controle. Também são armazenados os dados de torque, posição e velocidade das juntas que são apresentados posteriormente na forma de gráficos.

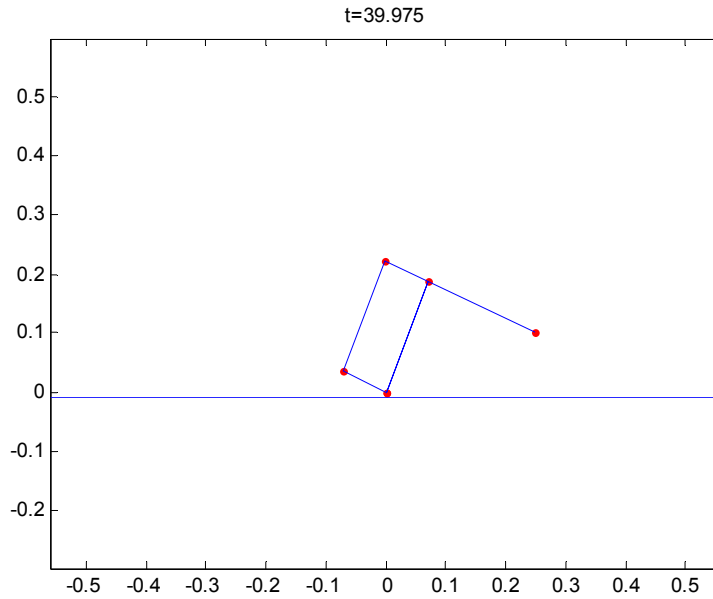


Figura 4-1

4.3 Resultados

Nesta secção serão mostrados e comentados os resultados dos controles estabelecidos. O manipulador começa na posição inicial $x = 20\text{cm}$ e $y = 0$. São considerados 3 destinos de posição para o manipulador o primeiro próximo a posição inicial, o segundo mais distante e o terceiro mostrando o contato com o solo onde o manipulador tem como objetivo ponto abaixo do solo.

4.3.1 PD

O controle PD se mostra eficiente dada suas limitações conhecidas. Como a teoria de controle Clássico mostra um “Off –set” entre o estado estacionário e a posição desejada. A gravidade influencia este “off-set” diferentemente em cada posição. Para a primeira posição desejada a distancia entre a posição e a posição desejada é menor do que na segunda posição pois o manipulador está mais contraído e seu centro de gravidade está menos afastado da base dos motores. Isto pode ser visto na Figura 4-2 que mostra a posição desejada das juntas ao longo do tempo e a posição das mesmas no tempo. A Figura 4-5 que mostra a posição da extremidade também mostra bem este resultado.

A Figura 4-4 que mostra o torque dos motores indica que o manipulador não tenta ir na direção do solo. Uma vez que a ultima posição desejada está abaixo do solo e este não possui nenhum torque saturado.

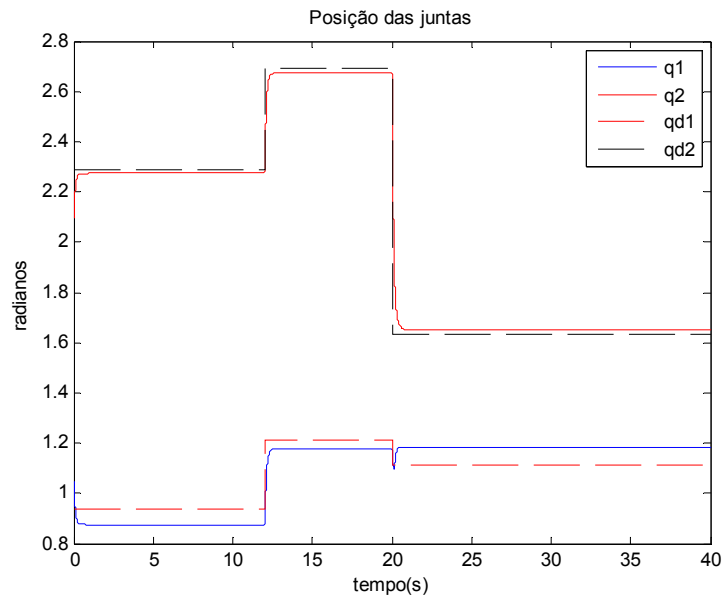


Figura 4-2

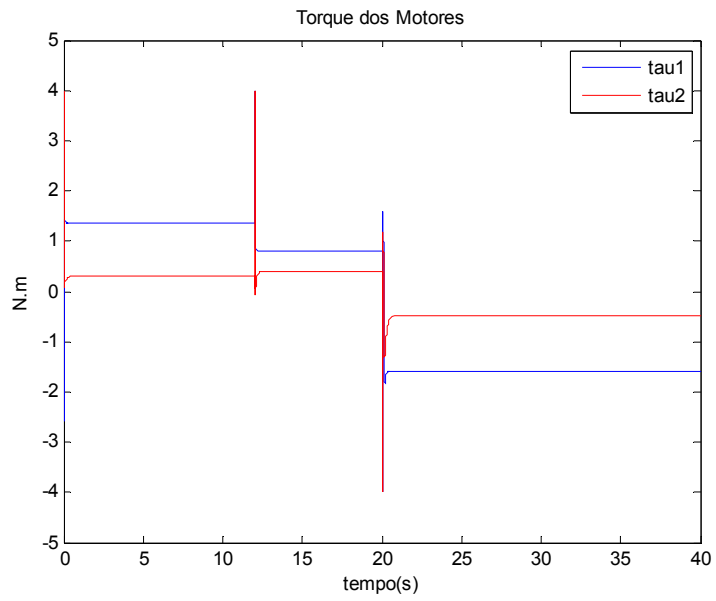


Figura 4-3

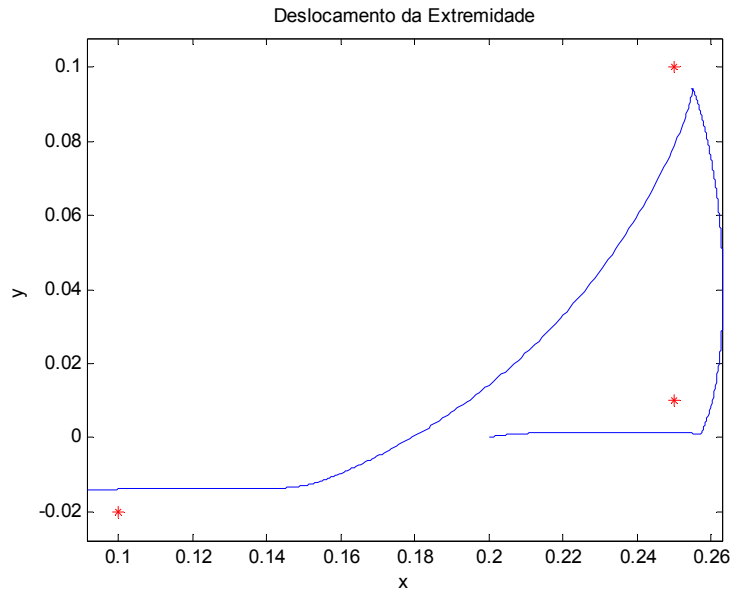


Figura 4-4

4.3.2 PD+G

A compensação de gravidade ajuda no controle PD a diminuir a distancia para os pontos desejados. Claramente vista na Figura 4-5 e Figura 4-7.

