

Universidade de Brasília
Departamento de Engenharia Elétrica



Modelagem e Identificação do Robô
UR3

Trabalho de Graduação 1

Aluno: Pedro Saman D. N. Cesarino

Matrícula: 15/0144890

Email: pedrosaman96@gmail.com

Orientador: Geovany Araujo Borges

Brasília
10 de Dezembro de 2019

Sumário

| | | |
|----------|--------------------------------------|-----------|
| 1 | Manipuladores Robóticos | 1 |
| 2 | Modelagem | 2 |
| 2.1 | Modelagem Cinemática | 2 |
| 2.2 | Modelagem Dinâmica | 4 |
| 3 | Identificação | 8 |
| 3.1 | <i>Simulated Annealing</i> | 10 |
| 4 | Resultados e Discussão | 12 |
| 5 | Conclusão e Trabalhos Futuros | 16 |

1 Manipuladores Robóticos

Manipuladores robóticos são amplamente utilizados em fábricas com as mais diversas atividades como pintura, soldagem, transporte e montagem. Porém para que seja possível a realização de tais atividades com precisão e segurança é necessário o projeto de controladores e para que isso seja feito com maior grau de confiabilidade é imprescindível a modelagem dinâmica do robô. Existem algumas formas de se modelar a relação entre torque e velocidade, ou aceleração de um robô, entre elas existe a modelagem Newton-Euler que utiliza as relações de movimento de corpos rígidos e as leis de Newton.

Porém, como vai ser visto na subseção 2.2, tal modelo depende de parâmetros que nem sempre estão totalmente disponíveis como a massa, momento de inércia, centro de massa e atrito das juntas. Por esse motivo é preciso realizar a identificação dos parâmetros desconhecidos, ou então parâmetros que não se tem confiança no valor disponível.



Figura 1: Foto do robô UR3 utilizado no trabalho.

Manipuladores podem ser definidos como um conjunto de links conectados por juntas que permitem movimento relativo entre eles, tais juntas podem ser rotativas ou lineares. Normalmente o número de graus de liberdade de um

robô é igual ao número de juntas que ele possui. Ao final do último link de um robô costuma-se ter alguma ferramenta chamada de efetuador terminal, tal ferramenta pode ser uma tocha de soldagem, uma furadeira ou então uma garra.

No caso deste trabalho foi utilizado o manipulador robótico UR3, imagem 1, disponível no Laboratório de Robótica e Automação (LARA). Este robô possui seis graus de liberdade rotativos e na última junta foi instalada uma garra como efetuador terminal. Por ser um produto comercializado, a fabricante não disponibiliza a modelagem cinemática ou dinâmica ou sequer todos os parâmetros do robô como a matriz de inércia.

Esse robô possui um computador que embarca toda sua parte de potência e controle já possuindo implementado de fábrica um controlador de velocidade, portanto ao usuário é possível apenas escolher velocidades e acelerações para cada junta não sendo possível injetar diretamente corrente elétrica. Para programar o robô é necessário passar comandos para ele via uma linguagem de *script* desenvolvida pelo fabricante sendo possível fazer diversas leituras de variáveis como de corrente elétrica, temperatura, posição, velocidade [ur3, 2019].

2 Modelagem

A modelagem de robôs é essencial para que seja possível realizar tarefas complicadas tanto em aplicações industriais quanto aplicações domésticas. Com a modelagem dinâmica é possível desenvolver controladores complexos de precisão como controle de impedância ou força. Uma correta modelagem permite também a concepção de robôs humanoides capazes de realizar atividades parecidas com as realizadas por nós humanos.

2.1 Modelagem Cinemática

A cinemática é a ciência que estuda o movimento de corpos, ou seja sua posição, velocidade e aceleração, sem se preocupar com as forças necessárias para gera-lo. Normalmente é de interesse determinar a cinemática inversa de um manipulador, ou seja quais ângulos ou posições as juntas precisam estar para que o efetuador do robô alcance um certo ponto do espaço, por outro lado a cinemática direta determina o ponto atual a partir de cada ângulo de junta.

Para se realizar esses modelos cinemáticos é necessário definir a matriz de Denavit-Hartenberg (DH) que é determinada a partir de quatro parâmetros que descrevem a relação entre os links de um robô. Dois desses parâmetros

descrevem o link em si e dois explicam como cada junta se conecta. Uma característica interessante desse procedimento é que três desses parâmetros são fixos para dado robô e o último depende apenas do ângulo de rotação que uma junta realizou, então apenas com esse valor é possível descrever a cinemática de um manipulador. A explicação detalhada de como nomear os frames de cada link e como conseguir determinar os parâmetros de DH pode ser vista em [Craig, 2009].

| i | a_i | α_i | d_i | θ_i |
|-----|-------|------------|-------|------------|
| 1 | 0 | $-\pi/2$ | 0,152 | q_1 |
| 2 | 0,244 | 0 | 0 | q_2 |
| 3 | 0,213 | 0 | 0 | q_3 |
| 4 | 0 | $-\pi/2$ | 0,112 | q_4 |
| 5 | 0 | $\pi/2$ | 0,085 | q_5 |
| 6 | 0 | 0 | 0,082 | q_6 |

Tabela 1: Tabela com os Parâmetros de Denavit-Hartenberg do UR3

A *toolbox* de robótica do *Matlab* foi usada para verificar a matriz de DH, validar o modelo cinemático direto, validar o modelo dinâmico proposto posteriormente e para se obter numericamente a cinemática inversa. Uma imagem do robô com todos ângulos iguais a zero pode ser vista na figura 2.

