

TRABALHO DE GRADUAÇÃO

ESTIMAÇÃO DE FORÇA DE INTERAÇÃO  
NO BRAÇO ROBÓTICO UR3

Pedro Garcia

Brasília, dezembro de 2019



ENGENHARIA  
MECATRÔNICA  
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO  
**ESTIMAÇÃO DE FORÇA DE INTERAÇÃO  
NO BRAÇO ROBÓTICO UR3**

**Pedro Garcia**

*Relatório submetido como requisito parcial de obtenção  
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Geovany Araújo Borges, ENE/UnB

*Orientador*

\_\_\_\_\_

Prof. Henrique Cezar Ferreira, ENE/UnB

*Examinador interno*

\_\_\_\_\_

Prof. João Yoshiyuki Ishihara, ENE/UnB

*Examinador interno*

\_\_\_\_\_

**Brasília, dezembro de 2019**

## FICHA CATALOGRÁFICA

PEDRO, GARCIA

Estimação de força de interação no braço robótico UR3,

[Distrito Federal] 2019.

xiv, 51p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2019). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Robótica cooperativa

2. Controle complacente

3. Estimação de estados

I. Mecatrônica/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

GARCIA, PEDRO. (2019). Estimação de força de interação no braço robótico UR3. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-*n*°017, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 65p.

## CESSÃO DE DIREITOS

AUTOR: Pedro Garcia

TÍTULO DO TRABALHO DE GRADUAÇÃO: Estimação de força de interação no braço robótico UR3.

GRAU: Engenheiro

ANO: 2019

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

---

Pedro Garcia

Campus Darcy Ribeiro, SG-11, Universidade de Brasília.

70919-970 Brasília – DF – Brasil.

## **Dedicatória**

*Dedico este trabalho a todos que fizeram parte da minha jornada até o presente momento,  
e àqueles que encontrarei nos meus caminhos futuros graças à minha formação.*

*Pedro Garcia*

## Agradecimentos

*Há tantos que mereceriam estar aqui que não vou arriscar começar a citar nomes e acabar esquecendo alguém. Sendo assim, começo agradecendo a Deus por tudo. Depois agradeço à minha família, pela minha educação, pelo suporte, pelo amor. Em seguida agradeço aos meus amigos, tanto os antigos quanto os novos, pelas memórias felizes, pelas discussões profundas, pelas raivas que passamos juntos. Agradeço também aos meus professores, por terem me mostrado um mundo que eu não conhecia, especialmente àqueles que contribuíram diretamente nesses últimos metros da maratona que foi esse curso. Sou grato também à UnB, à todos que trabalham para fazê-la funcionar. É fácil criticar e ver defeitos, mas ao parar para pensar dá pra encontrar uma série de coisas que a universidade proporciona que vão me marcar para sempre. Obrigado!*

*Pedro Garcia*

---

## RESUMO

Este trabalho consiste na estimação do esforço de interação no efetuador terminal do robô manipulador UR3. Para tanto, a solução proposta foi o uso de um estimador de estados e do modelo dinâmico do robô para obter a informação de força. O modelo dinâmico foi obtido através da derivação de Lagrange e seus parâmetros dinâmicos foram identificados através de um experimento com uma trajetória de excitação persistente. O estimador utilizado foi o Filtro de Kalman Unscented na forma de Raiz Quadrada. Usando essa técnica, foi possível estimar a força aplicada sobre o efetuador terminal do UR3. Esse resultado mostra que a solução proposta é viável para resolver o problema, embora requeira ajustes para fornecer uma estimativa mais acurada e precisa do esforço de interação.

Palavras Chave: estimação de força, robô manipulador, ur3, modelagem dinâmica, identificação de parâmetros dinâmicos, filtro de kalman unscented

---

## ABSTRACT

This work consists of the estimation of the force on the end effector of the robot manipulator UR3. The proposed solution was the use of a state estimator and the dynamic model of the robot in order to obtain the force information. The dynamic model was derived using the Lagrange formulation and the dynamic parameters were identified through and experiment with an persistently exciting trajectory. The estimator chosen was the Unscented Kalman Filter in its Square Root form. Using this technique, it was possible to estimate the force applied on the end effector of the UR3. This result show that the proposed solution is viable to solve the problem, although it requires tuning in order to provide a more accurate and precise estimate of the interaction force.

Keywords: force estimation, robot manipulator, ur3, dynamic modeling, dynamic parameter identification, unscented kalman filter

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	MOTIVAÇÃO	1
1.2	ESTIMAÇÃO DE FORÇA	2
1.3	OBJETIVO	3
1.4	RESULTADOS OBTIDOS	3
1.5	APRESENTAÇÃO DO MANUSCRITO	3
<b>2</b>	<b>Fundamentos de Modelagem dinâmica e Estimação de estados</b>	<b>4</b>
2.1	CINEMÁTICA DIRETA	4
2.1.1	TRANSFORMAÇÕES HOMOGÊNEAS	4
2.1.2	CONVEÇÃO DE DENAVIT-HARTENBERG	5
2.2	CINEMÁTICA DIFERENCIAL	6
2.2.1	JACOBIANO	7
2.3	MODELO DINÂMICO	8
2.3.1	EQUAÇÃO DE EULER-LAGRANGE	9
2.3.2	ENERGIA CINÉTICA DE UM ROBÔ MANIPULADOR	9
2.3.3	ENERGIA POTENCIAL DE UM ROBÔ MANIPULADOR	10
2.3.4	EQUAÇÕES DE MOVIMENTO	10
2.3.5	MODELAGEM DO EFEITO DO ATRITO	11
2.3.6	TORQUE NAS JUNTAS DEVIDO A FORÇAS EXTERNAS	11
2.3.7	DINÂMICA INVERSA	11
2.3.8	DINÂMICA DIRETA	11
2.3.9	PARÂMETROS DINÂMICOS	12
2.4	IDENTIFICAÇÃO DO MODELO DINÂMICO	12
2.4.1	RESOLUÇÃO DAS EQUAÇÕES DE IDENTIFICAÇÃO	12
2.4.2	IDENTIFICABILIDADE DOS PARÂMETROS DINÂMICOS	13
2.4.3	ESCOLHA DA TRAJETÓRIA	14
2.4.4	DESVIO PADRÃO DOS PARÂMETROS IDENTIFICADOS	15
2.5	ESTIMAÇÃO DA FORÇA DE INTERAÇÃO	15
2.5.1	FILTRO DE KALMAN	16
2.5.2	TRANSFORMADA UNSCENTED	17
2.5.3	FILTRO DE KALMAN UNSCENTED	18
2.5.4	SQUARE-ROOT UNSCENTED KALMAN FILTER	19

2.6	ROS.....	21
2.7	UR3.....	23
<b>3</b>	<b>Modelagem dinâmica, Identificação e Estimação de força .....</b>	<b>25</b>
3.1	CINEMÁTICA DIRETA .....	25
3.2	MODELO DINÂMICO .....	26
3.2.1	MODELAGEM .....	27
3.3	IDENTIFICAÇÃO DOS PARÂMETROS DINÂMICOS.....	29
3.4	IMPLEMENTAÇÃO DO FILTRO DE KALMAN UNSCENTED .....	30
3.5	COMUNICAÇÃO E CONTROLE DO UR3 PELO ROS .....	31
<b>4</b>	<b>Resultados.....</b>	<b>33</b>
4.1	IDENTIFICAÇÃO .....	33
4.2	ESTIMAÇÃO DE FORÇA .....	41
<b>5</b>	<b>Conclusão .....</b>	<b>47</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>50</b>

# LISTA DE FIGURAS

2.1	Diagrama mostrando os parâmetros de DH $\theta_1$ , $d_1$ , $a_1$ e $\alpha_1$ [1].....	6
2.2	Diagrama simplificado de um sistema ROS (“mensagem” abreviada como “Msg”) ...	23
2.3	Manipulador robótico UR3 [2].....	23
3.1	Sistemas de coordenadas dos links do UR3, atribuídos segundo a convenção de Denavit-Hartenberg. ....	25
3.2	Trajatória de identificação 0. ....	29
3.3	Trajatória de identificação 1. ....	29
3.4	Trajatória de identificação 2. ....	30
3.5	Diagrama mostrando o funcionamento da comunicação entre o computador (nó “Interface”) e o UR3.....	32
4.1	Diferenças entre a trajetória 1 planejada e a executada pelo UR3 para a junta 1. ....	33
4.2	Diferenças entre a trajetória 2 planejada e a executada pelo UR3 para a junta 1. ....	34
4.3	Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 0 sem modelar o atrito. ....	35
4.4	Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 0 modelando o atrito.....	35
4.5	Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 1 sem modelar o atrito. ....	36
4.6	Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 1 modelando o atrito.....	36
4.7	Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 2 sem modelar o atrito. ....	37
4.8	Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 2 modelando o atrito.....	37
4.9	Torques preditos opostos aos mensurados. Sinais da junta 1.....	40
4.10	Acelerações estimadas pelo UKF na Trajetória 0.....	42
4.11	Torques estimados pelo UKF na Trajetória 0. ....	42
4.12	Forças estimadas pelo UKF na Trajetória 0. ....	43
4.13	Acelerações estimadas pelo UKF na Trajetória 1.....	43
4.14	Torques estimados pelo UKF na Trajetória 1. ....	44
4.15	Forças estimadas pelo UKF na Trajetória 1. ....	44

4.16	Visualização de uma pequeno trecho dos sinais de aceleração. Ambos são da junta 2, mas um é do experimento com a Trajetória 0 e outro com a Trajetória 1.....	45
4.17	Forças estimadas pelo UKF na Trajetória 0. ....	46

# LISTA DE TABELAS

3.1	Parâmetros de Denavit-Hartenberg para o UR3.....	26
4.1	Números de condição das trajetórias de identificação executadas.....	34
4.2	Erro médio quadrático das validações.....	38
4.3	Modelo sem atrito.....	38
4.4	Modelo com atrito.....	38
4.5	Valores dos parâmetros identificados com as Trajetórias 0 e 2.....	38
4.6	Desvios padrões dos erros de estimação dos parâmetros. As colunas “sa” correspondem ao modelo sem atrito e as colunas “ca” ao modelo com atrito.....	39

# LISTA DE SÍMBOLOS

$a$	Parâmetro de Denavit-Hartenberg correspondente à translação ao longo do eixo $x$
$A(q, \dot{q})$	Componente do modelo dinâmico independente das acelerações
$\alpha$	Parâmetro de Denavit-Hartenberg correspondente à rotação em torno do eixo $x$
$B(q)$	Componente do modelo dinâmico que multiplica as acelerações, igual a $D(q)$
$c$	Número de parâmetros dinâmicos
$c_{ijk}$	Símbolos de Christoffel do primeiro tipo
$C$	Matriz de covariância
$C(q)$	Matriz de efeitos centrífugos e coriolis do modelo dinâmico
$d$	Parâmetro de Denavit-Hartenberg correspondente à translação ao longo do eixo $z$
$D(q)$	Matriz de inércia do modelo dinâmico
$\delta$	Componente de Fourier da trajetória de identificação
$g(q)$	Matriz do efeito da gravidade do modelo dinâmico
$I$	Tensor de inércia
$J$	Jacobiano
$J_v$	Componente da velocidade linear do Jacobiano
$J_\omega$	Componente da velocidade angular Jacobiano
$\mathcal{K}$	Energia cinética, ou
$\mathcal{K}$	Ganho de Kalman
$\mathcal{L}$	Lagrangiano
$\lambda$	Componente polinomial da trajetória de identificação, ou
$\lambda$	Parâmetro do Filtro de Kalman Unscented
$m$	Massa do link
$n$	Número de juntas do robô, no caso, 6
$\nu_c$	Coefficiente de atrito de Coulomb
$\nu_v$	Coefficiente de atrito viscoso
$\sigma_j^i$	Vetor de translação do sistema de coordenadas $i$ para o sistema de coordenadas $j$
$P$	Matriz de covariância do erro de estimação no filtro de Kalman