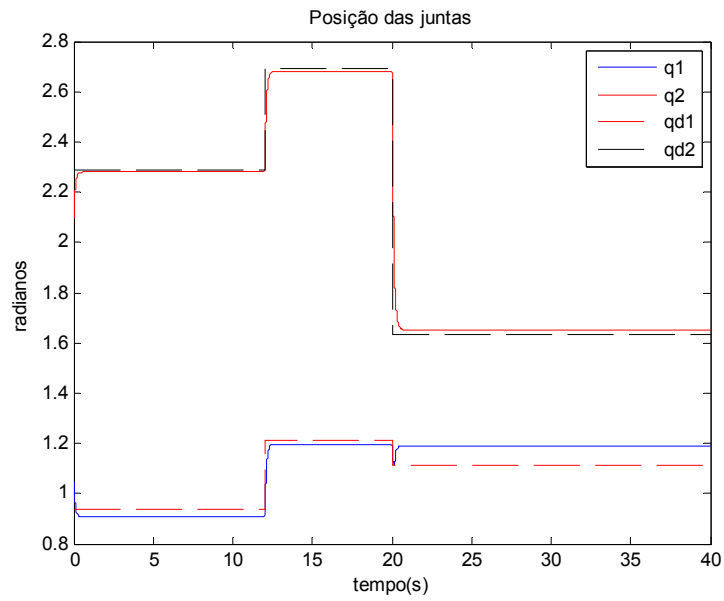


Figura 4-5

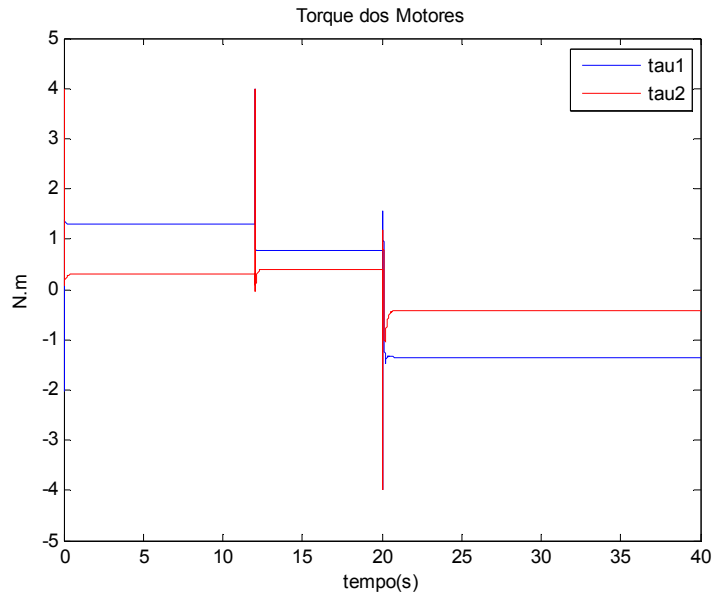


Figura 4-6

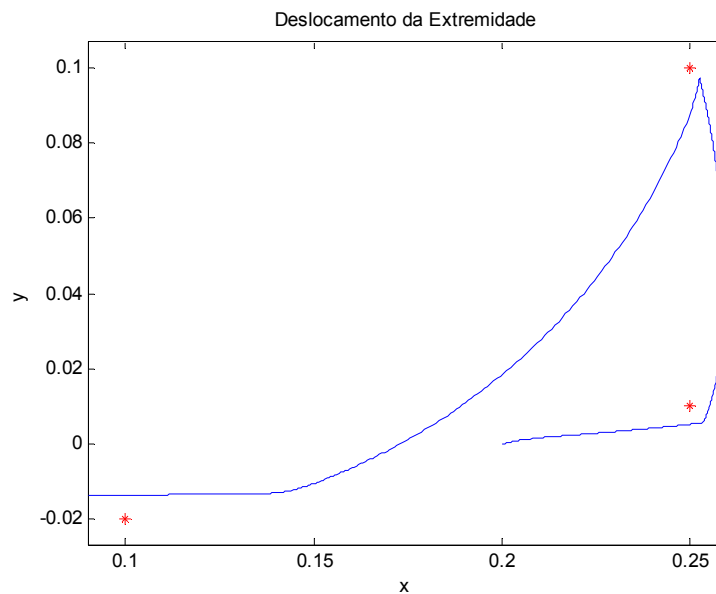


Figura 4-7

4.3.3 PID

O controle PID consegue chegar as posições desejadas ao contrario do controle anterior, mas apresenta pequeno “over-shot” para algumas posições. O tempo de amortecimento está entre 2 e 4 segundos para os pontos desejados como é visto na Figura 4-8. Bem como no controle PD o torque dos motores recebem pequena saturação quando há troca de posição objetivo como a satura;ao dura pouco tempo não há motivos para preocupação com danos nos motores.

Neste caso o torque no motor 1 é saturado quando o manipulador tenta rastrear a posição desejada abaixo do solo, vide Figura 4-9. Assim, este força o chão na direção desejada faz saturar o motor.

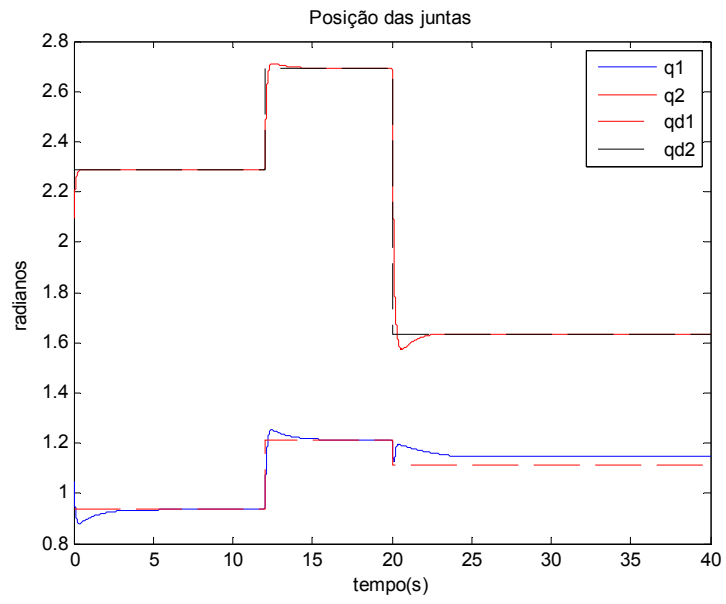


Figura 4-8

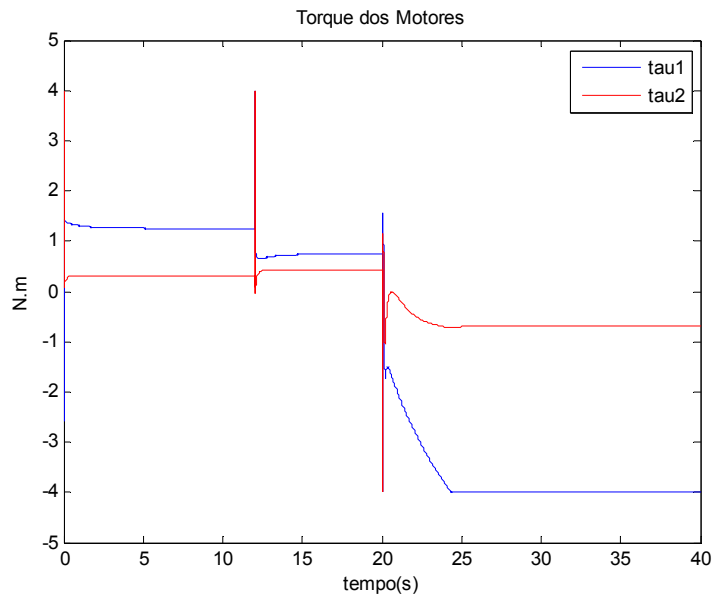


Figura 4-9

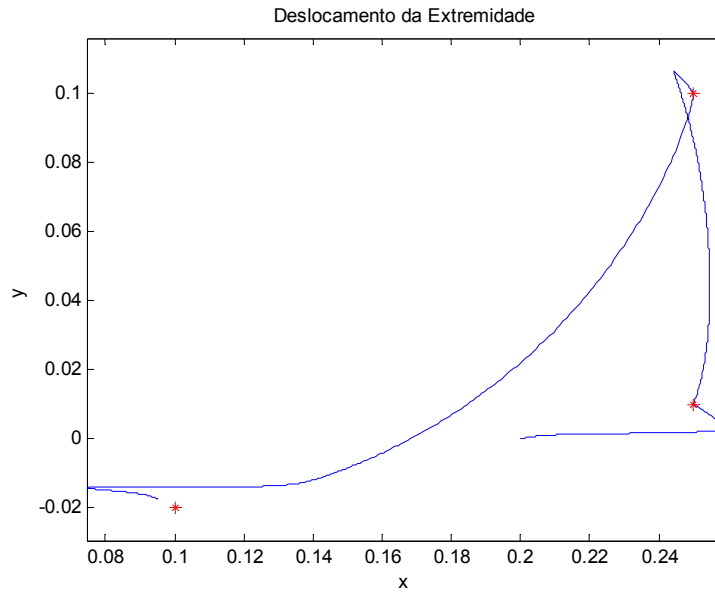


Figura 4-10

4.3.4 CTC

O controle com realimentação da dinâmica funciona muito melhor que os apresentados anteriormente na questão de tempo de resposta e amortecimento. Não foi observado “over-shot”, o tempo de amortecimento diminuiu para meio segundo em média e como o controle PID a posição desejada é atingida.