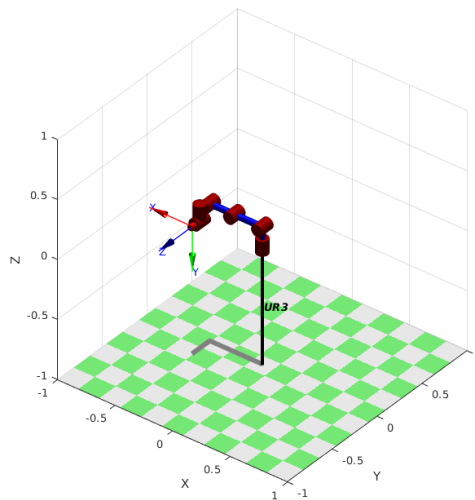


Figura 2: Representação do UR3 usando a Toolbox de robótica do Matlab.

2.2 Modelagem Dinâmica

A modelagem dinâmica é necessária para que seja possível se realizar de sistemas dinâmicos, portanto nessa seção será apresentada a modelagem de Newton-Euler que foi seguida para o trabalho utilizando a convenção de variáveis apresentada da equação 1 até 15 em que o índice i vai de zero até cinco para representar cada uma das 6 juntas do robô.

$$a_{c,i} = \text{aceleração linear do centro de massa do link } i \quad (1)$$

$$a_{e,i} = \text{aceleração linear do fim do link } i, \text{ começo do link } i+1 \quad (2)$$

$$\omega_i = \text{velocidade angular do frame } i \text{ com respeito ao frame } i \quad (3)$$

$$\alpha_i = \text{aceleração angular do frame } i \text{ com respeito ao frame } i \quad (4)$$

$$z_i = \text{eixo de rotação do frame } i \text{ com relação ao frame } i \quad (5)$$

$$g_i = \text{aceleração da gravidade com relação ao frame } i \quad (6)$$

$$f_i = \text{força exercida pelo link } i \quad (7)$$

$$\tau_i = \text{torque exercido pelo link } i \quad (8)$$

$$R_{i+1}^i = \text{rotação do sistema de coordenadas } i+1 \text{ para o } i \quad (9)$$

$$m_i = \text{massa do link } i \quad (10)$$

$$I_i = \text{matriz de inercia referente ao link } i \quad (11)$$

$$r_{i,i+1} = \text{distancia entre o frame } i \text{ e o } i+1 \quad (12)$$

$$r_{i,ci} = \text{distancia entre o frame } i \text{ e o centro de massa da junta } i \quad (13)$$

$$r_{i+1,ci} = \text{distancia entre o frame } i+1 \text{ e o centro de massa da junta } i \quad (14)$$

$$B_i = \text{Coeficiente de atrito viscoso da junta } i \quad (15)$$

Antes de falar da modelagem dinâmica, é necessário definir o que são matrizes de rotação entre sistemas de eixos de coordenadas. Considere um vetor $v(x, y, z)$ qualquer que foi definido inicialmente em um sistema $x_0y_0z_0$ e precisa ser representado em um segundo sistema $x_1y_1z_1$ que rotacionou em relação ao primeiro: γ graus em x , β graus em y e α em z . É possível realizar uma transformação de rotação entre os sistemas de eixos via a matriz de rotação mostrada em 16 em que c significa cosseno e s significa seno. Portanto temos: $v_1 = R_0^1 v_0$. Se houve-se um terceiro eixo de coordenadas que rotacionou em relação ao segundo via transformação R_1^2 , poderíamos escrever v como: $v(x_1, y_1, z_1) = R_1^2 v_1 = R_1^2 R_0^1 v_0$.

$$R_0^1 = \begin{bmatrix} c(\alpha) & -s(\alpha) & 0 \\ s(\alpha) & c(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(\beta) & 0 & s(\beta) \\ 0 & 1 & 0 \\ -s(\beta) & 0 & c(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\gamma) & -s(\gamma) \\ 0 & s(\gamma) & c(\gamma) \end{bmatrix} \quad (16)$$

A parametrização de Newton-Euler parte do princípio duas leis de movimento de corpos rígidos de Euler e da lei da ação e reação de Newton:

- Ação e reação: Toda ação tem uma reação de mesma magnitude e direção oposta. Ou seja, se o link 1 exerce um torque τ no link 2, este exerce um torque $-\tau$ no link 1.
- A variação do momento linear é igual a força resultante em link.
- A variação do momento angular é igual ao torque resultante em um link.

$$\sum f = m\dot{v} \quad (17)$$

$$\sum \tau_0 = \frac{d(I_0\omega_0)}{dt} \quad (18)$$

As equações 17 e 18 podem ser reescritas como em 19 e 20, considerando que a massa do link é constante, fazendo a transformação do torque e momento de inercia em relação ao frame inercial para o frame fixo no centro de massa do próprio link [Truesdell, 1992].