$\mathcal{P}$	Energia potencial	
$q$	Vetor de posições das juntas $q = [q_1, q_2, q_3, q_4, q_5, q_6]^T$	[rad]
$\dot{q}$	Vetor de velocidades das juntas $\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]^T$	[rad/s]
$\ddot{q}$	Vetor de acelerações das juntas $\ddot{q} = [\ddot{q}_1, \ddot{q}_2, \ddot{q}_3, \ddot{q}_4, \ddot{q}_5, \ddot{q}_6]^T$	[rad/s <sup>2</sup> ]
$\ddot{q}_{dyn}$	Acelerações fornecidas pelo modelo dinâmico direto	[rad/s <sup>2</sup> ]
$Q$	Matriz de covariância do erro do processo	
$R_j^i$	Matriz de rotação do sistema de coordenadas $i$ para o sistema de coordenadas $j$	
$R$	Matriz de covariância do erro de medição	
$r$	Número de pontos na trajetória	
$rx$	Coordenada $x$ do centro de massa do link, expressa no sistema de coordenadas do link	
$ry$	Coordenada $y$ do centro de massa do link, expressa no sistema de coordenadas do link	
$rz$	Coordenada $z$ do centro de massa do link, expressa no sistema de coordenadas do link	
$\rho$	Erro residual da identificação	
$S$	Raiz quadrada (decomposição de Cholesky) da matriz de covariância do erro de estimação	
$\sigma$	Valor singular, obtido pela decomposição em valores singulares, ou	
$\sigma$	Desvio padrão	
$T$	Vetor obtido através da concatenação vertical dos torques das juntas em vários pontos de uma trajetória	
$T_j^i$	Matriz de transformação homogênea do sistema de coordenadas $i$ para o sistema de coordenadas $j$	
$\tau$	Torque	[N.m]
$\tau_m$	Torque devido ao movimento	[N.m]
$\tau_{fr}$	Torque devido ao atrito	[N.m]
$\tau_F$	Torque devido ao esforço externo no efetuador terminal	[N.m]
$\tau_{idyn}$	Torques fornecidos pelo modelo dinâmico inverso	[N.m]
$\theta$	Vetor de parâmetros dinâmicos, ou	
$\theta$	Parâmetro de Denavit-Hartenberg correspondente à rotação em torno do eixo $z$	
$v$	Ruído gaussiano na medição	
$w(q, \dot{q}, \ddot{q})$	Matriz de regressão, que multiplica os parâmetros dinâmicos para obter o torque	
$w$	Ruído gaussiano no processo	
$W$	Vetor obtido através da concatenação vertical da matriz $w$ em vários pontos de uma trajetória	
$W^m$	Vetor de pesos para calculo da média	
$W^c$	Vetor de pesos para calculo da covariância	

$\mathcal{X}$  Pontos sigma

## Siglas

CAS	Computer Algebra System
EKF	Filtro de Kalman Extendido
UKF	Filtro de Kalman Unscented

# Capítulo 1

## Introdução

### 1.1 Motivação

Desde o surgimento da área da robótica, o ambiente industrial tem sido terreno fértil para o desenvolvimento e aplicação de novas tecnologias. A partir do Unimate em 1959 [3], a associação robótica-indústria formou-se e cresceu, numa relação que frequentemente envolve a pesquisa em robótica patrocinada pela indústria e esta última colhendo benefícios das pesquisas, como aumento em produtividade, qualidade e bem estar dos operários.

Uma área recente e importante de pesquisa, que tem grande potencial de aplicações no meio da indústria, é a cooperação humano-robô. Atualmente, embora haja uma grande taxa de robotização da produção, separa-se as regiões de trabalho dos operários humanos das regiões de trabalho dos robôs. Isso é feito, e até certo ponto com razão, por preocupações com a segurança dos trabalhadores. Embora tradicionalmente os robôs sejam muito bons em executar as tarefas para as quais eles foram programados, eles têm grandes dificuldade em lidar com situações imprevistas. Desse modo, é possível que durante a sua movimentação um manipulador robótico acabe atingindo um ser humano próximo, podendo causar severas lesões [4].

No entanto, os benefícios que poderiam advir de um sistema de cooperação homem-robô são múltiplos. Associando a precisão e repetibilidade de um robô com o extenso sistema sensorial humano seria possível elevar a qualidade e produtividade das indústrias a novos patamares.

Em um esforço na direção de colher esses benefícios, avanços vem sendo feitos para melhorar a segurança nesse tipo de interação, tanto no campo de regulamentação quanto no campo técnico. Enquanto que o primeiro está fora do escopo deste trabalho, o segundo está tematicamente próximo. No campo técnico, destaca-se o estudo de robôs complacentes.

Na Engenharia Mecânica, mecanismos complacentes são aqueles que ganham ao menos parte da sua mobilidade a partir da deflexão de membros flexíveis, em vez de apenas juntas móveis [5]. Na robótica, esse conceito está mais associado à flexibilidade do robô, ou à sua habilidade de reagir a forças de interação [6]. Um robô complacente seria aquele que, ao ter sua trajetória obstruída por um obstáculo, cessaria o seu movimento em vez de continuar exercendo força como

se o obstáculo não existisse.

A complacência contribuiria para a solução do problema da segurança na colaboração entre humanos e robôs no sentido de que caso ocorra uma colisão entre um robô manipulador e um operário, o controle complacente do robô atenuaria o choque de maneira a potencialmente prevenir uma lesão. Outro benefício da complacência em robôs seria a possibilidade do ser humano de guiar o robô e este serviria para “aumentar” a capacidade do operador.

Existem robôs onde a complacência está no próprio hardware do robô, com links flexíveis ou molas nos atuadores. Pode-se também realizar um controle complacente de um manipulador. No entanto, para esse último caso, é necessário conhecer as forças externas que atuam sobre o robô.

Existem sensores de força que podem ser acoplados ao efetuador terminal de um manipulador que fornecem as informações das forças de interação. Caso a utilização desse sensor não seja desejada, há formas de estimar a força externa sobre o efetuador terminal.

## 1.2 Estimação de força

Existem diversas maneiras de se estimar a força no efetuador terminal de um robô sem o sensor de força [7].

Uma maneira é usar os valores de torque nas juntas, em conjunto com um modelo dinâmico do robô. No entanto, esse método exige compensação das forças de perturbação (como atrito e gravidade). Ele também pode sofrer com ruído caso os motores das juntas tenham engrenagens de redução com razões grandes. Apesar disso, existem trabalhos [8] que mostram que essa abordagem é viável e possui resultados aceitáveis. Considerando o escopo deste trabalho, o UR3 em específico não tem sensores de torque nas juntas, mas é possível obter as correntes nos motores, que podem ser usadas para estimar os torques.

Outro método é utilizar observadores de perturbação. Desse modo, a partir do modelo do robô, pode-se atribuir os desvios no comportamento a perturbações causadas por forças externas. Existem diversas implementações baseadas em observadores.

Em [9] é proposto um método de estimação dos torques nas juntas usando observadores de perturbação em cada junta. O torque de perturbação estimado é realimentado para um controle de aceleração que, juntamente com o modelo dinâmico do manipulador, calcula o torque de reação e assim é realizado o controle de torque sem sensor. Embora o valor estimado não seja a força no efetuador terminal, é possível obter essa informação a partir do modelo do robô e dos torques estimados.

Colomé, Pardo, Alenyà e Torras propõem [10] o uso de um observador de perturbação para estimar a força externa. Utiliza-se um aprendizado local (devido ao problema da dimensionalidade com muitos graus de liberdade) para aprender um modelo dinâmico inverso do robô. Além disso, através de manipulações nas equações do modelo do sistema, realiza-se um aprendizado isolando as componentes dependentes da aceleração, que é tipicamente obtida através da derivação dupla do sinal de posição e, por isso, é extremamente ruidosa.

Depois de considerar os métodos citados acima, decidiu-se utilizar as informações do torque nas juntas do robô (provenientes das correntes), juntamente com um modelo dinâmico e um estimador de estados para estimar a força.

### **1.3 Objetivo**

O objetivo deste trabalho é desenvolver e implementar um método de estimação da força de interação externa com um braço robótico UR3, sem fazer uso de um sensor de força. Este trabalho vai proporcionar a eventual possibilidade do desenvolvimento de um sistema de controle de força para o UR3.

### **1.4 Resultados obtidos**

Foi obtido um modelo dinâmico do UR3, sobre o qual foi feito um procedimento de identificação dos parâmetros dinâmicos. Conjuntamente com outros alunos, uma interface com o robô foi desenvolvida utilizando o ROS, permitindo enviar comandos e receber leituras dos sensores. Implementou-se o Filtro de Kalman Unscented para fazer a estimação da força. Embora não tenha sido possível obter resultados quantitativos, os resultados qualitativos obtidos demonstram a viabilidade desse método de estimação de força e apontam um caminho claro para melhorias futuras.

### **1.5 Apresentação do manuscrito**

O Capítulo 2 fornece uma fundamentação teórica sobre os conceitos mais importantes para a realização e compreensão deste trabalho, que são a modelagem dinâmica do robô, a identificação dos parâmetros dinâmicos do modelo e a estimação de estados. São apresentadas equações e fórmulas, bem como o raciocínio e a intuição por trás. O Capítulo 3 consiste numa explicação dos procedimentos adotados durante a realização deste trabalho, mostrando como os fundamentos apresentados no capítulo anterior foram aplicados no desenvolvimento do trabalho. O Capítulo 4 apresenta os resultados obtidos na identificação dos parâmetros dinâmicos e na estimação da força de interação, mostrados sob a forma de gráficos e tabelas, bem como discussões sobre estes resultados. Finalmente, o Capítulo 5 conclui este manuscrito com uma análise retrospectiva sobre o trabalho como um todo e apresenta considerações a serem levadas em conta em uma futura continuação do trabalho.

## Capítulo 2

# Fundamentos de Modelagem dinâmica e Estimação de estados

Este capítulo fornece uma breve apresentação da teoria relacionada com este trabalho. Ele começa pelos conceitos relacionados com a modelagem do robô manipulador, partindo da cinemática direta, passando pela cinemática diferencial e finalmente chegando à modelagem dinâmica. Essas seções foram primariamente baseadas em [1] e [11]. Em seguida, expõe-se os conceitos de identificação dos parâmetros dinâmicos, tirados principalmente de [12]. Depois são apresentados os conceitos por trás do estimador de estados, o Filtro de Kalman Unscented, que foram extraídos de [13] e [14]. Finalmente, é feita uma exposição sobre o ROS e sobre o robô UR3.

### 2.1 Cinemática direta

Em robótica, o modelo cinemático relaciona as variáveis de junta à posição do efetuador terminal. A cinemática direta (ou modelo cinemático direto) obtém a posição do efetuador terminal a partir das variáveis de junta, enquanto que a cinemática inversa (ou modelo cinemático inverso) faz o contrário. Tipicamente, o cálculo das equações cinemáticas envolve **transformações homogêneas**.