Neste caso os torques oscilam mais e passam mais tempo em saturação. É importante reparar que os torques em estado estacionário são os mesmos do controle PID, o que mostra que os motores não são forçados nem mais nem menos para manter a posição.

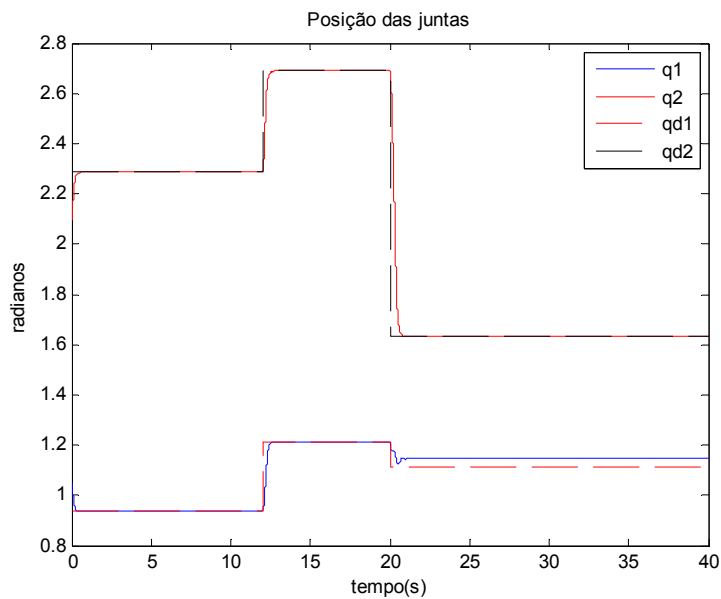


Figura 4-11

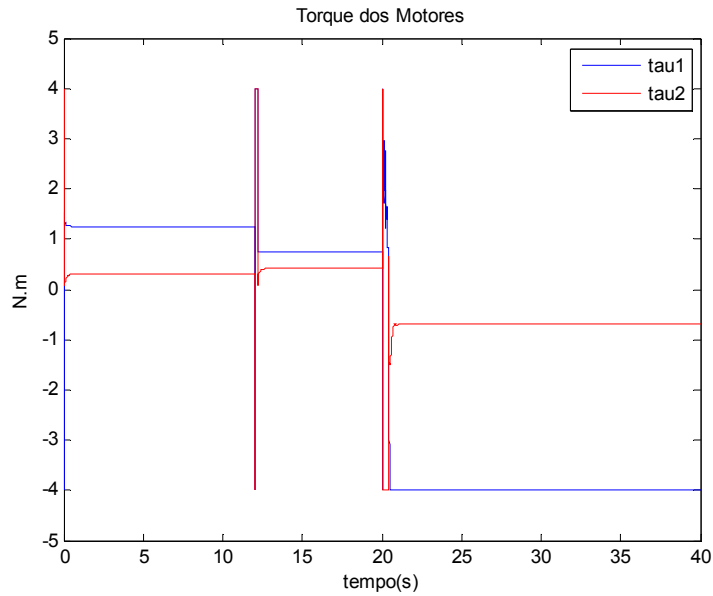


Figura 4-12

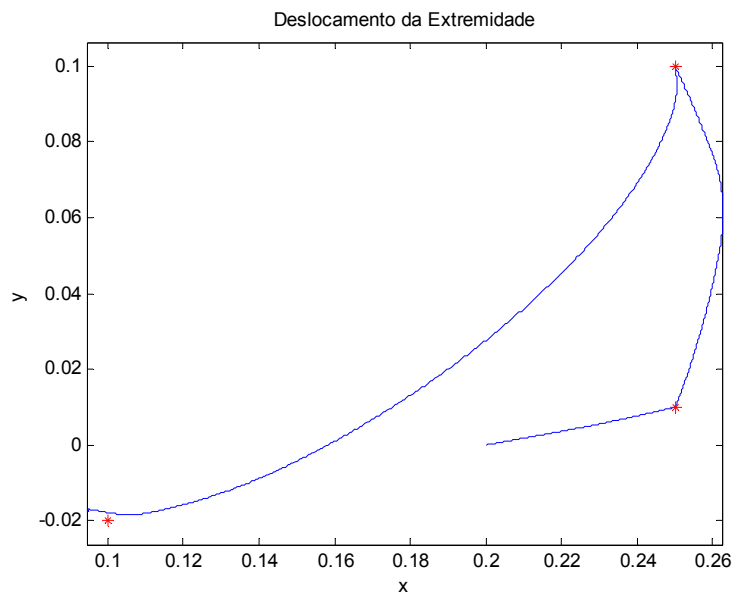


Figura 4-13

A validação destes resultados não pode ser feita, pois a implementação dos controles no manipulador não foram possíveis por falta de tempo. Um modulo de controle, onde são declarados em um arquivo de extensão “.mat” as correntes aplicadas aos motores, já está implementado bastando apenas comparar os resultados da simulação.

5 Conclusão

Com o presente trabalho foi possível ter noções de um projeto completo de um manipulador, passando pela forma conceito, projeto mecânico, montagem e simulação. Verificou-se que o projeto de um manipulador pode ter inúmeros objetivos e passar por diferentes critérios de avaliação já que muitas são as decisões a serem tomadas. Apesar de muitas delas serem decidida por conveniência, por preço, há espaço para muito trabalho de otimização e aplicação das teorias desenvolvidas na área.

A parte de montagem apresenta desafios tão animadores quanto a parte de projeto teórico, pois muitas coisas não são possíveis de ser feitas exatamente como planejado. O tempo também se mostrou um desafio uma vez que muitas coisas haviam sido planejadas, mas apenas algumas foram realizadas.

A modelagem mecânica do manipulador mostrou resultados muito interessantes sendo possível extrair relações importantes entre os elos que definiram a geometria do manipulador. A comparação deste com manipuladores robóticos comerciais sugeriu expectativa muito animadora quanto ao resultado prático, uma vez que as configurações apresentaram características parecidas.

O simulador concluiu o trabalho comprovando o entendimento da dinâmica por trás dos manipuladores onde se pode testar e comparar diversos tipos de controles estudados para o manipulador. Muitos são os controles que ainda podem ser testado no mesmo e espaço foi reservado no código para que isto fosse feito.

Trabalhos futuros poderão verificar melhor a relação entre velocidade e torque nos motores que podem chegar ao seu limite facilmente com o controle de posição. O uso de reduções é a sugestão mais imediata para resolver este problema. Também fica para trabalhos futuros a validação do simulador que já possui modulo para tal. Um projeto mecânico mais extensivo levando em conta a minimização do peso através da utilização de outros materiais também parece válido uma vez que possibilitaria maior folga aos torques dos motores.