$$\sum f = ma \quad (19)$$

$$\sum \tau = \omega \times (I\omega) + I\dot{\omega} \quad (20)$$

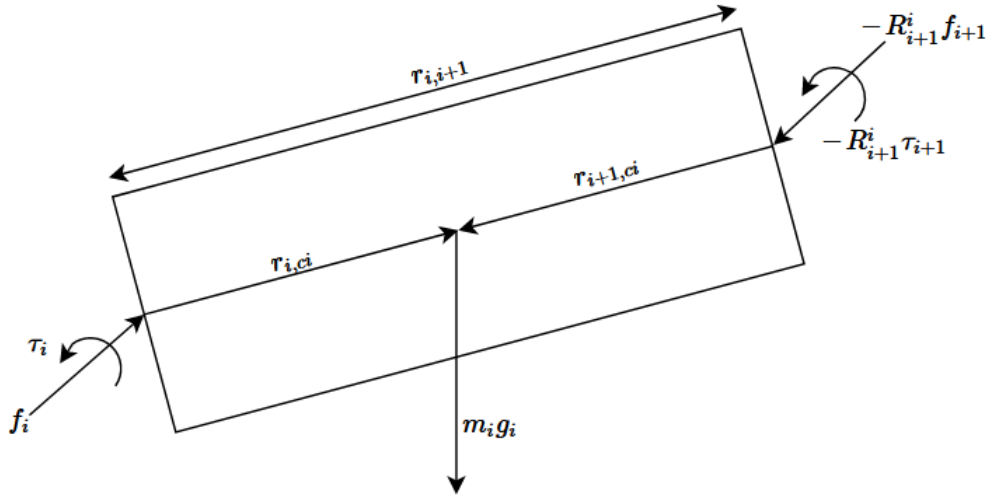


Figura 3: Forças e torques aplicados em um link i qualquer.

As forças que atuam em um link i qualquer são: A força do próprio link f_i , a força de reação do link posterior f_{i+1} e a força da gravidade $m_i g$. Então, considerando que a aceleração do link aparece no centro de massa dele, a expressão para a força exercida pelo link atual, partindo da equação 19 fica como em 22.

$$f_i - R_{i+1}^i f_{i+1} + m_i g = m_i a_{c,i} \quad (21)$$

$$f_i = R_{i+1}^i f_{i+1} + m_i a_{c,i} - m_i g + R_{i+1}^i f_{i+1} \quad (22)$$

Para calcular o torque exercido pelo link atual τ_i via o balanço de torques, equação 24, partindo da equação 20, temos que levar em consideração: o torque de reação do link posterior τ_{i+1} , o torque gerado pela força f_i vezes o a distância do início do link atual para o seu centro de massa e o torque resultante da força de reação do link posterior vezes a distância do link posterior até o centro de massa do link atual.

$$\tau_i - R_{i+1}^i \tau_{i+1} + f_i \times r_{i,ci} - R_{i+1}^i f_{i+1} \times r_{i+1,ci} = \omega \times (I\omega) + I\dot{\omega} \quad (23)$$

$$\tau_i = R_{i+1}^i \tau_{i+1} - f_i \times r_{i,ci} + (R_{i+1}^i f_{i+1}) \times r_{i+1,ci} + \omega \times (I\omega) + I\dot{\omega} \quad (24)$$

Algo que não é considerado na equação 24 é o torque de reação contrário ao movimento resultante do atrito presente nas conexões mecânicas entre o motor e o resto do braço. Uma forma simples de se incorporar o atrito no movimento é considerar apenas o atrito viscoso e que ele é proporcional a velocidade angular da junta [Bajd et al., 2010]. Adicionado esse componente com sinal oposto ao movimento na equação 24 obtemos a equação 25 para torque final.

$$\tau_i = R_{i+1}^i \tau_{i+1} - f_i \times r_{i,ci} + (R_{i+1}^i f_{i+1}) \times r_{i+1,ci} + \omega \times (I\omega) + I\dot{\omega} + B_i \omega_i. \quad (25)$$

O objetivo final é calcular os torques aplicados em cada junta partindo dos conhecidos torques e forças externas aplicados no último link do robô mais as leituras de posição, velocidade e aceleração providos pelos sensores em cada junta. Para isso é preciso expressar as componentes ω , α e $a_{c,i}$, da equação 25, em termos de q , \dot{q} e \ddot{q} .

Para ω partimos do fato que a velocidade angular de um link é igual a velocidade angular do frame anterior, rotacionado para o eixo de coordenadas do frame atual, mais a rotação feita pela junta atual, equação 26. Usando

a segunda lei de Newton em um eixo rotativo [Taylor, 2005] a derivada da expressão 26 fica como mostrado na equação 27.

$$\omega_i = R_{i-1}^i \omega_{i-1} + z_i \dot{q}_i \quad (26)$$

$$\alpha = R_{i-1}^i \alpha_{i-1} + z_i \ddot{q}_i + \omega_i \times z_i \dot{q}_i \quad (27)$$

Em fim, para $a_{c,i}$, partimos da expressão 28 da velocidade linear do centro de massa e considerando que a distância entre o link anterior e o centro de massa do link atual é constante temos a equação 30 para a aceleração.

$$v_{c,i} = v_{e,i-1} + \omega_i \times r_{i-1,ci} \quad (28)$$

$$a_{e,i} = R_{i-1}^i a_{e,i-1} + \dot{\omega} \times r_{i-1,i} + \omega \times (\omega \times r_{i-1,ci}) \quad (29)$$

$$a_{c,i} = R_{i-1}^i a_{e,i-1} + \dot{\omega} \times r_{i,ci} + \omega \times (\omega \times r_{i,ci}) \quad (30)$$