#### 2.1.1 Transformações homogêneas

Transformações homogêneas relacionam dois sistemas de coordenadas através de uma multiplicação de matrizes. A matriz de transformação homogênea  $H \in \mathbb{R}^{4 \times 4}$  tem a forma [1]

$$H = \begin{bmatrix} R & o \\ \mathbf{0} & 1 \end{bmatrix} \quad (2.1)$$

em que  $R \in \mathbb{R}^{3 \times 3}$  é uma matriz que descreve a rotação entre os sistemas de coordenadas e  $o \in \mathbb{R}^3$  é um vetor que representa a translação entre eles.

Para aplicar múltiplas transformações (como quando se realiza a transformação entre os sistemas de coordenadas subsequentes associados aos links de um robô), basta multiplicar as matrizes deste modo:

$$H_j^i = H_{i+1}^i \cdots H_j^{j-1} \quad (2.2)$$

em que  $H_j^i$  indica a transformação homogênea do sistema de coordenadas  $i$  para o sistema de coordenadas  $j$ .

Embora a atribuição de sistemas de coordenadas aos links de um robô seja arbitrária, existem convenções que visam sistematizar ou simplificar esse processo. Uma de tais convenções, e a mais famosa, é a convenção de **Denavit-Hartenberg**.

### 2.1.2 Conveção de Denavit-Hartenberg

De maneira geral, seriam necessários 6 parâmetros para definir a mudança entre sistemas de coordenadas em um espaço tridimensional: 3 ângulos de rotação e 3 distâncias de translação. A convenção de Denavit-Hartenberg consiste em uma estratégia de escolha da origem dos sistemas de coordenadas dos links de um robô de maneira a reduzir o número destes parâmetros para 4. Na convenção DH (Denavit-Hartenberg), cada transformação homogênea  $A_i$  é uma combinação de 4 transformações básicas: rotação em torno do eixo  $z$ , translação ao longo do eixo  $z$ , translação ao longo do eixo  $x$  e rotação em torno do eixo  $x$ . Matematicamente, isso se torna

$$\begin{aligned} H_i^{i-1} &= \text{Rot}_{z,\theta_i} \text{Trans}_{z,d_i} \text{Trans}_{x,a_i} \text{Rot}_{x,\alpha_i} \\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_ic_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_is_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.3)$$

em que se usa a notação  $s_x = \text{sen}(x)$  e  $c_x = \text{cos}(x)$  e  $\theta$ ,  $d$ ,  $a$  e  $\alpha$  são os 4 parâmetros de DH referidos anteriormente.

Antes de se obter estes parâmetros, é necessário atribuir os sistemas de coordenadas segundo a convenção de DH. Desse modo, os sistemas de coordenadas  $o_i x_i y_i z_i$ , que são associados à junta  $j_i$  e ao link  $l_i$  devem obedecer as seguintes regras (no caso de um robô que tem apenas juntas de revolução) [1]:

1. O eixo  $z_i$  é o eixo de revolução da junta  $j_{i+1}$ ;
2. O eixo  $x_i$  é perpendicular aos eixos  $z_{i-1}$  e  $z_i$ ;
3. O eixo  $x_i$  intercepta o eixo  $z_{i-1}$ ; e

4. O eixo  $y_i$  é determinado pela regra da mão direita.

Após atribuir os sistemas de coordenadas, os parâmetros  $\theta_i$ ,  $d_i$ ,  $a_i$  e  $\alpha_i$  são determinados desta forma [1]:

1.  $\theta_i$  é o ângulo de  $x_{i-1}$  para  $x_i$  em torno de  $z_{i-1}$ ;
2.  $d_i$  é a distância ao longo de  $z_{i-1}$  entre  $o_{i-1}$  e a interseção dois eixos  $x_i$  e  $z_{i-1}$ ;
3.  $a_i$  é a distância ao longo de  $x_i$  entre a interseção dos eixos  $x_i$  e  $z_{i-1}$  e  $o_i$ ; e
4.  $\alpha_i$  é o ângulo de  $z_{i-1}$  para  $z_i$  em torno de  $x_i$ .

A Figura 2.1 mostra a obtenção dos parâmetros de DH para dois sistemas de coordenadas arbitrários.

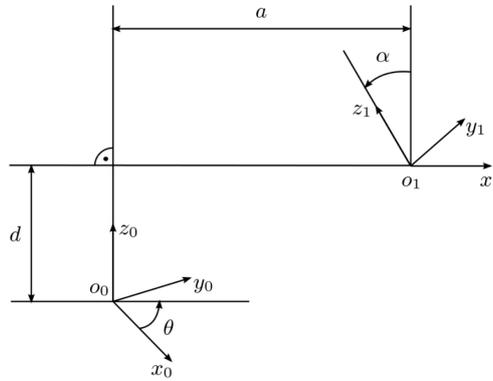


Figura 2.1: Diagrama mostrando os parâmetros de DH  $\theta_1$ ,  $d_1$ ,  $a_1$  e  $\alpha_1$  [1].

Como o UR3 possui apenas juntas de revolução, os parâmetros  $d_i$ ,  $a_i$  e  $\alpha_i$  são fixos, enquanto que  $\theta_i$  corresponde ao deslocamento angular das juntas produzido pelos atuadores e é, portanto, variável.

De posse dos parâmetros de DH do robô, é possível calcular as matrizes de transformação homogênea segundo a Equação (2.3) e, ao multiplicar as matrizes, obter a cinemática direta do robô.

Para obter a velocidade do efetuador terminal, desenvolve-se a **cinemática diferencial**.

## 2.2 Cinemática diferencial

A cinemática diferencial relaciona as velocidades linear e angular do efetuador terminal às velocidades de junta. A cinemática direta define uma função que fornece a posição e orientação do efetuador terminal a partir das posições das juntas. A cinemática diferencial é determinada pelo Jacobiano dessa função [1].

O Jacobiano  $J \in \mathbb{R}^{6 \times n}$  de um manipulador representa a transformação instantânea entre as velocidades de junta  $\dot{q} \in \mathbb{R}^n$  e as velocidades linear e angular do efetuador terminal  $\xi \in \mathbb{R}^6$ . É possível também obter Jacobianos que, a partir das velocidades de junta, fornecem as velocidades linear e angular de qualquer ponto do robô.

Além do cálculo de velocidades, o Jacobiano também é usado no planejamento e execução de trajetórias suaves, determinação de configurações singulares, cálculo das equações dinâmicas de movimento, transformação de esforços aplicados no efetuador terminal para esforços nas juntas, entre outros.

## 2.2.1 Jacobiano

A relação entre as velocidades angular e linear do efetuador terminal e as velocidades de junta é dada por

$$\xi = J\dot{q} \quad (2.4)$$

em que

$$\xi = \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} \quad \text{e} \quad J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad (2.5)$$

sendo, respectivamente,  $v_n^0 \in \mathbb{R}^3$  e  $\omega_n^0 \in \mathbb{R}^3$  os vetores de velocidade linear e angular do efetuador terminal (link  $n$ ) com respeito ao referencial inercial (sistema de coordenadas 0), e  $J_v \in \mathbb{R}^{3 \times n}$  e  $J_\omega \in \mathbb{R}^{3 \times n}$  as componentes do Jacobiano associadas às referidas velocidades.

O cálculo do Jacobiano do efetuador terminal é feito usando as matrizes de transformação homogênea do manipulador, calculadas usando a convenção de Denavit-Hartenberg conforme a Seção 2.1.2.

Primeiro, define-se a transformação do sistema de coordenadas inercial (base, ou link 0) para o sistema de coordenadas do link  $i$  como

$$T_i^0(q) = \begin{bmatrix} R_i^0(q) & o_i^0(q) \\ \mathbf{0} & 1 \end{bmatrix} = H_1^0 \dots H_i^{i-1} \quad (2.6)$$

Para um robô de  $n$  links, a parte linear do Jacobiano é

$$J_v = [J_{v_1}, \dots, J_{v_n}] \quad (2.7)$$

na qual

$$J_{v_i} = \frac{\partial o_n^0}{\partial q_i} = z_{i-1}^0 \times (o_n^0 - o_{i-1}^0) \quad (2.8)$$

sendo

$$z_{i-1}^0 = R_{i-1}^0 k \quad (2.9)$$

com  $k = [0, 0, 1]^T$ .

A parte angular do Jacobiano é dada por

$$J_\omega = [z_0^0, \dots, z_{n-1}^0] \quad (2.10)$$

O Jacobiano  $J$  é calculado substituindo as expressões (2.7) e (2.10) em (2.5).

O cálculo do Jacobiano de um ponto arbitrário em um link, que será usado para calcular o modelo dinâmico do manipulador na Seção 2.3.2, será definido a seguir.

### 2.2.1.1 Jacobiano de um ponto arbitrário em um link

Dado um vetor  $r_{P_j}^0$  que contém a posição de um ponto  $P$  no link  $j$  expressa no sistema de coordenadas inercial, a componente linear do jacobiano,  $J_{P_j,v}$ , é

$$\begin{aligned} J_{P_j,v} &= \frac{\partial r_{P_j}^0}{\partial q_i} = z_{i-1}^0 \times (r_{P_j}^0 - o_{i-1}^0), \quad \forall i \leq j \\ J_{P_j,v} &= \mathbf{0}, \quad \forall i > j \end{aligned} \quad (2.11)$$

Do mesmo modo, a componente angular é dada por

$$\begin{aligned} J_{P_j,\omega} &= z_{i-1}^0, \quad \forall i \leq j \\ J_{P_j,\omega} &= \mathbf{0}, \quad \forall i > j \end{aligned} \quad (2.12)$$

Assim, o Jacobiano  $J_{P_j}(q)$  consiste em

$$J_{P_j}(q) = \begin{bmatrix} z_0^0 \times (r_{P_j}^0 - o_0^0) & \dots & z_{j-1}^0 \times (r_{P_j}^0 - o_{j-1}^0) & \mathbf{0} & \dots & \mathbf{0} \\ z_0^0 & \dots & z_{j-1}^0 & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \quad (2.13)$$

Aqui vale ressaltar que as colunas do Jacobiano são  $\mathbf{0}$  para  $i > j$  porque a velocidade do link  $j$  não é afetada pelo movimento dos links que vêm depois dele.

Sabendo obter os Jacobianos do manipulador, pode-se iniciar a obtenção do **modelo dinâmico**

## 2.3 Modelo dinâmico

Enquanto as equações cinemáticas descrevem o movimento do robô sem considerar as forças e os torques produzindo o movimento, as equações dinâmicas descrevem explicitamente as relações entre força e movimento [11].

Existem diversos algoritmos para se obter o modelo dinâmico de um robô, mas de forma geral eles consistem em variações de dois métodos: a equação de **Euler-Lagrange** e a formulação de **Newton-Euler**. O primeiro é derivado a partir do princípio de trabalho virtual, analisando o sistema do ponto de vista da sua energia, enquanto que o segundo pode ser aplicado recursivamente sobre cada link do robô manipulador, característica muito atrativa quando se trata de manipuladores com muitos links.

Neste trabalho, escolheu-se utilizar a modelagem de Euler-Lagrange.

### 2.3.1 Equação de Euler-Lagrange

Define-se o Lagrangiano  $\mathcal{L}$ :

$$\mathcal{L} = \mathcal{K} - \mathcal{P} \quad (2.14)$$

sendo  $\mathcal{K}$  é a energia cinética do sistema e  $\mathcal{P}$  é a energia potencial.

Partindo da segunda lei de Newton e das equações de movimento de um sistema, é possível chegar à equação de Euler-Lagrange [11], neste caso já aplicadas ao caso de um robô manipulador:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \quad (2.15)$$

em que  $q_i$  são as chamadas *coordenadas generalizadas*, aqui equivalentes à posição angular das juntas do robô, e  $\tau_i$  é a *força generalizada*, nesse caso correspondente ao torque exercido pelos atuadores nas juntas.