6 Referências

- [1] Ravi Doddannavar, Andries Barnard; Practical Hydraulics Systems, 2005.
- [2] Michael J. Grimble, Michael A. Johnson; Robotics, Modelling, Planning and Control, página 191
- [3] Meggiolaro , Marco Antonio; RioBotz Combat Robot Tutorial by, página 264;
- [4] Ming Xie, Fundamentals of robotics linking perception to action, página 222
- [5] Manual RoboteQ AX2550, página 26
- [6] Paul E. Sandin; Robot Mechanism and Mechanical Devices Illustrated, 2003, página 85
- [7] Haruhiko Asada, Takeo Kanade, Ichiro Takeyama; Control of a direct drive arm
- [8] De Silva, C.W.; Control Sensors and Actuators, 2007, página 327
- [9] Spong, Hutchinson and Vidyasagar ; Robot Modelling and Control –, página 232
- [10] Measurement, Instrumentation, e Sensors Handbook CRCnetBase, página 188.
- [11] H. Asada, K. Youcef-Toumi e S.K. L M ; Joint Torque Measurement Of A Direct-Drive Arm - 1984
- [12] Autonomous Mobile Robots – sensing, control, decision making and applications.
- [13] Sadao Kawamura, Fumio Miyazaki e Suguru Arimoto; Realization of Robot Motion Based on Learning Method –IEEE Transections on Systems; 1988
- [14] Asada, H. ; Slotine, J.E. ;Robot Analysis and Control, 1986
- [15] Craig, J.J.; Introduction to Robotics, Mechanics and Control, 1989

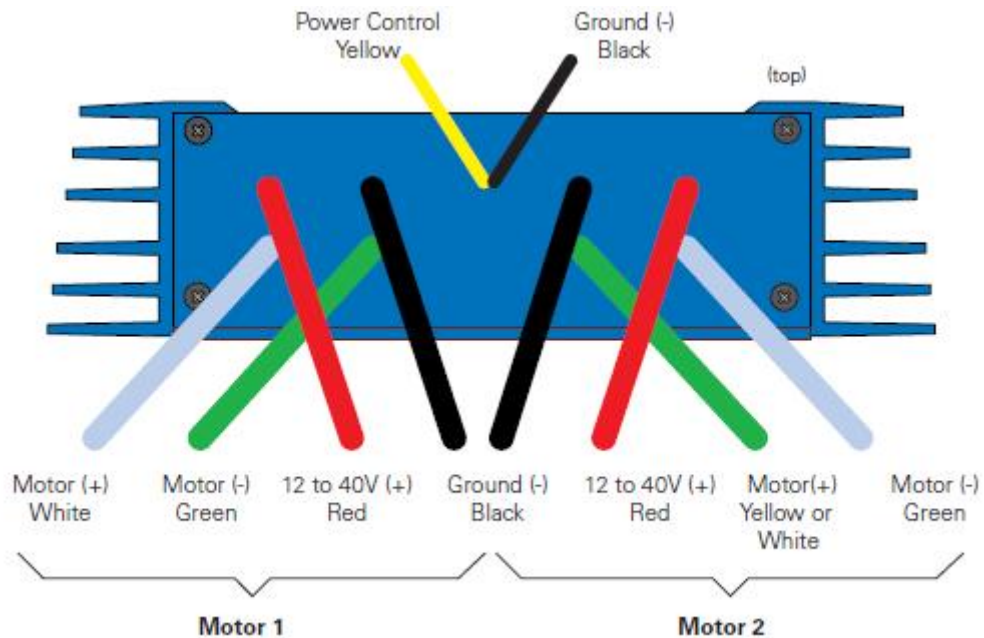
A. RoboteQ

Configurações

PARAMETRO	VALOR
Modo do sinal de controle	Analógico
Modo de controle dos motores	Canais A e B separados
Limite de Corrente	105A
Aceleração	Media
Zona Morta do controle	0%
Tipo de resposta canal um	Linear
Tipo de resposta canal dois	Linear
Ajuste esquerda/direita	Sem ajuste

Informações sobre o que significa cada um dos parâmetros podem ser obtidas em [5].

Cabeamento – Figura obtida de [5]



B. Magmotor S28-150

Voltage (V)	24
P_{output_max} (W)	2.183
Weight (kg)	1.7
Power/Weight	1.284
I_{stall} / I_{no_load}	110
K_t (N·m/A)	0.03757
K_v (RPM/V)	254
R_{motor} (Ω)	0.064
I_{no_load} (A)	3.4

Todas as informações foram obtidas de [3].

C. Propriedades de Massa – SolidWorks®

Propriedades de massa de Elo1 (Part Configuration - Valor predeterminado)

Sistema de coordenadas de saída: Sistema de coordenadas1

Densidade = 0.00 gramas por milímetro³

Massa = 232.62 gramas

Volume = 86154.24 milímetros³

Área de superfície = 32325.70 milímetros²

Centro de massa: (milímetros)

$$X = -7.75$$

$$Y = 97.13$$

$$Z = 0.00$$

Eixos principais de inércia e momentos de inércia principais: (gramas * milímetros²)

Tomado no centro da massa.

$$I_x = (0.01, 1.00, 0.00) \quad P_x = 30249.54$$

$$I_y = (-0.00, 0.00, 1.00) \quad P_y = 1281398.11$$

$$I_z = (1.00, -0.01, 0.00) \quad P_z = 1283144.87$$

Momentos de inércia: (gramas * milímetros²)

Obtido no centro de massa e alinhado com o sistema de coordenadas de saída.

$$L_{xx} = 1283080.33 \quad L_{xy} = 8991.98 \quad L_{xz} = -0.01$$

$$L_{yx} = 8991.98 \quad L_{yy} = 30314.08 \quad L_{yz} = 0.21$$

$$L_{zx} = -0.01 \quad L_{zy} = 0.21 \quad L_{zz} = 1281398.11$$

Momentos de inércia: (gramas * milímetros²)

Tomados no sistema de coordenadas de saída.

$$I_{xx} = 3477856.61 \quad I_{xy} = -166018.87 \quad I_{xz} = -0.02$$

$$I_{yx} = -166018.87 \quad I_{yy} = 44269.39 \quad I_{yz} = 0.39$$

$$I_{zx} = -0.02 \quad I_{zy} = 0.39 \quad I_{zz} = 3490129.71$$

Propriedades de massa de Elo2 (Part Configuration - Valor predeterminado)

Sistema de coordenadas de saída: Sistema de coordenadas1

Densidade = 0.00 gramas por milímetro³

Massa = 125.56 gramas

Volume = 46502.72 milímetros³

Área de superfície = 15464.73 milímetros²

Centro de massa: (milímetros)

$$X = -8.08$$

$$Y = 42.92$$

$$Z = 0.00$$

Eixos principais de inércia e momentos de inércia principais: (gramas * milímetros²)

Tomado no centro da massa.

$$I_x = (0.03, 1.00, 0.00) \quad P_x = 14982.18$$

$$I_y = (0.00, 0.00, 1.00) \quad P_y = 150907.49$$

$$I_z = (1.00, -0.03, 0.00) \quad P_z = 151702.94$$

Momentos de inércia: (gramas * milímetros²)

Obtido no centro de massa e alinhado com o sistema de coordenadas de saída.

$$L_{xx} = 151588.78 \quad L_{xy} = 3949.06 \quad L_{xz} = 0.00$$

$$L_{yx} = 3949.06 \quad L_{yy} = 15096.34 \quad L_{yz} = 0.00$$

$$L_{zx} = 0.00 \quad L_{zy} = 0.00 \quad L_{zz} = 150907.49$$

Momentos de inércia: (gramas * milímetros²)

Tomados no sistema de coordenadas de saída.

$$I_{xx} = 382920.98 \quad I_{xy} = -39611.97 \quad I_{xz} = 0.00$$

$$I_{yx} = -39611.97 \quad I_{yy} = 23299.10 \quad I_{yz} = 0.00$$

$$I_{zx} = 0.00 \quad I_{zy} = 0.00 \quad I_{zz} = 390442.45$$

Propriedades de massa de Elo3_ (Part Configuration - Valor predeterminado)

Sistema de coordenadas de saída: Sistema de coordenadas2

Densidade = 0.00 gramas por milímetro³

Massa = 206.83 gramas

Volume = 76603.77 milímetros³

Área de superfície = 34087.99 milímetros²

Centro de massa: (milímetros)

$$X = 12.70$$

$$Y = 101.76$$

$$Z = 0.00$$

Eixos principais de inércia e momentos de inércia principais: (gramas * milímetros²)

Tomado no centro da massa.

$$I_x = (0.00, 1.00, 0.00) \quad P_x = 29240.21$$

$$I_y = (0.00, 0.00, 1.00) \quad P_y = 1150540.61$$

$$I_z = (1.00, 0.00, -0.00) \quad P_z = 1152088.55$$

Momentos de inércia: (gramas * milímetros²)

Obtido no centro de massa e alinhado com o sistema de coordenadas de saída.