Com isso finalizamos a formulação proposta que relaciona a posição, velocidade e aceleração de cada junta e o torque exercido para realizar aquele movimento. Como pode ser visto essa formulação é recursiva, ou seja, para um manipulador com n juntas, o calculo do torque da junta 0 depende do torque das $n - 1$ juntas posteriores, por outro lado o calculo da última junta depende da definição da aceleração dos centros de massa de todos links anteriores. Dessa forma o calculo pode ser dividido assim:

- **Recursão Direta:**

Considerando as condições iniciais $\omega_0 = \alpha = a_{c,0} = a_{e,0} = 0$, isto significa dizer que o manipulador tem movimento nulo em relação a um referencial, é possível resolver as equações 26, 27, 29 e 30, nesta ordem, partindo do link 0 até o link $n - 1$.

- **Recursão Inversa:**

Considerando as condições iniciais $f_n = 0$ e $\tau_n = 0$, ou seja não há forças ou torques externos sendo aplicados no efetuador terminal, é possível resolver as equações 22 e 25 para obter o torque que esta sendo realizado em cada uma das n juntas.

Um algoritmo que implementa todos os passos das recursões direta e inversa descritos acima foi implementado no ambiente *Matlab* utilizando como base e inspiração o trabalho do [Høifødt, 2011]. Tal algoritmo foi escrito com variáveis de entrada as variáveis de junta e como saída fornece os torques realizados nas juntas.

Como é possível ver pelas equações a realização do calculo depende, além das variáveis de junta, da definição das seguintes constantes para cada uma das n juntas: distancia entre a origem do frame da junta e o seu centro de massa nas 3 direções, matriz de inércia 3×3 que possui 6 valores distintos, massa e coeficiente de atrito viscoso. Portanto o conhecimento dessas onze constates de cada junta se torna necessária. Foram consideradas conhecidas as distâncias entre os frames, equação 12, fornecidos pelo fabricante igual a tabela ??, os valores estão em metros.

| i | 0 | 1 | 2 | 3 | 4 | 5 |
|--------------|----------|----------|----------|----------|----------|----------|
| $rx_{i,i+1}$ | 0 | 0.244 | 0.213 | 0 | 0 | 0 |
| $ry_{i,i+1}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $rz_{i,ci}$ | 0.152 | 0 | 0 | 0.112 | 0.085 | 0.082 |

Tabela 2: Distancias nos três eixos entre frames adjacentes.

3 Identificação

Devido ao grau de complexidade do modelo levantado e a grande não linearidade presente nas transformações de rotação entre as velocidades e os torques não foi possível utilizar as técnicas tradicionais de identificação como os mínimos quadrados ou o filtro de kalman estendido. Foi necessário então utilizar alguma técnica que permitisse o uso de modelos não lineares que não envolvesse calculo da matriz jacobina já que esta requiriria o cálculo analítico de um número muito grande de derivadas parciais.

Como foi dito na seção 1, o robô permite apenas que o usuário mande ao robô referências de velocidade e de aceleração desejadas tendo internamente um controlador que garanta tais referências.

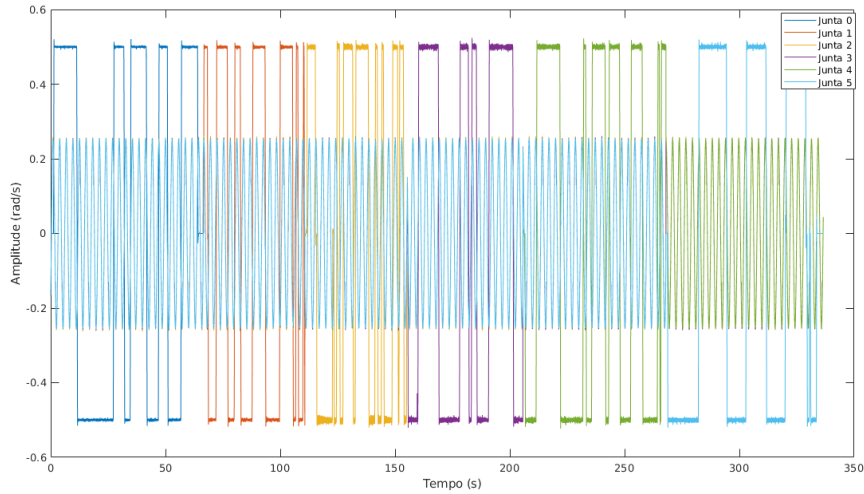
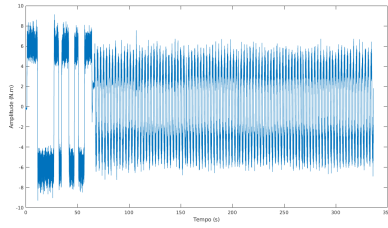


Figura 4: Sinal de entrada do da coleta de dados.