### 2.3.2 Energia cinética de um robô manipulador

Sabe-se que a energia cinética de um corpo rígido é dada pela soma da sua energia cinética translacional com a sua energia cinética rotacional:

$$\mathcal{K} = \frac{1}{2} m v^T v + \frac{1}{2} \omega^T \mathcal{I} \omega \quad (2.16)$$

em que  $m$  é a massa do corpo,  $v$  e  $\omega$  são respectivamente os vetores de velocidade linear e angular e  $\mathcal{I}$  é o tensor de inércia.

Considerando um robô manipulador de  $n$  links, a sua energia cinética é

$$\mathcal{K} = \frac{1}{2} \dot{q}^T \sum_{i=1}^n \left[ m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i^0(q) I_i R_i^0(q)^T J_{\omega_i}(q) \right] \dot{q} \quad (2.17)$$

na qual  $J_{v_i}$  e  $J_{\omega_i}$  são as componentes do Jacobiano do centro de massa do link  $i$  (calculado conforme a Equação (2.13)) associadas às velocidades linear e angular e  $R_i^0(q)$  é a matriz de rotação entre o sistema de coordenadas associado ao link  $i$  e o sistema de coordenadas inercial, que é utilizada para transformar o tensor de inércia  $I$  expresso no sistema de coordenadas do link em um tensor de inércia expresso no sistema de coordenadas inercial.

A equação (2.17) pode ser condensada na forma

$$\mathcal{K} = \frac{1}{2} \dot{q}^T D(q) \dot{q} \quad (2.18)$$

em que  $D(q)$  é a **matriz de inércia**.

### 2.3.3 Energia potencial de um robô manipulador

Considerando que o robô manipulador possui apenas links rígidos, a única forma de energia potencial é a gravitacional. A energia potencial total de um robô com  $n$  links é então

$$\mathcal{P} = \sum_{i=1}^n g^T r_{c_i} m_i \quad (2.19)$$

em que  $g$  é o vetor da aceleração da gravidade e  $r_{c_i}$  é o vetor que indica a posição do centro de massa do link  $i$ .

### 2.3.4 Equações de movimento

Substituindo as equações (2.18) e (2.19) em (2.14), calcula-se o Lagrangiano do robô:

$$\mathcal{L} = \mathcal{K} - \mathcal{P} = \frac{1}{2} \sum_{i,j} d_{ij}(q) \dot{q}_i \dot{q}_j - \mathcal{P}(q) \quad (2.20)$$

Calculando as derivadas necessárias, obtém-se

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_k} = \sum_j d_{kj} \ddot{q}_j + \sum_{i,j} \frac{\partial d_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j \quad (2.21)$$

$$\frac{\partial \mathcal{L}}{\partial q_k} = \frac{1}{2} \sum_{i,j} \frac{\partial d_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j - \frac{\partial \mathcal{P}}{\partial q_k} \quad (2.22)$$

Substituindo em (2.15)

$$\sum_j d_{kj} \ddot{q}_j + \sum_{i,j} \left\{ \frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2} \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i \dot{q}_j + \frac{\partial \mathcal{P}}{\partial q_k} = \tau_k \quad (2.23)$$

É possível manipular o somatório da segunda parcela levando em consideração a simetria e expressá-lo em função dos **símbolos de Christoffel** (do primeiro tipo)  $c_{ijk}$ , definidos como

$$c_{ijk} := \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \quad (2.24)$$

A equação (2.23) fica

$$\sum_j d_{kj}(q) \ddot{q}_j + \sum_{i,j} c_{ijk}(q) \dot{q}_i \dot{q}_j + \frac{\partial \mathcal{P}}{\partial q_k} = \tau_k, \quad k = 1, \dots, n \quad (2.25)$$

Finalmente, chega-se à forma matricial

$$\tau_m = D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) \quad (2.26)$$

sendo  $\tau_m$  o torque nas juntas devido ao movimento,  $D(q)$  a matriz de inércia definida na equação (2.18),  $g(q)$  um vetor contendo a derivada da energia potencial com respeito a cada variável de junta, e a matriz  $C(q, \dot{q})$  consiste de elementos  $c_{kj}$  da forma

$$c_{kj} = \sum_{i=1}^n \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i \quad (2.27)$$

### 2.3.5 Modelagem do efeito do atrito

Uma das formas de modelar o impacto do atrito nos torques das juntas é considerando uma composição entre o atrito de Coulomb e o atrito viscoso. Isso se dá por

$$\tau_{fr_i} = \nu_{ci} \text{sign}(\dot{q}_i) + \nu_{vi} \dot{q}_i \quad (2.28)$$

sendo  $\nu_{ci}$  e  $\nu_{vi}$  os coeficientes de atrito de Coulomb e viscoso, respectivamente, e  $\text{sign}(x)$  uma função que retorna -1 se a entrada for negativa, 0 se ela for nula e 1 se for positiva. No entanto, tipicamente usa-se uma margem de erro  $|\delta|$  para a região nula.

### 2.3.6 Torque nas juntas devido a forças externas

Baseando-se no princípio do trabalho virtual, é possível provar que os torques exercidos pelos atuadores das juntas do robô para compensar o efeito de uma força externa no efetuador terminal são dados pela relação

$$\tau_F = J^T F_{ext} \quad (2.29)$$

em que  $J$  é o Jacobiano do efetuador terminal e  $F_{ext}$  é o vetor que descreve as componentes de força e momento aplicadas sobre o efetuador terminal.

### 2.3.7 Dinâmica inversa

Agora é possível definir o modelo dinâmico inverso, que fornece o torque nas juntas do manipulador a partir das variáveis de junta e da força no efetuador terminal. Combinando as Equações (2.26), (2.28) e (2.29), obtém-se

$$\tau(q, \dot{q}, \ddot{q}, F_{ext}) = \tau_m + \tau_F + \tau_{fr} = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \nu_c \text{sign}(\dot{q}) + \nu_v \dot{q} + J^T F_{ext} \quad (2.30)$$

### 2.3.8 Dinâmica direta

Considerando que as acelerações das juntas não são medidas diretamente, mas só são obtidas a partir da derivação do sinal de velocidade e, por causa disso, são muito suscetíveis a ruído, convém definir a dinâmica direta.

O modelo dinâmico direto fornece as acelerações nas juntas com base nas posições e velocidades, no torque e na força externa exercida sobre o efetuador terminal. Para obter uma fórmula analítica, primeiro reescreve-se a Equação (2.30) na forma

$$\begin{aligned} \tau &= A(q, \dot{q}) + \bar{B}(q, \ddot{q}) + J^T F_{ext} \\ \tau &= A(q, \dot{q}) + B(q)\ddot{q} + J^T F_{ext} \end{aligned} \quad (2.31)$$

em que

$$A(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q) + \tau_{fr} \quad \text{e} \quad B(q) = D(q) \quad (2.32)$$

A dinâmica direta pode então ser escrita como

$$\ddot{q}(q, \dot{q}, \tau, F_{ext}) = B(q)^{-1} [\tau - A(q, \dot{q}) - J^T F_{ext}] \quad (2.33)$$

### 2.3.9 Parâmetros dinâmicos

O modelo dinâmico da Equação (2.30), desconsiderando o termo referente à força externa, pode ser reescrito da forma

$$\tau = w(q, \dot{q}, \ddot{q})\theta \quad (2.34)$$

na qual  $w(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times r}$  é chamada matriz de regressão e  $\theta \in \mathbb{R}^r$  é um vetor contendo os chamados parâmetros dinâmicos do sistema. Esses parâmetros dizem respeito às características inerciais do sistema, como elementos do tensor de inércia ou massa dos links.

Os parâmetros dinâmicos do UR3 serão obtidos através da **identificação**, que será explicada na Seção 2.4.

## 2.4 Identificação do modelo dinâmico

Como foi visto na seção anterior, o modelo dinâmico do robô manipulador depende dos chamados parâmetros dinâmicos, que são os tensores de inércia, massa e posição dos centros de massa dos links. Esses parâmetros adéquam as equações do modelo ao robô físico, concreto.

Esses parâmetros podem ser obtidos de algumas maneiras. Eles podem ser fornecidos pelo fabricante do robô ou podem ser obtidos através de medições, desmontando o robô e pesando os seus links, encontrando os centros de massa, usando algum software CAD pra obter os tensores de inércia. Frequentemente os fabricantes não fornecem as informações (ou não todas elas), e mesmo se fornecessem, seriam apenas dados de especificação e o robô em particular poderia ser um pouco diferente. Em contrapartida, fazer medições é um processo trabalhoso e propenso a erro. Uma outra maneira de se obter os parâmetros dinâmicos é realizar uma identificação.

Uma identificação dos parâmetros dinâmicos de um robô consiste em realizar experimentos com o robô e, tomando os dados de entrada (trajetória, definida por  $q, \dot{q}, \ddot{q}$ ) e de saída (torques medidos,  $\tau$ ) e o modelo dinâmico (Equação (2.30)) encontrar os parâmetros dinâmicos que melhor se ajustam. As seções a seguir tratarão sobre esse tema [12].

### 2.4.1 Resolução das equações de identificação

Como o modelo dinâmico de Lagrange é linear em função dos parâmetros dinâmicos, o modelo de identificação pode ser escrito na forma da Equação (2.34).

Aplicando essa equação em um número suficiente de pontos (como nas amostras de uma trajetória) e concatenando verticalmente os valores obtidos para  $\tau$  e  $w$ , constrói-se o sistema de equações lineares determinado:

$$T = W(q, \dot{q}, \ddot{q})\theta + \rho \quad (2.35)$$

na qual, sendo  $r$  o número de pontos,  $c$  o número de parâmetros dinâmicos e 6 o número de juntas,  $T \in \mathbb{R}^{6r \times 1}$  é um vetor com os torques das juntas em cada ponto da trajetória,  $W \in \mathbb{R}^{6r \times c}$  é a chamada matriz de observação, ou regressor,  $\theta \in \mathbb{R}^{c \times 1}$  é o vetor de parâmetros dinâmicos e  $\rho$  é o vetor de erro residual.

Existem diversas maneiras de solucionar a Equação (2.35). Uma maneira que fornece bons resultados e tira proveito da linearidade com respeito aos parâmetros dinâmicos é a resolução através do método dos mínimos quadrados. Dessa forma, o valor estimado para os parâmetros dinâmicos é aquele que minimiza o quadrado do erro residual:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\rho\|^2 \quad (2.36)$$

cuja solução, caso  $W$  tenha posto completo, é dada por

$$\hat{\theta} = (W^T W)^{-1} W^T T = W^+ T \quad (2.37)$$

sendo  $W^+$  a matriz inversa de Moore-Penrose de  $W$ .

A condição de que  $W$  deve ter posto completo leva à consideração sobre quais parâmetros dinâmicos podem ser identificados, o que será discutido na próxima seção.

## 2.4.2 Identificabilidade dos parâmetros dinâmicos

Os parâmetros dinâmicos podem ser divididos em três grupos: completamente identificáveis, identificáveis em combinações lineares e não-identificáveis. Isso ocorre por causa da complexa cadeia cinemática do robô, onde alguns parâmetros não afetam o comportamento dinâmico (não-identificáveis) e outros afetam de maneira conjunta (identificáveis em combinações lineares). A consequência disso é que a matriz de observação  $W$  não possui posto completo e, portanto, a Equação (2.37) não pode ser usada.

Este problema pode ser resolvido encontrando o conjunto dos chamados *parâmetros dinâmicos base*, que é o conjunto dos parâmetros dinâmicos que podem ser identificados independentemente.