$$\begin{aligned}L_{xx} &= 1152088.55 & L_{xy} &= -0.45 & L_{xz} &= 0.00 \\L_{yx} &= -0.45 & L_{yy} &= 29240.21 & L_{yz} &= -0.40 \\L_{zx} &= 0.00 & L_{zy} &= -0.40 & L_{zz} &= 1150540.61\end{aligned}$$

Momentos de inércia: (gramas * milímetros²)

Tomados no sistema de coordenadas de saída.

$$\begin{aligned}I_{xx} &= 3293832.96 & I_{xy} &= 267297.02 & I_{xz} &= 0.02 \\I_{yx} &= 267297.02 & I_{yy} &= 62599.90 & I_{yz} &= -0.31 \\I_{zx} &= 0.02 & I_{zy} &= -0.31 & I_{zz} &= 3325644.71\end{aligned}$$

Propriedades de massa de Elo4 (Part Configuration - Valor predeterminado)

Sistema de coordenadas de saída: Sistema de coordenadas1

Densidade = 0.00 gramas por milímetro³

Massa = 320.53 gramas

Volume = 118714.31 milímetros³

Área de superfície = 41787.68 milímetros²

Centro de massa: (milímetros)

$$\begin{aligned}X &= -12.70 \\Y &= 51.72 \\Z &= -0.00\end{aligned}$$

Eixos principais de inércia e momentos de inércia principais: (gramas * milímetros²)

Tomado no centro da massa.

$$\begin{aligned}I_x &= (0.00, 1.00, 0.00) & P_x &= 42074.78 \\I_y &= (0.00, 0.00, 1.00) & P_y &= 2780078.43 \\I_z &= (1.00, 0.00, 0.00) & P_z &= 2781927.94\end{aligned}$$

Momentos de inércia: (gramas * milímetros²)

Obtido no centro de massa e alinhado com o sistema de coordenadas de saída.

$$\begin{aligned}L_{xx} &= 2781927.94 & L_{xy} &= 0.43 & L_{xz} &= -0.00 \\L_{yx} &= 0.43 & L_{yy} &= 42074.78 & L_{yz} &= 0.31 \\L_{zx} &= -0.00 & L_{zy} &= 0.31 & L_{zz} &= 2780078.43\end{aligned}$$

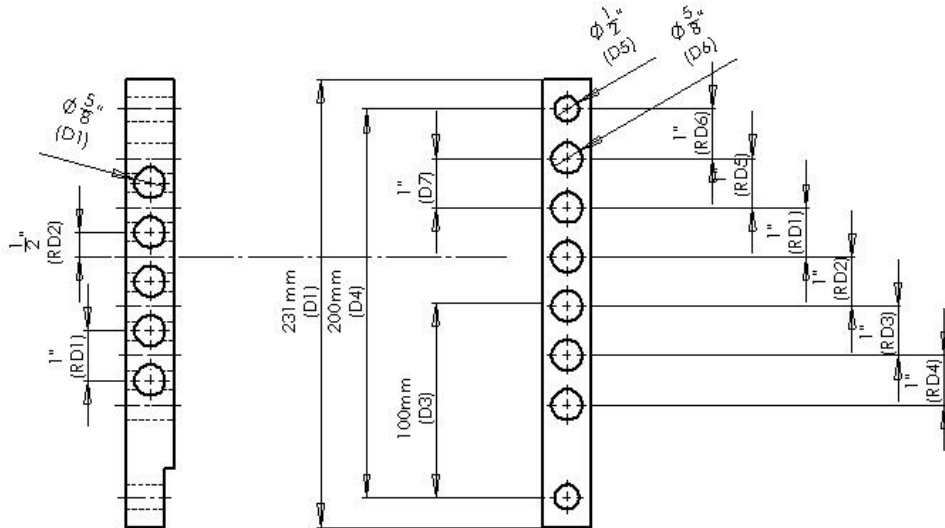
Momentos de inércia: (gramas * milímetros²)

Tomados no sistema de coordenadas de saída.

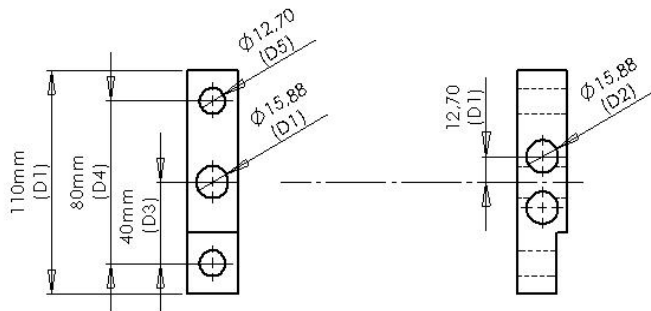
$$\begin{aligned}I_{xx} &= 3639257.77 & I_{xy} &= -210528.16 & I_{xz} &= 0.01 \\I_{yx} &= -210528.16 & I_{yy} &= 93772.84 & I_{yz} &= 0.27\end{aligned}$$

$$I_{zx} = 0.01 \quad I_{zy} = 0.27 \quad I_{zz} = 3689106.32$$

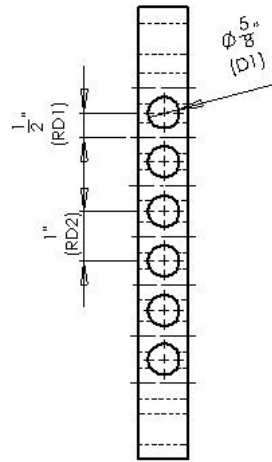
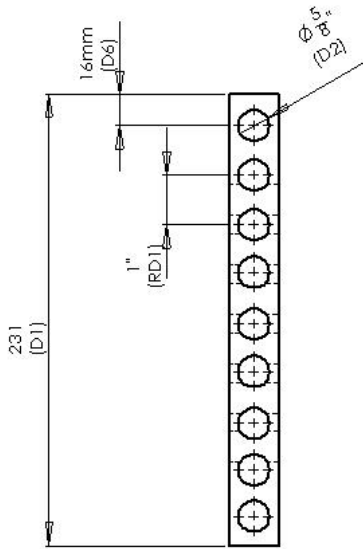
D. Planta dos Elos



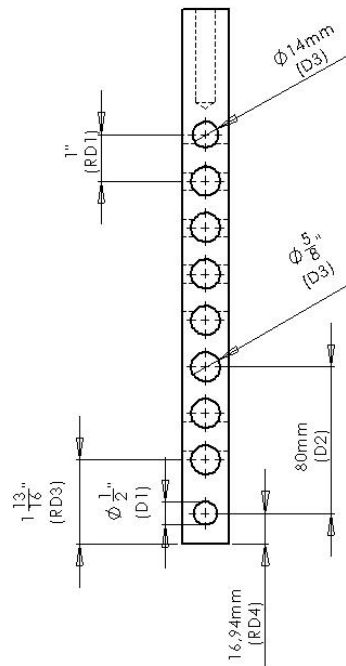
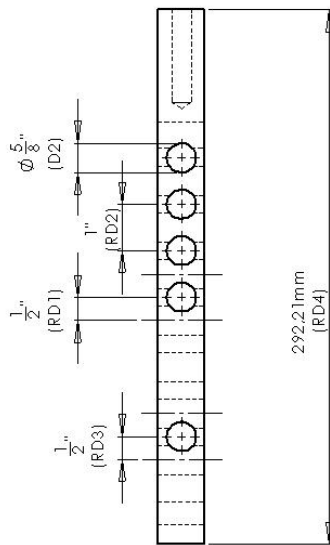
Elo1



Elo2



Elo3



Elo4

E.Código Matlab de Simulação

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sim_robo MP - 2D paralelogram manipulator simulator %
% by Marcos Paulo Faria Lima Barreto, Dez.20 2009 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
clc
figure(1)

ControlType = 7; %0-off, 1-PD, 2-PID, 3-PD+G, 4-CTC, 5-CTC with
estimated values, 6-Adaptive, 7-Robust-PD+G, 8-ga
f = 0.5; %0.2, 0.5, 1 - frequency in Hz of the trajectory
PIDIntegralTerm = 0; %initializes the integral from PID
IntegralTerm = 1; % For CTC control to have Integral term in control
asin 1 for not asing 0;
CurrentControl = 0; % If wished a Current direct control can be
applied
load 'current.mat' % file with current values in time an correspondent
time[current ctime] all line vectors
cindex=0;