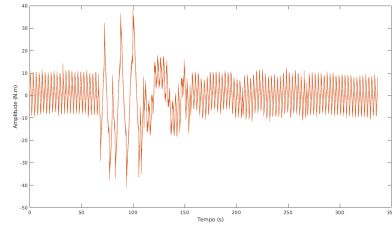
Foi proposto um sinal de entrada de velocidade com valor de regime permanente igual a $0,5 \text{ rad/s}$ e aceleração de $7,5 \text{ rad/s}^2$ em que uma junta por vez faria um movimento pseudo-aleatório com níveis mais ou menos $0,5 \text{ rad/s}$ enquanto as outras juntas todas iriam fazer um movimento senoidal. O sinal senoidal foi escolhido para tirar o braço da mesma pose durante o movimento de regime da junta que realizava o movimento PRBS.

Tal sinal foi proposto em detrimento de um sinal PRBS completo pois ele é menos agressivo ao robô, ou seja o PRBS poderia danificar o robô tanto por gerar muitos trancos nas juntas tanto porque se usássemos uma banda do sinal maior para evitar tantas transições o braço iria se chocar com a mesa ou ultrapassar o limite mecânico de rotação.

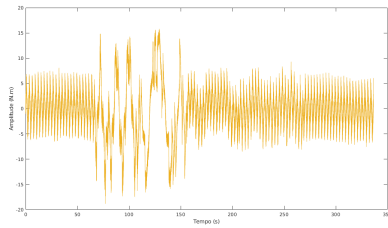
Na figura 4 podemos ver o sinal mandado para todas as 6 juntas sendo possível ver que cada uma realizou o PRBS isoladamente. Enquanto o sinal de torque lido para a entrada proposta pode ser visto nas figuras em 5.



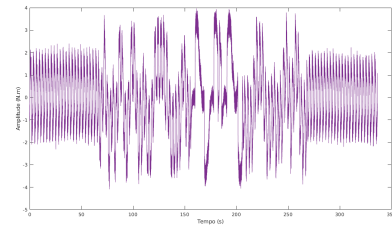
(a) Junta 0.



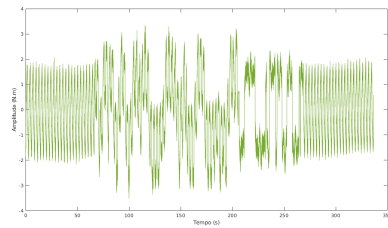
(b) Junta 1.



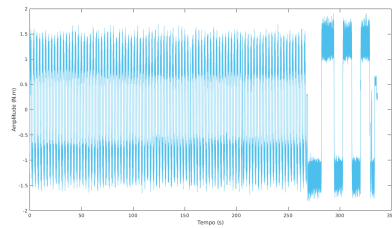
(c) Junta 2.



(d) Junta 3.



(e) Junta 4.



(f) Junta 5.

Figura 5: Torque realizado em cada junta durante o experimento.

3.1 *Simulated Annealing*

O algoritmo de otimização de funções *Simulated Annealing* é utilizado em situações que o espaço de busca é muito grande especialmente quando o mínimo global está escondido em volta de diversos mínimos locais ruins e, como o número de possibilidades é grande, se torna impossível testar todas combinações até exaustão [Flannery et al., 1992].

A motivação física desse algoritmo se baseia no recozimento de metais em que uma liga metálica é liquefeita e depois resfriada lentamente para que seja formado um cristal sólido. Esse cristal, se não houver imperfeições, é

o estado de menor energia do material em que cada átomo está alinhado perfeitamente e a uma distância grande de todos outros átomos vizinhos. Porém se o resfriamento não for feito lentamente não será possível se chegar em tal estado cristalino e será necessário refazer o procedimento.

De forma análoga o *simulated annealing* escolhe um conjunto novo de parâmetros vizinhos a solução atual e via uma função de avaliação verifica se a nova solução é melhor, se sim a antiga solução é substituída, porém mesmo se for pior existe uma probabilidade da solução também ser aceita dependendo do valor atual da temperatura T , por analogia ao processo térmico, sendo que quanto maior o valor de T maior a componente aleatória sendo somada a solução atual no próximo passo.

Usualmente a temperatura começa com um valor alto e decai com o número de iterações já tomadas. O processo finaliza quando a temperatura já esta muito baixa e as soluções novas tem avaliação igual a anterior indicando um mínimo local. Se por acaso a solução convergir para um mínimo local pior que o inicial ou a taxa de rejeição for alta ocorre o *reannealing*, em que a temperatura sobe de novo para tentar buscar outros mínimos.

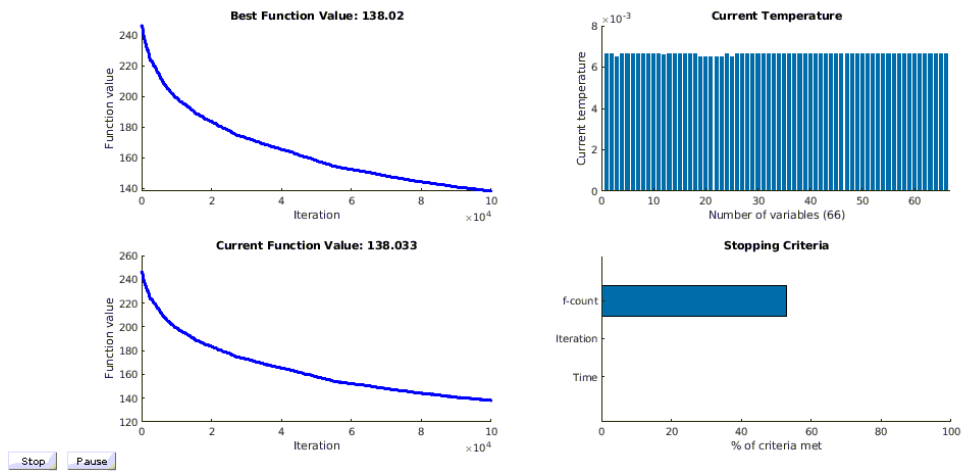


Figura 6: Tela de andamento do *Simulated Annealing*.