Entre as formas de encontrar os parâmetros dinâmicos base, uma que se destaca pela sua simplicidade é a numérica [15]. Ela consiste em calcular a matriz  $W$  para valores aleatórios de posição, velocidade e aceleração. Então, o posto de  $W$  determina o número de parâmetros dinâmicos base do modelo,  $n_{\theta_b}$ . Os parâmetros não identificáveis correspondem às colunas nulas de  $W$ . Encontra-se  $n_{\theta_b}$  colunas linearmente independentes entre as colunas não nulas de  $W$ , que correspondem a  $n_{\theta_b}$  parâmetros dinâmicos. Esses parâmetros são os parâmetros base e os demais parâmetros podem ser encontrados através de combinações lineares deles, sendo os coeficientes dessas combinações iguais aos coeficientes das combinações lineares das colunas de  $W$ .

Executando esse procedimento, obtém-se uma matriz  $W$  reduzida e de posto completo, com os parâmetros dinâmicos base correspondentes  $\theta$ , e utiliza-se a Equação (2.37) para encontrar os parâmetros base.

A componente faltante do processo de identificação é a escolha da trajetória, que será abordada na próxima seção.

### 2.4.3 Escolha da trajetória

Uma boa trajetória de identificação deve apresentar uma boa taxa de convergência e fazer a estimação pelos mínimos quadrados resistente ao ruído. Tal trajetória é chamada de trajetória de excitação persistente [12].

A sensibilidade da solução dos mínimos quadrados com respeito aos erros de modelagem e ruído está relacionada com o número de condição da matriz de observação  $W$  [12]. Gautier e Khalil [16] apresentam as relações

$$\frac{\|\delta\hat{\theta}\|}{\|\hat{\theta}\|} \leq \text{Cond}(W) \frac{\|\delta T\|}{\|T\|}, \quad \text{com } \delta W = 0 \quad (2.38)$$

$$\frac{\|\delta\hat{\theta}\|}{\|\hat{\theta} + \delta\hat{\theta}\|} \leq \text{Cond}(W) \frac{\|\delta W\|}{\|W\|}, \quad \text{com } \delta T = 0 \quad (2.39)$$

em que o número de condição de norma 2  $\text{Cond}(W)$  pode ser calculado a partir da decomposição em valores singulares (SVD) da matriz  $W$ , sendo ele a razão entre o maior valor singular  $\sigma_1$  e o menor  $\sigma_N$ :

$$\text{Cond}(W) = \frac{\sigma_1}{\sigma_N} \quad (2.40)$$

As Equações (2.38) e (2.39) mostram que o número de condição da matriz de observação fornece um limite superior aos erros de estimação dos parâmetros dinâmicos. Dessa forma, ao escolher uma trajetória de identificação, deve-se procurar minimizar o número de condição da matriz de observação correspondente a tal trajetória.

Tipicamente, toma-se uma trajetória parametrizada e executa-se um algoritmo de otimização sobre os parâmetros, minimizando o número de condição da matriz de observação. O espaço dos parâmetros é limitado pelas restrições de posição, velocidade e aceleração máxima das juntas, bem como pela posição do robô no espaço cartesiano de modo a evitar colisões com o ambiente e com si mesmo.

A parametrização de trajetórias para identificação dos parâmetros dinâmicos evoluiu consideravelmente com o tempo. Gautier e Khalil [16] propuseram uma parametrização polinomial, com um conjunto de posições e velocidades separados no tempo, que eram interpoladas para obter a trajetória completa. Swevers et al. utilizaram uma série de Fourier para definir a trajetória, o que permitia uma excitação periódica (possibilitando a execução de uma média no domínio do tempo para reduzir o ruído) e obtenção analítica das velocidades e acelerações (evitando ter que derivar sinais ruidosos dos sensores).

A forma de parametrização escolhida foi a proposta por Park [17]. Essa parametrização consiste na soma entre uma componente polinomial e uma componente de série de Fourier. Essa técnica apresenta vantagens na convergência da otimização e permite a escolha das condições iniciais e finais. A parametrização é dada por

$$q_i(t) = \lambda_i(t) + \delta_i(t) \quad (2.41)$$

em que

$$\lambda_i(t) = \sum_{j=0}^5 \lambda_{ij} t^j, \quad \delta_i(t) = \sum_{l=1}^N a_{il} \cos(l\omega_0 t) \quad (2.42)$$

sendo  $\omega_0$  a frequência da oscilação. O termo constante da série de Fourier está incluso no polinômio.

Pode-se escrever as condições de contorno

$$q_i(0) = \lambda_i(0) + \delta_i(0), \quad q_i(t_f) = \lambda_i(t_f) + \delta_i(t_f) \quad (2.43)$$

$$\dot{q}_i(0) = \dot{\lambda}_i(0) + \dot{\delta}_i(0), \quad \dot{q}_i(t_f) = \dot{\lambda}_i(t_f) + \dot{\delta}_i(t_f) \quad (2.44)$$

$$\ddot{q}_i(0) = \ddot{\lambda}_i(0) + \ddot{\delta}_i(0), \quad \ddot{q}_i(t_f) = \ddot{\lambda}_i(t_f) + \ddot{\delta}_i(t_f) \quad (2.45)$$

nas quais  $t_f = \frac{2\pi}{\omega_0}$  é o período.

Com as Equações (2.43), (2.44) e (2.45), é possível definir os coeficientes do polinômio  $\lambda_{ij}$  em função dos coeficientes de Fourier e das condições de contorno. Se estas forem predefinidas, a trajetória de cada junta é definida por apenas  $N$  parâmetros.

Para se obter a trajetória de identificação, executa-se um algoritmo de otimização sobre a Equação (2.41), para todas as juntas simultaneamente, minimizando o número de condição  $Cond(W)$  e verificando se as restrições de movimento no espaço das juntas e no espaço cartesiano estão sendo respeitadas.

#### 2.4.4 Desvio padrão dos parâmetros identificados

Pode-se utilizar os desvios padrões dos parâmetros identificados como uma forma de avaliar a qualidade da identificação.

Os desvios padrões dos parâmetros estimados são calculados assumindo que  $W$  é determinístico e que  $\rho$  é um ruído aditivo com média nula e desvio padrão  $\sigma_\rho$ . Uma estimativa não enviesada de  $\sigma_\rho$  é dada por [12]

$$\sigma_\rho^2 = \frac{\|T - W\hat{\theta}\|^2}{r - c} \quad (2.46)$$

em que  $r$  e  $c$  são as dimensões de  $W$ .

A matriz de covariância do erro de estimação é dada por

$$C_{\hat{\theta}} = \sigma_\rho^2 (W^T W)^{-1} \quad (2.47)$$

O desvio padrão do parâmetro dinâmico  $i$  é dado por

$$\sigma_{\hat{\theta}_i} = \sqrt{C_{\hat{\theta}}(i, i)} \quad (2.48)$$

## 2.5 Estimação da força de interação

Ao observar as Equações (2.30) e (2.33) fica evidente a contribuição da força de interação exercida sobre o efetuador terminal na dinâmica do robô.

A forma escolhida para obter essa força sem medição foi o uso de um estimador de estados, mais especificamente do Filtro de Kalman Unscented (UKF).

### 2.5.1 Filtro de Kalman

O filtro de Kalman é um estimador de estados que produz uma estimativa ótima, onde o valor médio da soma dos erros de estimação possui um valor mínimo [18]. Entretanto, esse fato é verdadeiro de acordo com certas condições. As mais importantes são: o sistema deve ser linear e as variáveis devem apresentar comportamento gaussiano com relação ao ruído.

O filtro de Kalman em sua forma mais simples e em tempo discreto consiste em um algoritmo recursivo que alterna entre fases de predição e atualização. Na fase de predição, considera-se a estimativa do instante anterior e a dinâmica do sistema para gerar a estimativa do instante atual. Essa é a estimativa *a priori*, ou antes da medição. Na fase de atualização, o filtro toma a estimativa *a priori* e o valor da medição e gera a estimativa *a posteriori*, que é a estimativa definitiva do instante atual [19].

Para entender melhor o algoritmo, considere um sistema dinâmico descrito por

$$\begin{aligned}x_k &= Ax_{k-1} + w \\y_k &= Cx_k + v\end{aligned}\tag{2.49}$$

em que  $w$  é o ruído do processo e  $v$  é o ruído de medição, e eles possuem matrizes de covariância  $Q$  e  $R$ , respectivamente.

A fase de predição consiste em

$$\hat{x}_k^- = A\hat{x}_{k-1} \quad \text{estimativa } a \text{ priori do estado} \tag{2.50}$$

$$P_k^- = AP_{k-1}A^T + Q \quad \text{covariância do erro de estimação de } \hat{x}_k^- \tag{2.51}$$

$$\hat{y}_k^- = C\hat{x}_k^- \quad \text{estimativa } a \text{ priori da saída} \tag{2.52}$$

Durante a fase de atualização é feito

$$\mathcal{K} = P_k^- C^T (C P_k^- C^T + R)^{-1} \quad \text{cálculo do ganho de Kalman} \tag{2.53}$$

$$\hat{x}_k = \hat{x}_k^- + \mathcal{K} (y_k - \hat{y}_k^-) \quad \text{estimativa } a \text{ posteriori do estado} \tag{2.54}$$

$$P_k = (I - \mathcal{K}C) P_k^- \quad \text{covariância do erro de estimação de } \hat{x}_k \tag{2.55}$$

O ganho de Kalman é continuamente refinado de acordo com as covariâncias dos ruídos e, como pode ser visto na Equação (2.53), determina quanto é valorizada a estimativa *a priori* quando confrontada com a medição. Em outras palavras, ele representa a relação entre a confiança no modelo e a confiança na medição.

Embora o filtro de Kalman seja simples e efetivo, a condição de que o sistema observado deve ser linear é bastante limitante. No caso deste trabalho, o modelo dinâmico do UR3 é não linear e, portanto, não pode ser observado por um filtro de Kalman.

O Filtro de Kalman Extendido (EKF) é uma alternativa popular ao filtro de Kalman que pode ser aplicado a sistemas não-lineares. Ele lida com sistemas não-lineares fazendo uma linearização de primeira ordem da dinâmica do sistema. Isso torna possível a aplicação das equações do filtro de Kalman, mas a linearização é uma simplificação que se torna uma fonte de erro e também é necessário calcular os Jacobianos das funções não-lineares, o que pode ser uma tarefa árdua caso estas sejam demasiado complexas (como no presente caso).

Outra alternativa, ainda baseada no Filtro de Kalman, é o Filtro de Kalman Unscented (UKF). Esse filtro é capaz de capturar a média e covariância das variáveis gaussianas acuradamente até a 3ª ordem para qualquer não-linearidade, sem a necessidade de calcular Jacobianos. Por causa dessas características, esse foi o filtro escolhido, e o seu funcionamento será explicado nas seções a seguir.

## 2.5.2 Transformada Unscented

A Transformada Unscented é a característica principal do Filtro de Kalman Unscented. Ela é a responsável por lidar com os efeitos das não-linearidades sobre as Funções de Distribuição de Probabilidade (FDPs) das variáveis gaussianas.

O grande problema de sistemas não-lineares é a dificuldade em transformar uma FDP através de uma função não-linear. A propagação da FDP pela linearização de tal função, como é feito no Filtro de Kalman Extendido, é muitas vezes uma aproximação ruim da transformação real. A Transformada Unscented aborda esse problema com base em dois princípios [19]:

1. É simples e fácil aplicar uma transformação não-linear em um único ponto (em vez de aplicar na FDP inteira); e
2. É possível encontrar um conjunto de pontos no espaço de estados cuja FDP da amostra aproxime a FDP verdadeira do vetor de estados.