%simulation parameters
tf = 40; %simulation final time
deltat = 0.001; %simulation step, in s
deltatControl = 0.005; %Control step, in s
cycleindex = floor(1/(f*deltat)); %number of integration steps in each
trajectory
plotstep = 3*0.025/deltat; %plots every plotstep calculation steps,
empirically calibrate to achieve real time animation
deltaq = 1e-6; %infinitesimal value to numerically evaluate the
Christoffel parameters
viewtype = 1; %robot animation view in 2D or 3D: 1 for plotxy, 2 for
plotxz, 3 for plotyz, 4 for plotxyz
InerciaType = 'SolidWorks'; % may be : 'SolidWorks', 'Calculated'
robotcolor = 'b'; %robot animation color
robotbasecolor = strcat(robotcolor, '^'); robotendeffectorcolor =
strcat(robotcolor, '+');
robotdesiredcolor = 'ro'; %desired end-effector position color

%initializes variables for the manipulator
n = 2; %number of joints (all vectors below must have n columns)
joint = [0 0]'; %0 for rotary joint and 1 for prismatic
q = [pi/3 2*pi/3]'; %[10 10 10]*pi/180; %initial joint positions
in radians (or m)
qdot = [0 0]*pi/180; %initial joint speeds in radians/s (or m/s)
% Motor Variables
% tausat = [10000 10000]'; %saturation torques in N.m (put very large
values to ignore them)
tausat = [4.005 4.005]'; % Saturation Torques [N*m]from motor
datasheet
Kt = 0.03757;
% Environment Variables
g = [0 -9.79 0]'; %in m/s^2 at the base frame, where g=[0 -9.81 0]'
if y0 is vertical or g=[0 0 -9.81]' if z0 is vertical
Ksolo = 8000; %[N/m]
Kdsolo = 0.5; %[N/(m/s)]
Ysolo = -0.010; %[m]

```

```

%link i is assumed as a 3D brick with dimensions lxi, lyi, lzi (along
xi,yi,zi)
l(1) = 0.200; % length of the 1st link in [mm]
l(2) = 0.080; % length of the 2nd link in [mm]
l(3) = l(1); % length of the 3rd link in [mm]
l(4) = l(1)+l(2); % length of the 4th link in [mm] - the length of
this link should be equal to l1+l2 for dexterity
lc(1) = 0.100; % 1st link mass center joint distance in [mm]
lc(2) = 0.040; % 2nd link mass center joint distance in [mm]
lc(3) = 0.100; % 3rd link mass center joint distance in [mm]
lc(4) = 0.140; % 4th link mass center joint distance in [mm]
ly = 0.0254; % link's width in one of the perpendicular directions
lz = 0.0254; % link's width in another perpendicular directions

%parameters for the dynamics
M = 2.5; %total manipulator mass, in kg
TotalLength = sum(l); %total length of the links, assuming they are
thin bars with ly = 0
m = 1./TotalLength*M; %mass of each link, assuming uniform
distribution
mest = m*0.75; %m.*(0.5+1.0*rand(n,1)) ; %estimated masses, assuming
constant or random errors
I = zeros(3,3,n); %inertia tensor of each link at the center of mass,
at the link frame
Iest = zeros(3,3,n); %estimated tensor with errors
if strcmp(InerciaType, 'Calculated')
    for i=1:n
        I(1:3,1:3,i) = m(i)/12*[[ (ly^2+lz^2) 0 0];[0 (l(i)^2+lz^2)
0];[0 0 (l(i)^2+ly^2)]];
        %Iest(1:3,1:3,i) = mest(i)/12*[[ (ly^2+lz^2) 0 0];[0
(l(i)^2+lz^2) 0];[0 0 (l(i)^2+ly^2)]];
    end
else
    % Inercias at the center of mass from SolidWorks
    I(1)=0.001281398; %[kg*m^2] %1281398[g*mm^2]
    I(2)=0.000150907; %[kg*m^2] %150907[g*mm^2]
    I(3)=0.001150540; %[kg*m^2] %1150540[g*mm^2]
    I(4)=0.002780078; %[kg*m^2] %2780078[g*mm^2]
    m(1)=0.2326; %[kg]
    m(2)=0.1256; %[kg]
    m(3)=0.2068; %[kg]
    m(4)=0.3205; %[kg]
    M=sum(m);

end

%joint friction damping matrix
Kdq = eye(n);
for i = 1:n
    Kdq(i,i) = 0; %in N.m/(rad/s) or N/(m/s)
end

%joint-space control gain matrices
Kp = eye(n);
for i = 1:n
    Kp(i,i) = tausat(i)/(10*pi/180); %proportional gain, estimated
assuming saturation if error larger than 10 degrees
end
Kd = Kp/10; %derivative gain matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SIMPLIFICATIONS FOR TESTING:

```

```

%Kp = Kp*0; %to turn off control
%Kd = Kd*0; %to turn off control
%Kdq = Kdq*0; %turn off joint friction
%Kdq = Kdq/100; %for small joint friction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ki = Kp; %integral gain matrix
Kpaux = Kp; %auxiliary matrix to store Kp for CTC
Kdaux = Kd; %auxiliary matrix to store Kd for CTC

%learning control:
Kp2 = Kp; Kd2 = Kd; %gains for the norm(qd-q)^2 term
Kp1 = 0.4*Kp; Kd1 = 0.2*Kd; %learning gains

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SIMULATION STARTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
qold = q; %values from the last iteration step, needed by the Newmark-
Beta method
qdotold = qdot;

% Newmark-Beta method coefficients
% beta = Newmark constant (1/6 or 1/4 usually, or zero for the
explicit version)
% gamma = Newmark constant (1/2)
beta = 1/4;
gamma = 1/2;
nb0 = 1/(beta*deltat*deltat);
nb1 = gamma/(beta*deltat);
nb2 = 1/(beta*deltat);
nb3 = 1/(beta*2) - 1;
nb4 = gamma/beta - 1;
nb5 = 0.5*deltat*(gamma/beta - 2);
nb6 = deltat*(1 - gamma);
nb7 = deltat*gamma;

%tables to store the results to be plotted later:
tvector = [0];
qvector = q';
qdotvector = qdot';
qIntegral=zeros(n,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%main loop:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for tindex=1:(tf/deltat)
    t = tindex*deltat;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Desired trajectory
    %this value should only be updated in the frequency of the control
system
    %to reflect the actual control input, and not the simulation
frequency;
    %this has not been implemented yet in this simulation

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %f is the frequency in Hz, defined in the beginning

```

```

    xcentertraj = [0.150 0.150]'; %coordinate of the center of the
desired circular trajectory
    rtraj = 0.100; %radius of the desired circular trajectory
    xd = xcentertraj + rtraj*[cos(2*pi*f*t) sin(2*pi*f*t)]'; %desired
end-effector position
    nd = [1 1]; td = [-1 1]; %desired directions of the n and t unit
vectors of the end-effector
    % nd = nd/norm(nd); td = td/norm(td); bd = cross(nd,td); bd =
bd/norm(bd);
    %set the pose vector
    pd(1:n,1) = [0.250 0.010]; %xd;%desired end-effector position
    if t>12
        pd(1:n,1) = [0.250 0.100];
    end
    if t>20
        pd(1:n,1) = [0.100 -0.02]; %[0.100 0.200];
    end

    %Inverse Kinematics for the desired values
    if tindex==1 %it's the first iteration, qd hasn't been calculated
yet
        qd = calcIK(pd,1)';%calcIK(pd, 1); %finds only the solution
close to the actual q in the first iteration
        qdold = qd; %the old is the same as the first value, to avoid
large qdotd values
        qdoldold = qd; %the value before qold, to estimate the desired
accelerations
    else
        qdoldold = qdold;
        qdold = qd; %to compute qdotd
        qd = calcIK(pd,1)';%calcIK(pd, 1); %finds only the solution
close to the actual q in the first iteration
    end
    qdotd = (qd - qdold)/deltat; %desired joint speed (in rad/s or m/s)
    qdotdold = (qd - 2*qdold + qdoldold)/deltat/deltat;