Neste trabalho foi utilizado o algoritmo de *simulated annealing* implementado na *toolbox* de otimização de funções do *Matlab*. Tomando como solução inicial uma estimativa grosseira dos 66 parâmetros desconhecidos do robô e como função a ser otimizada o erro quadrático médio entre o valor lido de torque e o estimado utilizando os parâmetros da solução atual na equação 25. Porém foi constatado que desta forma as juntas com magnitude maior de torque tinha um peso muito maior do que as outras juntas e por isso foi

feita uma normalização no valor da função de otimização com base no valor quadrático médio de cada torque.

Os parâmetros do algoritmo em si foram: Temperatura inicial = 10 e Temperatura atual = Temperatura inicial / iteração atual, ou seja o decaimento da temperatura começa com velocidade maior no começo e diminui com o tempo. Na figura 6 podemos ver a tela de monitoração do andamento do algoritmo durante os testes realizados. Os valores em *Best Function Value* e *Current Function Value* são a média aritmética do erro quadrático médio do torque individual dividido pela média quadrática do sinal.

4 Resultados e Discussão

Após uma semana passada do algoritmo rodando sem convergir um resultado preliminar dos parâmetros foi coletado apesar de que como é possível ver na figura 6 o resultado ainda estava diminuindo. Um novo experimento foi feito realizando movimentos aleatórios no robô via *joystick* tentando passar por vários pontos diferentes da área de trabalho para validar a qualidade da identificação.

Nas figuras 7 a 12 é possível ver a comparação dos torques medidos e estimados para o novo experimento, já na tabela 4 podemos ver os parâmetros estimados, todos no SI, e, por fim, na tabela 3 podemos ver o *fitness* em porcentagem de cada junta seguindo a fórmula 31. Lembrando que essa função tem valor máximo de 100% caso os sinais sejam idênticos e mínimo de menos infinito caso sejam completamente descorrelacionados.

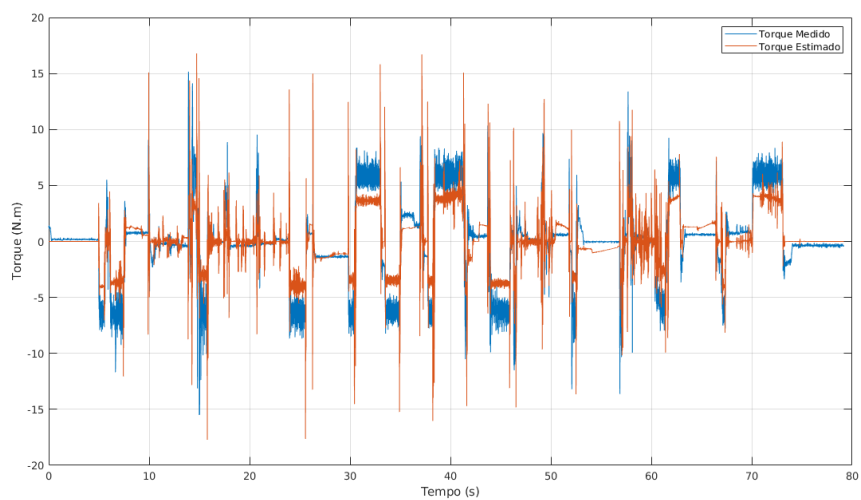


Figura 7: Comparação entre o Torque medido e estimado para a junta 0.

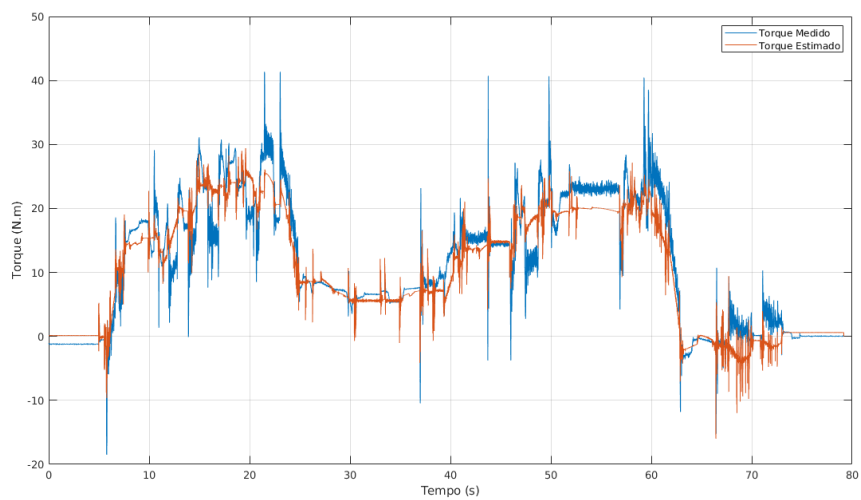


Figura 8: Comparação entre o Torque medido e estimado para a junta 1.