Ao combinar esses dois princípios, é possível definir o procedimento da Transformada Unscented.

Toma-se um vetor  $x$ , do qual é conhecida a média  $\bar{x}$  e a covariância  $P$ . Em seguida, encontra-se um conjunto determinístico de vetores, chamados pontos *sigma*, cuja média e covariância são iguais a  $\bar{x}$  e  $P$ . Então, aplica-se a função não linear  $y = f(x)$  em cada um dos pontos sigma para obter os vetores transformados. A ideia da Transformada Unscented é que a média e covariância destes vetores transformados serão boas estimativas da média e covariância reais de  $y$  [19].

Na implementação do Filtro de Kalman Unscented, escolhe-se os pontos sigma de uma maneira determinística para que, juntamente com um vetor de pesos para a média e para a covariância, possa-se representar a FDP do vetor de estados. Para um vetor de estados de tamanho  $L$ , primeiro

define-se os vetores de pesos, cada um com  $2L + 1$  elementos, da seguinte forma [13]

$$W_0^m = \frac{\lambda}{L + \lambda} \quad (2.56)$$

$$W_0^c = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \quad (2.57)$$

$$W_i^m = W_i^c = \frac{1}{2(L + \lambda)}, \quad i = 1, \dots, 2L \quad (2.58)$$

em que  $\alpha \leq 1$  é um parâmetro que determina o espalhamento dos pontos sigma em torno da média,  $\beta$  é usado para incorporar o conhecimento anterior sobre a distribuição de  $x$  (para distribuições gaussianas utiliza-se o valor ótimo  $\beta = 2$ ) e  $\lambda = L(\alpha^2 - 1)$ .

Considerando esses pesos, e definindo  $\eta = \sqrt{L + \lambda}$ , os pontos sigmas são escolhidos de acordo com

$$\mathcal{X}_0 = \bar{x} \quad (2.59)$$

$$\mathcal{X}_i = \bar{x} + (\eta\sqrt{P})_i, \quad i = 1, \dots, L \quad (2.60)$$

$$\mathcal{X}_i = \bar{x} - (\eta\sqrt{P})_{i-L}, \quad i = L + 1, \dots, 2L \quad (2.61)$$

em que  $(\eta\sqrt{P})_i$  é a coluna  $i$  da raiz quadrada da matriz, sendo essa raiz quadrada definida como a decomposição de Cholesky  $chol(A) = B \rightarrow B^T B = A$ .

### 2.5.3 Filtro de Kalman Unscented

O algoritmo do Filtro de Kalman Unscented é análogo ao do Filtro de Kalman convencional (Seção 2.5.1), com a adição da Transformada Unscented. Considere um sistema da forma

$$\begin{aligned} x_{k+1} &= F(x_k) + w \\ y_k &= H(x_k) + v \end{aligned} \quad (2.62)$$

sendo  $F$  e  $H$  transformações não-lineares,  $w$  e  $v$  os ruídos do processo e medição e  $Q$  e  $R$  as respectivas matrizes de covariância.

A fase de predição do UKF é definida como

$$\mathcal{X}_{k-1} = \left[ \hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} \pm \eta\sqrt{\mathbf{P}_{k-1}} \right] \quad \text{cálculo dos pontos sigma} \quad (2.63)$$

$$\mathcal{X}_k = F(\mathcal{X}_{k-1}) \quad \text{transformação dos pontos sigma} \quad (2.64)$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^m \mathcal{X}_{k,i} \quad \text{estimativa } a \text{ priori do estado} \quad (2.65)$$

$$P_k^- = \sum_{i=0}^{2L} W_i^c \left[ \mathcal{X}_{k,i} - \hat{\mathbf{x}}_k^- \right] \left[ \mathcal{X}_{k,i} - \hat{\mathbf{x}}_k^- \right]^T + Q \quad \text{covariância do erro de estimação de } \hat{\mathbf{x}}_k^- \quad (2.66)$$

$$\mathcal{Y}_k = H(\mathcal{X}_k) \quad \text{medição dos pontos sigma transformados} \quad (2.67)$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^m \mathcal{Y}_{k,i} \quad \text{estimativa } a \text{ priori da saída} \quad (2.68)$$

A fase de atualização consiste nas operações

$$P_{yy} = \sum_{i=0}^{2L} W_i^c \left[ \mathcal{Y}_{k,i} - \hat{\mathbf{y}}_k^- \right] \left[ \mathcal{Y}_{k,i} - \hat{\mathbf{y}}_k^- \right]^T + R \quad \text{covariância do erro de estimação de } \hat{\mathbf{y}}_k^- \quad (2.69)$$

$$P_{xy} = \sum_{i=0}^{2L} W_i^c \left[ \mathcal{X}_{k,i} - \hat{\mathbf{x}}_k^- \right] \left[ \mathcal{Y}_{k,i} - \hat{\mathbf{y}}_k^- \right]^T \quad \text{covariância entre os erros de } \hat{\mathbf{x}}_k^- \text{ e } \hat{\mathbf{y}}_k^- \quad (2.70)$$

$$\mathcal{K} = P_{xy} P_{yy}^{-1} \quad \text{cálculo do ganho de Kalman} \quad (2.71)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathcal{K} \left( y_k - \hat{\mathbf{y}}_k^- \right) \quad \text{estimativa } a \text{ posteriori do estado} \quad (2.72)$$

$$P_k = P_k^- - \mathcal{K} P_{yy} \mathcal{K}^T \quad \text{covariância do erro de estimação de } \hat{\mathbf{x}}_k \quad (2.73)$$

Percebe-se que o algoritmo é equivalente ao filtro simples, com algumas operações adicionadas para acomodar a Transformada Unscented.

O UKF como foi apresentado seria suficiente para o propósito deste trabalho. Entretanto, ao implementá-lo computacionalmente, existe o risco de, devido à limitação na precisão numérica, a matriz  $P$  deixar de ser semi-definida positiva, o que torna impossível a obtenção da sua raiz quadrada através da decomposição de Cholesky. Para evitar esse problema, van der Merwe e Wan [14] propuseram uma variação do UKF, chamada Square-Root Unscented Kalman Filter (SRUKF).

## 2.5.4 Square-Root Unscented Kalman Filter

A cada iteração do Filtro de Kalman Unscented, durante o cálculo dos pontos sigma, é necessário computar a raiz quadrada da matriz de covariância do estado,  $P$ , o que é feito através da decomposição de Cholesky, onde  $chol(P) = S \rightarrow S^T S = P$ . Ainda que raiz quadrada de  $P$  seja uma parte fundamental do UKF, a matriz completa de covariância  $P$  é a que é atualizada recursivamente. Na implementação do Square-Root Unscented Kalman Filter, a matriz  $S$  é propagada diretamente, evitando a necessidade de calcular a fatoração toda iteração. Além disso, a estabi-

lidade numérica do SRUKF é superior à do UKF e há a garantia de que a matriz de covariância (implícita, no caso dessa implementação) é positiva semi-definida [14].

A implementação do SRUKF utiliza três técnicas da álgebra linear (usando o MATLAB como base):

1. **Decomposição QR:** A decomposição QR de uma matriz  $A \in \mathbb{R}^{L \times N}$  é dada por  $A^T = QR$ , onde  $Q \in \mathbb{R}^{N \times N}$  é ortogonal,  $R \in \mathbb{R}^{N \times L}$  é triangular superior e  $N \geq L$ . A parte triangular superior de  $R$ ,  $\tilde{R}$ , é a transposta do fator de Cholesky de  $P = AA^T$ , isto é,  $\tilde{R} = S^T$ , de forma que  $\tilde{R}^T \tilde{R} = AA^T$ . Na apresentação do algoritmo, será usada a notação  $qr\{\cdot\}$  para se referir à decomposição QR de uma matriz onde  $\tilde{R}$  é retornado.
2. **Atualização do fator de Cholesky:** Se  $S$  é o fator de Cholesky original de  $P = AA^T$ , então o fator de Cholesky do update (ou downdate) de rank 1  $P \pm \sqrt{\nu}uu^T$  é dado por  $S = cholupdate\{S, u, \pm\nu\}$ , onde  $cholupdate$  é a função do MATLAB. Se  $u$  for uma matriz e não um vetor, então o resultado é dado por  $M$  atualizações sucessivas do fator de Cholesky usando as  $M$  colunas de  $u$ .
3. **Mínimos quadrados:** A solução da equação  $(AA^T)x = A^Tb$  também corresponde à solução do problema de mínimos quadrados sobredeterminado  $Ax = b$ . Esse cálculo é resolvido de maneira eficiente usando uma decomposição QR com escalonamento, o que é feito através do operador  $\backslash$  no MATLAB.

Incorporando essas técnicas ao algoritmo do UKF, é estabelecido o SRUKF.

O algoritmo é, iniciando pela predição

$$\mathcal{X}_{k-1} = \begin{bmatrix} \hat{\mathbf{x}}_{k-1} & \hat{\mathbf{x}}_{k-1} \pm \eta S_{k-1} \end{bmatrix} \quad \text{cálculo dos pontos sigma} \quad (2.74)$$

$$\mathcal{X}_k = F(\mathcal{X}_{k-1}) \quad \text{transformação dos pontos sigma} \quad (2.75)$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^m \mathcal{X}_{k,i} \quad \text{estimativa } a \text{ priori do estado} \quad (2.76)$$

$$S_k^- = qr\left\{ \left[ \sqrt{W_i^c} (\mathcal{X}_{k,i} - \hat{\mathbf{x}}_k^-) \quad \sqrt{Q} \right] \right\} \quad \text{cálculo do fator de Choleski, } i = 1, \dots, 2L \quad (2.77)$$

$$S_k^- = cholupdate\left\{ S_k^-, \mathcal{X}_{k,0} - \hat{\mathbf{x}}_k^-, W_0^c \right\} \quad \text{atualização do fator de Cholesky} \quad (2.78)$$

$$\mathcal{Y}_k = H(\mathcal{X}_k) \quad \text{medição dos pontos sigma transformados} \quad (2.79)$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^m \mathcal{Y}_{k,i} \quad \text{estimativa } a \text{ priori da saída} \quad (2.80)$$

A atualização é definida por

$$S_y = qr \left\{ \left[ \sqrt{W_i^c} \left( \mathcal{Y}_{k,i} - \hat{y}_k^- \right) \quad \sqrt{R} \right] \right\} \quad \text{cálculo do fator de Choleski, } i = 1, \dots, 2L \quad (2.81)$$

$$S_y = cholupdate \left\{ S_y, \mathcal{Y}_{k,0} - \hat{y}_k^-, W_0^c \right\} \quad \text{atualização do fator de Cholesky} \quad (2.82)$$

$$P_{xy} = \sum_{i=0}^{2L} W_i^c \left[ \mathcal{X}_{k,i} - \hat{x}_k^- \right] \left[ \mathcal{Y}_{k,i} - \hat{y}_k^- \right]^T \quad \text{covariância entre os erros de } \hat{x}_k^- \text{ e } \hat{y}_k^- \quad (2.83)$$

$$\mathcal{K} = (P_{xy} / S_y^T) / S_y^T \quad \text{cálculo do ganho de Kalman} \quad (2.84)$$

$$\hat{x}_k = \hat{x}_k^- + \mathcal{K} \left( y_k - \hat{y}_k^- \right) \quad \text{estimativa } a \text{ posteriori} \text{ do estado} \quad (2.85)$$

$$U = \mathcal{K} S_y \quad (2.86)$$

$$S_k = cholupdate \left\{ S_k^-, U, -1 \right\} \quad \text{covariância do erro de estimação de } \hat{x}_k \quad (2.87)$$

Esse algoritmo é o que foi implementado para o uso neste trabalho.