    % Current Position
    p = [l(1)*cos(q(1))-(l(4)-l(2))*cos(q(2));...
        l(1)*sin(q(1))-(l(4)-l(2))*sin(q(2))];

    %Jacobian and Gravity Centers Jacobian calculations
    J = zeros(2,n);
    Jc1 = zeros(2,n);
    Jc2 = zeros(2,n);
    Jc3 = zeros(2,n);
    Jc4 = zeros(2,n);
    J = [-l(1)*sin(q(1)) (l(4)-l(2))*sin(q(2)) ; l(1)*cos(q(1)) -
(l(4)-l(2))*cos(q(2))]; %end-effector Jacobian, including only linear
not angular parts
    Jc1 = [-lc(1)*sin(q(1)) 0 ;...
        lc(1)*cos(q(1)) 0];
    Jc2 = [0 -lc(2)*sin(q(2));...
        0 lc(2)*cos(q(2))];
    Jc3 = [-lc(3)*sin(q(1)) -l(2)*sin(q(2)) ;...
        lc(3)*cos(q(1)) l(2)*cos(q(2))];
    Jc4 = [-l(1)*sin(q(1)) lc(4)*sin(q(2)) ;...
        l(1)*cos(q(1)) -lc(4)*cos(q(2))];

    % End-Effector Speed

```

```

V = J*qdot;

%Inertia Matrix calculation
H = zeros(n,n);
Hest = zeros(n,n); %estimated

H(1,1) = I(1)+m(1)*lc(1)^2+ I(3) + m(3)*lc(3)^2 +m(4)*l(1)^2;
H(2,2) = I(2)+m(2)*lc(2)^2+ I(4) + m(4)*lc(4)^2 +m(3)*l(2)^2;

H(1,2) = (m(3)*l(2)*lc(3)-m(4)*l(1)*lc(4))*cos(q(1)-q(2));

%Centrifugal and Coriolis terms
Hdq = zeros(n,n);
dHdq = zeros(n,n,n); %dHdq(i,j,k) is defined as dHij/dqk
hijk = zeros(n,n,n); %hijk(i,j,k) is Christoffel's hijk
coefficient
dHdq(1,2,1) = (m(4)*l(1)*lc(4)-m(3)*l(2)*lc(3))*sin(q(1)-q(2));
dHdq(2,1,1) = (m(4)*l(1)*lc(4)-m(3)*l(2)*lc(3))*sin(q(1)-q(2));
dHdq(1,2,2) = (m(3)*l(2)*lc(3)-m(4)*l(1)*lc(4))*sin(q(1)-q(2));
dHdq(2,1,2) = (m(3)*l(2)*lc(3)-m(4)*l(1)*lc(4))*sin(q(1)-q(2));
for i=1:n
    for j=1:n
        for k=1:n
            hijk(i,j,k) = dHdq(i,j,k) - 0.5*dHdq(j,k,i);
        end
    end
end
end
h = zeros(n,1);
for j = 1:n
    for k = 1:n
        h(1:n,1) = h(1:n,1) + hijk(1:n,j,k) * qdot(j) * qdot(k);
    end
end
end

%Gravity terms

G = - (Jc1'*m(1)+Jc2'*m(2)+Jc3'*m(3)+Jc4'*m(4))*g;
Gest = - (Jc1'*mest(1)+Jc2'*mest(2)+Jc3'*mest(3)+Jc4'*mest(4))*g;

%joint friction damping torque is -Kdq*qdot, it is already
included in the damping matrix Kdq

%external forces and their effect at each joint
tauext = zeros(n,1); %this will be the resulting torque
(generalized force) at each joint (generalized coordinate)
%external forces (besides gravity) and moments acting at the
center of mass of each link (e.g. wind drag, water current)
Fc = zeros(2,n); %here you can enter external forces acting at
each link center of mass
Mc = zeros(2,n); %here you can enter external moments acting at
each link center of mass

tauext = (Jc1'*m(1)+Jc2'*m(2)+Jc3'*m(3)+Jc4'*m(4))*g;
%Fc(1:3,i) is an external force acting at the center of mass of link i
tauext = (Jc1'*m(1)+Jc2'*m(2)+Jc3'*m(3)+Jc4'*m(4))*g;
%Mc(1:3,i) is an external moment acting at the center of mass of link
i

```

```

% Floor impact
if p(2)<Ysolo
    Fsolo = [-Kdsolo*(V(1)) ; Ksolo*(Ysolo-p(2))-Kdsolo*(V(2))]; %
Considering Spring and Draging Reactions
    tauext = tauext + J'*Fsolo;
end
%external forces and moments acting at the end-effector (e.g.
payload weight, environment contact forces, etc.)
Fe = zeros(2,1); %here you can enter external forces acting at the
end-effector
Me = 0; %here you can enter external moments acting at the end-
effector
tauext = tauext + J'*Fe;
tauext(2) = tauext(2) + Me;%JA(1:3,1:n) '*Me;

%Iw antes das col. = Iw Depois & Col. Inerl;

%Checar momentos não gene.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Control laws
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if rem(t,deltatControl)
    switch ControlType
        case {0} %no control
            tau = 0*q;
            tauaux = 0*q;
            Kp = zeros(n); Kd = zeros(n); %tauaux is calculated
below in the Newmark-Beta algorithm from Kp and Kd, so they must be
set to zero

            case {1,2,3} %joint-space PD, PID or PD+G control law
                %the law is tau = Kp*(qd-q) + Kd*(qdotd-qdot) =
(Kp*qd+Kd*qdotd) - (Kp*q+Kd*qdot) = tau - tauaux
                %the I or G term can be added to tau
                tau = Kp*qd + Kd*qdotd;
                tauaux = Kp*q + Kd*qdot; %tauaux can only have Kp
and Kd, because of the Newmark-Beta algorithm
                if (ControlType == 2) %PID
                    PIDIntegralTerm = PIDIntegralTerm + (qd-q)*deltat;
                    tau = tau + Ki*PIDIntegralTerm;
                elseif (ControlType == 3) %PD+G
                    tau = tau + G; %G or Gest (mest, Iest, Hest and
Gest are the estimated values, with errors)
                end

            case {4} %Computer Torque Control with exact parameters
                %the PD law is u = Kpaux*(qd-q) + Kdaux*(qdotd-qdot) =
(Kpaux*qd+Kdaux*qdotd) - (Kpaux*q+Kdaux*qdot)
                %the CTC law is tau = H*u + h + G =
H*(Kpaux*qd+Kdaux*qdotd)+h+G - H*(Kpaux*q+Kdaux*qdot) = tau - tauaux
                Kp = H*Kpaux; %Kp considering the decoupling due to H
                Kd = H*Kdaux; %Kd considering the decoupling due to H
                %to normalize results:
                auxdet = det(H); Kp = Kp/auxdet; Kd = Kd/auxdet;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

        tau_d = H*qdotd_d + Kp*q_d + Kd*qdot_d + h + G; %already
includes the effect of H
        tauaux = Kp*q + Kd*qdot; %already includes the effect
of H; tauaux can only have Kp and Kd, because of the Newmark-Beta
algorithm

        case {5} %Computer Torque Control with estimated
parameters
            Kp = Hest*Kp_aux; %Kp considering the decoupling due to
Hest
            Kd = Hest*Kd_aux; %Kd considering the decoupling due to
Hest