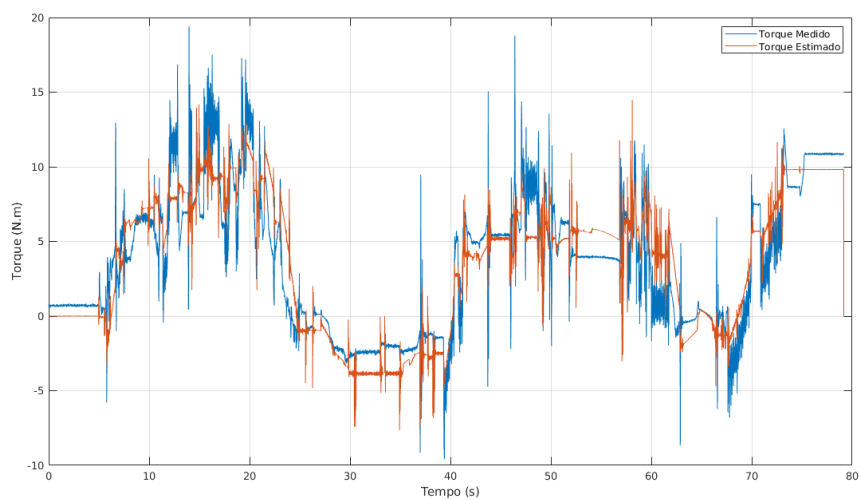


Figura 9: Comparação entre o Torque medido e estimado para a junta 2.

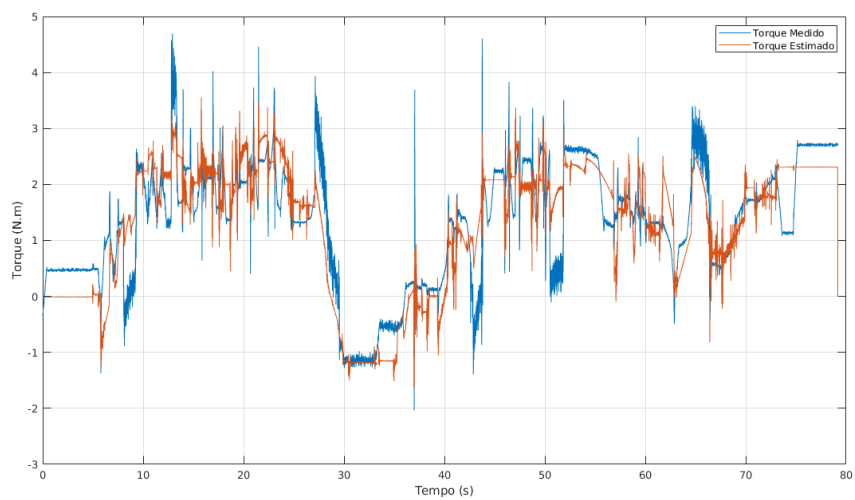


Figura 10: Comparação entre o Torque medido e estimado para a junta 3.

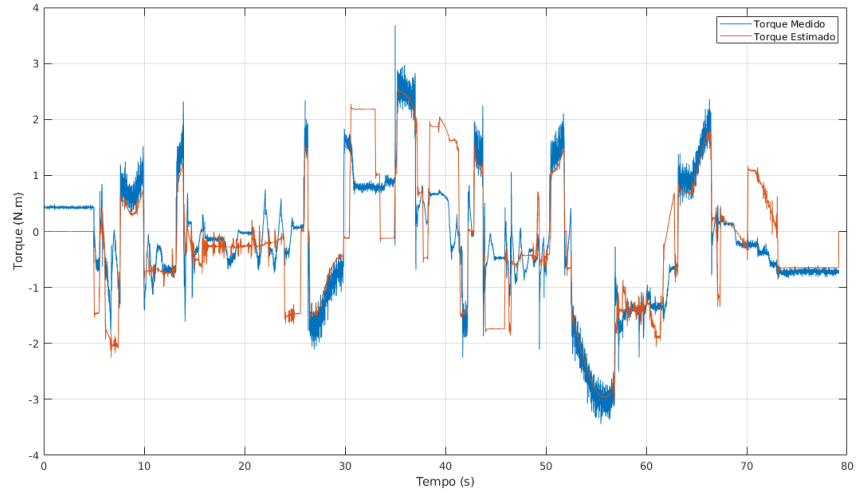


Figura 11: Comparação entre o Torque medido e estimado para a junta 4.

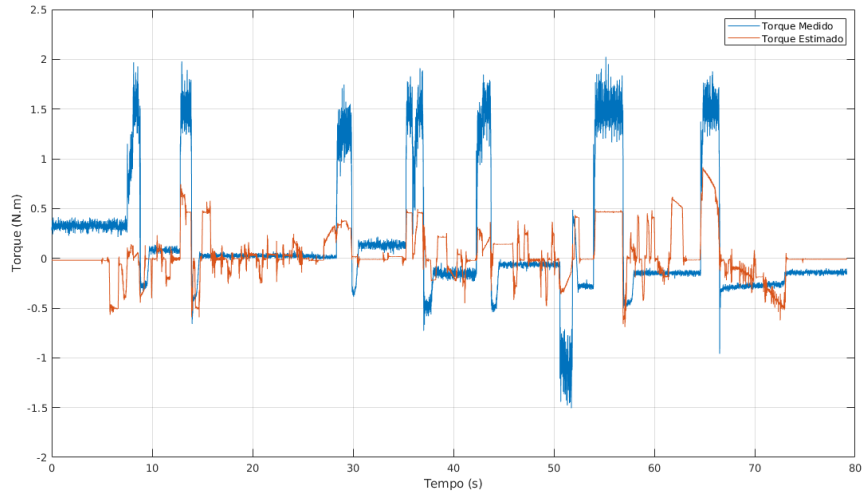


Figura 12: Comparação entre o Torque medido e estimado para a junta 5.