## 2.6 ROS

Para realizar a comunicação com o robô, utilizou-se o ROS.

De acordo com a documentação oficial [20], o ROS é um sistema operacional em código aberto para robôs. Ele fornece serviços típicos de sistemas operacionais, como abstração de hardware, controle de dispositivos de baixo nível, implementação de operações frequentes, passagem de mensagem entre processos e gerenciamento de pacotes. Ele também provê ferramentas e bibliotecas para manipular e compilar código através de vários computadores.

Ao mesmo tempo que ele oferece acesso de baixo nível, o ROS também permite abstrair uma série de operações básicas em um sistema robotizado, oferecendo formatos e canais predefinidos que possibilitam uma filosofia quase que *plug and play* para a adição de novos programas e robôs. Isso é possível graças à grande modularidade de sistemas ROS.

Embora o ROS seja bem complexo e forneça muitas ferramentas e opções, para o escopo deste trabalho basta entender apenas algumas partes dele:

- **Nós:** nós são processos que executam computações. O ROS foi projetado para ser altamente modular. Um sistema de controle de um robô geralmente consiste de muitos nós. Por exemplo, um nó controla um sensor de distância laser, um nó controla os motores das rodas, um nó executa a localização, um nó faz o planejamento de trajetória, um nó provê uma visualização gráfica do sistema, etc. Um nó ROS é escrito usando uma biblioteca como *roscpp* ou *rospy*.
- **Mestre (*Master*):** o ROS Master fornece registro de nomes e busca no resto do grafo de computação. Sem o ROS Master os nós não conseguiriam encontrar uns aos outros, trocar mensagens ou chamar serviços.

- **Mensagens:** os nós se comunicam através da troca de Mensagens. Uma mensagem é simplesmente uma estrutura de dados, composta de campos. Tipos primitivos padrão (inteiro, ponto flutuante, booleano, etc.) são suportados, assim como arrays de tipos primitivos. Mensagens podem conter estruturas e arrays arbitrariamente aninhados (semelhantemente a registros em C).
- **Tópicos:** mensagens são encaminhadas por um sistema de transporte com semântica publicação/inscrição (publish/subscribe). Um nó envia uma mensagem publicando-a em um dado tópico. O tópico é o nome usado para identificar o conteúdo da mensagem. Um nó interessado em um tipo específico de informação irá se inscrever em um tópico apropriado. Podem existir muitos publishers e subscribers em um mesmo tópico, assim como um nó pode publicar e se inscrever em vários tópicos. De maneira geral, publishers e subscribers não estão cientes da existência um do outro. A ideia é separar a produção e o consumo da informação. Cada tópico tem um nome e qualquer um pode se conectar a este tópico para enviar ou receber mensagens, desde que estas sejam do tipo correto.
- **Bags:** Bags são um formato para armazenamento e recuperação de dados no ROS. Elas são um mecanismo importante para guardar dados, como dados de sensores, que podem ser difíceis de coletar, mas necessários para o desenvolvimento e teste de algoritmos.

O ROS Master age como um serviço de nomes no grafo de computação. Ele guarda informações de registro de tópicos e serviços para os nós. Os nós se comunicam com o Master para reportar as suas informações de registro. Através dessa comunicação, eles recebem informação dos outros nós registrados, e fazem conexões de acordo com o seu propósito. O Master também faz chamadas para esses nós quando as informações de registro mudam, permitindo que os nós criem conexões dinamicamente quando novos nós são executados.

Os nós se conectam diretamente, com o Master fornecendo apenas informação de busca. Nós que se inscrevem em um tópico vão requisitar conexões com nós que publicam naquele tópico, e vão estabelecer tais conexões de acordo com protocolos estabelecidos. O protocolo mais comum é o ROSTCP, que utiliza sockets TCP/IP padrão.

Essa arquitetura permite operação desconjunta, onde os nomes são os meios através dos quais sistemas maiores e mais complexos são construídos. Nomes têm uma função fundamental no ROS: nós, tópicos, serviços e parâmetros tem nomes. Todas as bibliotecas de cliente ROS suportam remapeamento de nomes via linha de comando, fazendo com que um programa compilado possa ser reconfigurado em tempo de execução para operar em uma topologia diferente de grafo de computação.

A Figura 2.2 fornece uma explicação visual simplificada do funcionamento de um sistema ROS típico. Os nós podem se comunicar através de mensagens advindas de publicações e inscrições em tópicos. As mensagens publicadas em um tópico podem ser coletadas em uma Bag. Por cima de tudo, o Mestre possibilita que os nós conheçam as informações relevantes (como por exemplo os tópicos disponíveis).

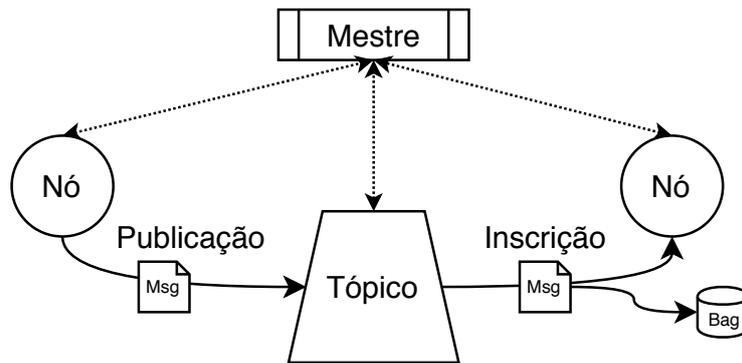


Figura 2.2: Diagrama simplificado de um sistema ROS (“mensagem” abreviada como “Msg”)

## 2.7 UR3



Figura 2.3: Manipulador robótico UR3 [2].

O robô no qual será implementada a estimação de força é o manipulador robótico UR3, da Universal Robotics. Ele é descrito pelo fabricante como um robô colaborativo de mesa capaz de realizar tarefas leves de montagem e automação da bancada de trabalho. Ele possui 6 juntas rotacionais e é capaz de carregar um peso de 3kg. Devido ao seu tamanho diminuto, ele é ideal para aplicações simples.

O UR3 pode ser programado utilizando o ambiente Polyscope, que possibilita uma forma de programação simples e intuitiva. Isso pode ser feito utilizando o *teach pendant* conectado ao controlador do robô ou através do download do software do fabricante. Outra maneira de se comunicar com o robô e enviar comandos é através da linguagem URScript. [21]

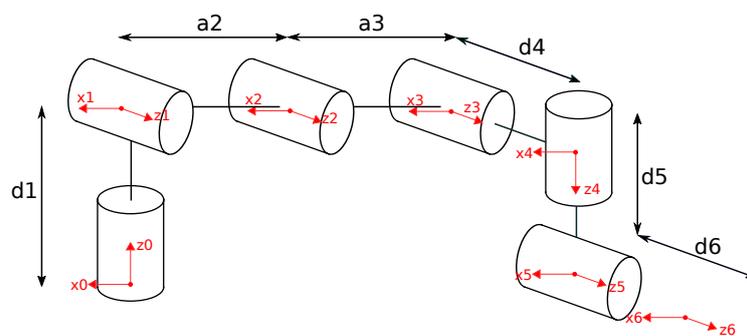
A linguagem URScript é uma linguagem de programação adotada pela Universal Robots para programar os seus robôs. Ela possui uma sintaxe similar a Python e possui variáveis, tipos, controle de fluxo e ferramentas para monitorar e controlar entrada e saída e os movimentos do robô.

A principal vantagem em se utilizar URScript em vez de Polyscape é que essa linguagem permite um acesso a funções de mais baixo nível do robô, fornecendo um grau maior de controle e informação. Outra vantagem é que pode-se programar o UR3 remotamente, utilizando sockets TCP/IP para enviar as instruções e programas para o controlador do manipulador. Por causa disso, é possível integrar a programação e interação eficientemente ao ambiente ROS.

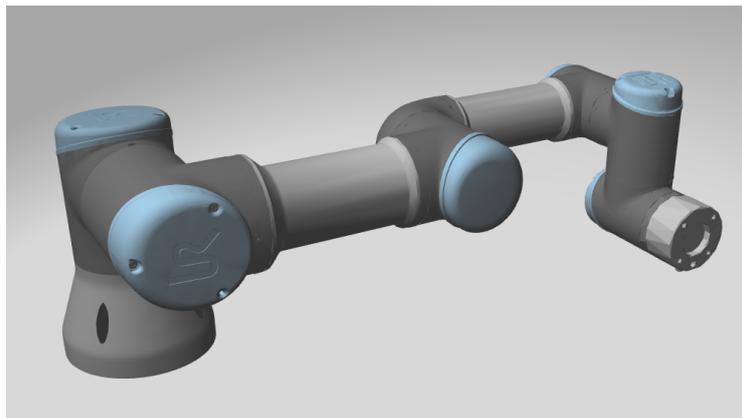
## Capítulo 3

# Modelagem dinâmica, Identificação e Estimação de força

### 3.1 Cinemática direta



(a) Diagrama esquemático do UR3.



(b) Modelo do UR3 na mesma pose do diagrama.

Figura 3.1: Sistemas de coordenadas dos links do UR3, atribuídos segundo a convenção de Denavit-Hartenberg.

Seguindo a convenção de Denavit-Hartenberg (Seção 2.1.2), atribuiu-se os sistemas de coord-

nadas dos links do UR3. A Figura 3.1 mostra um diagrama que ilustra os referidos sistemas de coordenadas. Os cilindros na Figura 3.1a correspondem às juntas do robô, e os eixos principais dos cilindros correspondem aos eixos de rotação dos atuadores.

Com o auxílio da documentação oficial do fabricante (encontrada em [22]), foram extraídos os parâmetros de DH, os quais podem ser encontrados na Tabela 3.1. Note que os valores de  $\theta$ ,  $q_i$ ,  $i = 1, \dots, 6$  são as próprias posições das juntas.

Os sistemas de coordenadas atribuídos (e, portanto, os parâmetros de DH associados) são ligeiramente diferentes dos fornecidos pelo fabricante, e foram atribuídos desse modo para evitar o surgimento de parâmetros negativos. No entanto, ambos os modelos são matematicamente equivalentes e respeitam a convenção de DH.

Tabela 3.1: Parâmetros de Denavit-Hartenberg para o UR3

$\theta$	$\mathbf{d}$	$\mathbf{a}$	$\alpha$
$q_1$	0.1519	0	$-\frac{\pi}{2}$
$q_2$	0	0.24365	0
$q_3$	0	0.21325	0
$q_4$	0.11235	0	$-\frac{\pi}{2}$
$q_5$	0.08535	0	$\frac{\pi}{2}$
$q_6$	0.0819	0	0

## 3.2 Modelo dinâmico

Como a estratégia de modelagem dinâmica escolhida foi baseada no modelo de Euler-Lagrange, foi necessário encontrar a forma analítica das equações da Seção 2.3. Para alcançar esse fim, foi necessário utilizar um *software* CAS (*computer algebra system*).

A primeira escolha foi o MATLAB, devido à familiaridade e versatilidade. No entanto, ao longo do desenvolvimento do modelo dinâmico, as expressões simbólicas começaram a ficar muito grandes e complexas, e a *Toolbox* Simbólica do MATLAB se mostrou insuficiente para a realização dessa computação.

Na busca por alternativas, foi encontrado o Wolfram Mathematica. Embora possuísse uma sintaxe de programação diferente, esse *software* se mostrou capaz de calcular todas as equações algébricas.