            %to normalize results:
            auxdet = det(Hest); Kp = Kp/auxdet; Kd = Kd/auxdet;
%%%%%%%%%%%%%%
            tau_d = Hest*qdotd_d + Kp*q_d + Kd*qdot_d + h + Gest;
%already includes the effect of Hest
            if IntegralTerm
                PIDIntegralTerm = PIDIntegralTerm + (q_d - q)*deltat;
                tau_d = tau_d + Ki*PIDIntegralTerm;
            end
            tauaux = Kp*q + Kd*qdot; %already includes the effect
of Hest; tauaux can only have Kp and Kd, because of the Newmark-Beta
algorithm

        case {6} %Adaptive control
            %not implemented yet, the Regression Matrix must be
deduced on a
            %note: tauaux can only have Kp and Kd, because of the
Newmark-Beta algorithm
            %desativa;áo dos controles para inibir erro
            tau_d = 0*q;
            tauaux = 0*q;
            Kp = zeros(n); Kd = zeros(n); %tauaux is calculated
below in the Newmark-Beta algorithm from Kp and Kd, so they must be
set to zero

        case {7} %Robust+PD+G
            tau_d = Kp*q_d + Kd*qdot_d + G;
            tauaux = Kp*q + Kd*qdot; %tauaux can only have Kp and
Kd, because of the Newmark-Beta algorithm
            filterederror = (qdot_d - qdot) + 1*(q_d - q); %in general,
filterederror = (qdot_d - qdot) + Lambda*(q_d - q), where Lambda is positive
definite

            epsilon = 0.1; %a small value
            taumax = mean(tausat)/10; %a value larger than all
dynamic effects, impossible to predict
            if filterederror > epsilon
                tau_d = tau_d +
filterederror/norm(filterederror)*taumax;
            else
                tau_d = tau_d + filterederror/epsilon*taumax;
            end

        case {8}
            %Genetic Algorithms
            %not implemented yet
            tau_d = 0*q;
            tauaux = 0*q;

```



```

        Kp = zeros(n); Kd = zeros(n); %tauaux is calculated
below in the Newmark-Beta algorithm from Kp and Kd, so they must be
set to zero
    end
end

tau = taud - tauaux;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%saturation:
for i=1:n
    if tau(i)>tausat(i)
        tau(i) = tausat(i);
    elseif tau(i)<-tausat(i)
        tau(i) = -tausat(i);
    end
end
taud = tau + tauaux; %corrects taud if necessary due to
saturation

%equation is tau + tauext = taud - tauaux + tauext = taud -
(Kp*q+Kd*qdot) + tauext = H*qdotdot + Kdq*qdot + h + G
%which is rewritten as tau = taud + tauext - h - G = H*qdotdot +
(Kd+Kdq)*qdot + Kp*q
tau = taud + tauext - h - G; %sum of torques and dynamic effects

%Current direct control - suppress control from simulation
if CurrentControl
    if time>ctime(cindex)
        cindex = cindex+1;
    end
    tau = current(:,cindex)*kt;
end

% if (and only if) this is the first step, initialize qdotdotold
and qdotdotvector, and the vectors of the desired values
if tindex==1
    qdotdotold = inv(H)*(tau - (Kd+Kdq)*qdot - Kp*q); %joint
accelerations
    qdotdotvector = qdotdotold'; %initialized the acceleration
vector
    qdvector = qd'; %initialized the desired position vector
    qdotdvector = qdotd'; %initialized the desired speed vector
    tauvector = tau'; %initialized the actuator torque vector
    pvector = p'; %initialized actual end-effector pose
    pdvector = pd'; %initialized desired end-effector pose
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NEWMARK-BETA integration method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Keff = nb0*H + nb1*(Kd+Kdq) + Kp; %Effective stiffness matrix
Kinv = inv(Keff);
q = Kinv*(tau + H*(nb0*qold+nb2*qdotold+nb3*qdotdotold) +
(Kd+Kdq)*(nb1*qold+nb4*qdotold+nb5*qdotdotold)); %new value for q
qdotdot = nb0*(q-qold) - nb2*qdotold - nb3*qdotdotold; %new value
for qdotdot
qdot = qdotold + nb6*qdotdotold + nb7*qdotdot; %new value for qdot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%updates old = current for the Newmark-Beta method
qold = q;
qdotold = qdot;
qdotdotold = qdotdot;

%stores a table of the time, position, speed, acceleration and
joint torque
tvector = [tvector; t];
qvector = [qvector; q'];
qdotvector = [qdotvector; qdot'];
qdotdotvector = [qdotdotvector; qdotdot'];
qdvvector = [qdvvector; qd']; %desired position vector
qdotdvvector = [qdotdvvector; qdotd']; %desired speed vector
tauvector = [tauvector; tau']; %joint torque vector
pvector = [pvector; p']; %actual end-effector pose
pdvector = [pdvector; pd']; %desired end-effector pose

if mod(tindex,plotstep)==0 %plots every plotstep calculation
steps
%       x = [r(1,:)];
%       y = [r(2,:)];
%       z = [r(3,:)];
hold off
switch viewtype
case 1 %xy plane
PlotManipulator(q,1,Ysolo)
%plot(0,0,robotbasecolor)
hold on
%axis([-L,L,-L,L])
%       axis([0,2*TotalLenght/3,-
TotalLenght/3,2.2*TotalLenght/3])
%       axis equal
%       %plot(x,y,robotcolor);
%       %plot(re(1),re(2),robotendeffectorcolor);
%       plot(pd(1),pd(2),robotdesiredcolor);
case 2 %xz plane
plot(0,0,robotbasecolor)
hold on
axis([-TotalLenght,TotalLenght,-
TotalLenght,TotalLenght])
axis equal
plot(x,z,robotcolor);
plot(re(1),re(3),robotendeffectorcolor);
plot(pd(1),pd(3),robotdesiredcolor);
case 3 %yz plane
plot(0,0,robotbasecolor)
hold on
axis([-TotalLenght,TotalLenght,-
TotalLenght,TotalLenght])
axis equal
plot(y,z,robotcolor);
plot(re(2),re(3),robotendeffectorcolor);
plot(pd(2),pd(3),robotdesiredcolor);
case 4 %xyz view
plot3(0,0,0,robotbasecolor)
hold on
axis([-TotalLenght,TotalLenght,-
TotalLenght,TotalLenght,-TotalLenght,TotalLenght])
axis equal

```

```

        plot3(x,y,z,robotcolor);
        plot3(re(1),re(2),re(3),robotendeffectorcolor);
        plot3(pd(1),pd(2),pd(3),robotdesiredcolor);
    end
    title(strcat('t=',num2str(t)))
    pause(deltat)
end

end %end of main loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%plots graphs at the end of simulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
colorcode = 'brkmcg';
figure
string_matrix = cell(2*n,1);
for i=1:n %plots actual position vector
    plot(tvector,qvector(:,i),colorcode(mod(i-1,6)+1))
    hold on
    string_matrix(i,1) = {strcat('q', num2str(i))};
end
for i=1:n %plots desired position vector
    plot(tvector,qdvector(:,i),strcat(colorcode(mod(i,6)+1),'--'))
    string_matrix(n+i,1) = {strcat('qd', num2str(i))};
end
%English
% legend(string_matrix)
% title('joint positions')
% xlabel('time(s)')
% ylabel('radians or meters')
%Portuguese
legend(string_matrix)
title('Posição das juntas')
xlabel('tempo(s)')
ylabel('radianos')

figure
plot(pvector(:,1),pvector(:,2),robotcolor)
hold on
axis equal
plot(pdvector(:,1),pdvector(:,2),'r*')
title('Deslocamento da Extremidade');
xlabel('x'); ylabel('y');

figure
string_matrix = cell(n,1);
for i=1:n %plots actuator torque vector
    plot(tvector,tauvector(:,i),colorcode(mod(i-1,6)+1))
    hold on
    string_matrix(i,1) = {strcat('tau', num2str(i))};
end
%English
% legend(string_matrix)
% title('joint torques (generalized forces)')
% xlabel('time(s)')
% ylabel('N.m or N')
%Portuguese
legend(string_matrix)
title('Torque dos Motores')
xlabel('tempo(s)')

```

```
ylabel('N.m')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%,%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% calcIK - calculates the numerical Inverse Kinematics of a 2D  
% hibrid robotic manipulator %  
% by Marcos Paulo Faria Lima Barreto, Jul.16 2009  
%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function q = calcIK(ptarget, l)  
term1 = 2*atan2(ptarget(2),ptarget(1))+pi;  
term2 = acos(-(ptarget(1)^2+ptarget(2)^2)/(2*l(1)^2)+1);  
q = [(term1-term2)/2 (term1+term2)/2];  
%se colidir não da para chegar  
end
```

```
function Xe = calcDK(q,l)  
Xe= [l(1)*cos(q(1))-(l(4)-l(2))*cos(q(2)); l(1)*sin(q(1))-(l(4)-  
l(2))*sin(q(2))];  
end
```