$$Fit = 100 \frac{(1 - norm(torque - torquemedido))}{norm(torque - mean(torque))} \quad (31)$$

| | Junta 0 | Junta 1 | Junta 2 | Junta 3 | Junta 4 | Junta 5 |
|----------------|---------|---------|---------|---------|---------|---------|
| <i>Fitness</i> | -0.95% | 60.60% | 51.70% | 47.56% | 42.81% | -98.71% |

Tabela 3: *Fitness* de cada junta utilizando o experimento de validação.

| | Junta 0 | Junta 1 | Junta 2 | Junta 3 | Junta 4 | Junta 5 |
|-------------|----------------|----------------|----------------|----------------|----------------|----------------|
| m_i | 3.166 | 4.696 | 1.814 | 0.003 | 1.811 | 2.327 |
| $rx_{i,ci}$ | -0.056 | -0.003 | 0.0129 | -0.157 | 0 | 0 |
| $ry_{i,ci}$ | 0.176 | 0.231 | 0 | 0.020 | -0.006 | 0 |
| $rz_{i,ci}$ | 0.497 | -0.357 | 0.465 | -1.159 | -0.006 | 0.068 |
| Ixx_i | 0.012 | 0.149 | -0.805 | -0.040 | 0.150 | 0.048 |
| Ixy_i | 0.359 | 0.302 | -0.209 | 0.225 | -0.019 | 0.005 |
| Ixz_i | 0.090 | 0.486 | 0.066 | -0.010 | -0.018 | -0.011 |
| Iyy_i | -0.014 | 0.447 | -1.364 | -0.390 | 1.060 | 0.212 |
| Iyz_i | 0.067 | -0.444 | -0.282 | 0.044 | -0.025 | 0.002 |
| Izz_i | -1.086 | 0.003 | -0.178 | -1.094 | -0.209 | 0.007 |
| B_i | 5.256 | 1.745 | 2.875 | 0.921 | 2.940 | 0.948 |

Tabela 4: 66 parâmetros identificados.

5 Conclusão e Trabalhos Futuros

Os resultados obtidos de *fitting* não foram satisfatórios, apesar do algoritmo da identificação ainda não ter terminado sendo os resultados apresentados aqui parciais e a figura 6 indicar que o erro ainda está em uma tendência de queda.

É possível ver que ao redor do valor de torque zero, ou seja a possível zona morta, o resultado é especialmente ruim e uma melhora no modelo poderia ser tentar incluir esse aspecto do robô à modelagem colocando uma restrição no torque gerado se ele for pequeno em torno do zero ou então com técnicas de bifurcação transcística [Aguirre, 2004].

Os valores observados na tabela 4 estão de acordo com valores fisicamente possíveis, ou seja não foram encontrados valores de distância fora da realidade, massas ou coeficientes de atrito negativos. Isso indica que o algoritmo de modelagem está condizente com a realidade e o algoritmo de identificação está explorando uma área de busca que faz sentido.

A partir do momento que o modelo dinâmico do robô está identificado é possível começar a pensar em design de controladores que não sejam puramente cinemáticos como os controladores de impedância, complacentes ou de força. Inclusive sendo possível fazer estratégias de controle que incorporem a colaboração com outros manipuladores ou então robôs móveis.

Então é proposto para a segunda etapa do trabalho de graduação a melhoria do modelo tentando incorporar fenômenos que não foram considerados neste trabalho e, se o resultado da identificação chegarem em pontos aceitáveis, o começo do desenvolvimento de técnicas de controle para o manipulador levando consideração a possibilidade de cooperação com robôs móveis.

Referências

- [ur3, 2019] (2019). *Universal Robots e-Series User Manual*. Universal Robotics, 5.5 edition.
- [Aguirre, 2004] Aguirre, L. A. (2004). *Introdução à identificação de sistemas—Técnicas lineares e não-lineares aplicadas a sistemas reais*. Editora UFMG.
- [Bajd et al., 2010] Bajd, T., Mihelj, M., Lenarčič, J., Stanovnik, A., and Munih, M. (2010). *Robotics*, volume 43. Springer Science & Business Media.
- [Craig, 2009] Craig, J. J. (2009). *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India.
- [Flannery et al., 1992] Flannery, B. P., Press, W. H., Teukolsky, S. A., and Vetterling, W. (1992). Numerical recipes in c. *Press Syndicate of the University of Cambridge, New York*, 24:78.
- [Høifødt, 2011] Høifødt, H. (2011). Dynamic modeling and simulation of robot manipulators: The newton-euler formulation. Master’s thesis, Institutt for teknisk kybernetikk.
- [Taylor, 2005] Taylor, J. R. (2005). *Classical mechanics*. University Science Books.
- [Truesdell, 1992] Truesdell, C. A. (1992). *A first course in rational continuum mechanics*, volume 1. Academic Press.