Para a computação numérica, entretanto, o MATLAB ainda foi a escolha principal. As expressões simbólicas simplificadas foram exportadas do Mathematica para o MATLAB, o qual conseguiu realizar os cálculos numéricos sem dificuldades.

### 3.2.1 Modelagem

Primeiro foram calculadas as matrizes de transformação homogênea entre os sistemas de coordenadas das juntas  $H_1^0(q)$ ,  $H_2^1(q)$ ,  $H_3^2(q)$ ,  $H_4^3(q)$ ,  $H_5^4(q)$  e  $H_6^5(q)$ , de acordo com a Equação (2.3), utilizando os parâmetros de Denavit-Hartenberg da Tabela 3.1. Essas matrizes foram multiplicadas de acordo com a Equação (2.2) para obter as matrizes de transformação do sistema de coordenadas inercial para cada sistema de coordenada das juntas. Foram extraídos dessas matrizes as matrizes de rotação  $R_i^0$  e os vetores de translação  $o_i^0$  de acordo com a Equação (2.1).

$$\begin{aligned}
H_1^0(q) &= H_1^0(q) \\
H_2^0(q) &= H_1^0(q)H_2^1(q) \\
H_3^0(q) &= H_1^0(q)H_2^1(q)H_3^2(q) \\
H_4^0(q) &= H_1^0(q)H_2^1(q)H_3^2(q)H_4^3(q) \\
H_5^0(q) &= H_1^0(q)H_2^1(q)H_3^2(q)H_4^3(q)H_5^4(q) \\
H_6^0(q) &= H_1^0(q)H_2^1(q)H_3^2(q)H_4^3(q)H_5^4(q)H_6^5(q)
\end{aligned} \tag{3.1}$$

Em seguida foram computadas as posições dos centros de massa. Para tanto, definiu-se os vetores de posição de cada centro de massa  $cm_i$  com respeito ao sistema de coordenadas do seu link como variáveis. Esses vetores foram então expressos no sistema de coordenadas inercial usando as matrizes de transformação calculadas anteriormente.

$$\begin{aligned}
r1(q) &= o_1^0(q) + R_1^0(q) \cdot cm1 \\
r2(q) &= o_2^0(q) + R_2^0(q) \cdot cm2 \\
r3(q) &= o_3^0(q) + R_3^0(q) \cdot cm3 \\
r4(q) &= o_4^0(q) + R_4^0(q) \cdot cm4 \\
r5(q) &= o_5^0(q) + R_5^0(q) \cdot cm5 \\
r6(q) &= o_6^0(q) + R_6^0(q) \cdot cm6
\end{aligned} \quad \text{em que} \quad \begin{cases} cm1 = [rx1, ry1, rz1]^T \\ cm2 = [rx2, ry2, rz2]^T \\ cm3 = [rx3, ry3, rz3]^T \\ cm4 = [rx4, ry4, rz4]^T \\ cm5 = [rx5, ry5, rz5]^T \\ cm6 = [rx6, ry6, rz6]^T \end{cases} \tag{3.2}$$

Foram definidas variáveis para as massas dos links ( $mi$ ,  $i = 1, \dots, 6$ ), que juntamente com as posições dos centros de massa permitiram calcular a energia potencial do robô (Equação (2.19)). A expressão da energia potencial foi prontamente derivada com respeito a cada variável de junta para obter o vetor  $g(q)$  da Equação (2.26).

$$g(q) = \frac{\partial P(q)}{\partial q_1} + \frac{\partial P(q)}{\partial q_2} + \frac{\partial P(q)}{\partial q_3} + \frac{\partial P(q)}{\partial q_4} + \frac{\partial P(q)}{\partial q_5} + \frac{\partial P(q)}{\partial q_6} \quad \text{em que} \quad P(q) = \sum_{i=1}^6 mi \cdot g \cdot ri(q) \tag{3.3}$$

sendo  $g$  o vetor da aceleração da gravidade, considerando 9.81 como a sua magnitude.

Foram então calculados os Jacobianos dos centros de massa usando a Equação (2.13).

$$\begin{aligned}
J_{r1}(q) &= \begin{bmatrix} \frac{\partial r1(q)}{\partial q_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ z_0^0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
J_{r2}(q) &= \begin{bmatrix} \frac{\partial r2(q)}{\partial q_1} & \frac{\partial r2(q)}{\partial q_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ z_0^0 & z_0^1(q) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
J_{r3}(q) &= \begin{bmatrix} \frac{\partial r3(q)}{\partial q_1} & \frac{\partial r3(q)}{\partial q_2} & \frac{\partial r3(q)}{\partial q_3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ z_0^0 & z_0^1(q) & z_0^2(q) & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
J_{r4}(q) &= \begin{bmatrix} \frac{\partial r4(q)}{\partial q_1} & \frac{\partial r4(q)}{\partial q_2} & \frac{\partial r4(q)}{\partial q_3} & \frac{\partial r4(q)}{\partial q_4} & \mathbf{0} & \mathbf{0} \\ z_0^0 & z_0^1(q) & z_0^2(q) & z_0^3(q) & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
J_{r5}(q) &= \begin{bmatrix} \frac{\partial r5(q)}{\partial q_1} & \frac{\partial r5(q)}{\partial q_2} & \frac{\partial r5(q)}{\partial q_3} & \frac{\partial r5(q)}{\partial q_4} & \frac{\partial r5(q)}{\partial q_5} & \mathbf{0} \\ z_0^0 & z_0^1(q) & z_0^2(q) & z_0^3(q) & z_0^4(q) & \mathbf{0} \end{bmatrix} \\
J_{r6}(q) &= \begin{bmatrix} \frac{\partial r6(q)}{\partial q_1} & \frac{\partial r6(q)}{\partial q_2} & \frac{\partial r6(q)}{\partial q_3} & \frac{\partial r6(q)}{\partial q_4} & \frac{\partial r6(q)}{\partial q_5} & \frac{\partial r6(q)}{\partial q_6} \\ z_0^0 & z_0^1(q) & z_0^2(q) & z_0^3(q) & z_0^4(q) & z_0^5(q) \end{bmatrix}
\end{aligned} \tag{3.4}$$

Então, criou-se variáveis para os tensores de inércia dos links ( $I_i$ ,  $i = 1, \dots, 6$ ). Essas variáveis, juntamente com as massas dos links, as matrizes de rotação e os Jacobianos dos centros de massa foram usadas para calcular a matriz de inércia  $D(q)$  de acordo com a Equação (2.17).

$$D(q) = \sum_{i=1}^6 m_i \cdot J_{r_{i_v}}(q)^T \cdot J_{r_{i_v}}(q) + J_{r_{i_\omega}}(q)^T \cdot R_i^0(q) \cdot I_i \cdot R_i^0(q)^T \cdot J_{r_{i_\omega}}(q) \tag{3.5}$$

A partir da matriz de inércia, foram gerados os símbolos de Christoffel conforme a Equação (2.24), os quais foram utilizados para computar as entradas da matriz  $C(q, \dot{q})$  de acordo com a Equação (2.27).

$$C(q, \dot{q}) = (c_{kj}) \in \mathbb{R}^{6 \times 6} \mid c_{kj} = \sum_{i=1}^6 \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i \tag{3.6}$$

De posse das matrizes  $D(q)$  e  $C(q, \dot{q})$  e do vetor  $g(q)$ , obteve-se a expressão para os torques nas juntas devido ao movimento do manipulador, calculando a Equação (2.26).

Definiu-se os coeficientes de atrito como variáveis, sendo os coeficientes do atrito viscoso  $\nu_{vi}$ ,  $i = 1, \dots, 6$  e os coeficientes do atrito de Coulomb  $\nu_{ci}$ ,  $i = 1, \dots, 6$ .

O Jacobiano do efetuador terminal foi calculado de acordo com a Equação (2.5), possibilitando obter a expressão do torque devido ao esforço externo conforme a Equação (2.29), utilizando variáveis para as componentes do esforço externo.

$$J = \begin{bmatrix} \frac{\partial o_6^0(q)}{\partial q_1} & \frac{\partial o_6^0(q)}{\partial q_2} & \frac{\partial o_6^0(q)}{\partial q_3} & \frac{\partial o_6^0(q)}{\partial q_4} & \frac{\partial o_6^0(q)}{\partial q_5} & \frac{\partial o_6^0(q)}{\partial q_6} \\ z_0^0 & z_0^1(q) & z_0^2(q) & z_0^3(q) & z_0^4(q) & z_0^5(q) \end{bmatrix} \tag{3.7}$$

Finalmente, obteve-se a expressão para a dinâmica direta conforme a Equação (2.33).

### 3.3 Identificação dos parâmetros dinâmicos

O modelo dinâmico inverso, representado pela Equação (2.30) (desconsiderando por hora a componente de força externa), foi reescrito de forma linear com respeito aos parâmetros dinâmicos, conforme a Equação (2.34). Isso foi feito no Mathematica usando as funções *Expand* e *Level*. A primeira função executa a propriedade distributiva da multiplicação sobre a adição, fazendo com que a expressão resultante apresente apenas somas de termos. A segunda função gera um array com esses termos. Esse array é então separado em uma multiplicação de parâmetros dinâmicos e coeficientes (que dependem das variáveis de juntas). Desse modo obteve-se a expressão analítica da matriz de observação.

Adotou-se a parametrização de Park (Equação (2.41)) para descrever a trajetória. Definiu-se as condições de contorno como sendo nulas e a ordem da série de Fourier como 5. Definiu-se as restrições das variáveis de juntas e o critério de otimização como sendo o número de condição da matriz  $W$ . A otimização foi feita utilizando a função *fmincon* do MATLAB, que utiliza um algoritmo *interior-point*. Foram geradas duas trajetórias, uma com número de condição  $4.6362 \cdot 10^5$  e a outra com  $6.3798 \cdot 10^4$ . Essas trajetórias foram chamadas de Trajetórias 1 e 2, respectivamente. Além disso, para efeito de comparação, utilizou-se uma outra trajetória, gerada anteriormente por outro aluno trabalhando no UR3, com um PRBS como referência de velocidade. Essa trajetória foi denominada Trajetória 0. As três trajetórias podem ser vistas nas Figuras 3.2, 3.3 e 3.4.

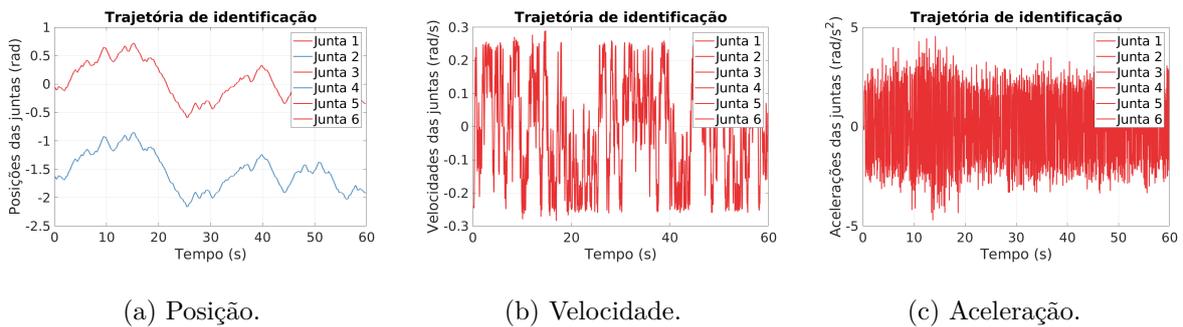


Figura 3.2: Trajetória de identificação 0.

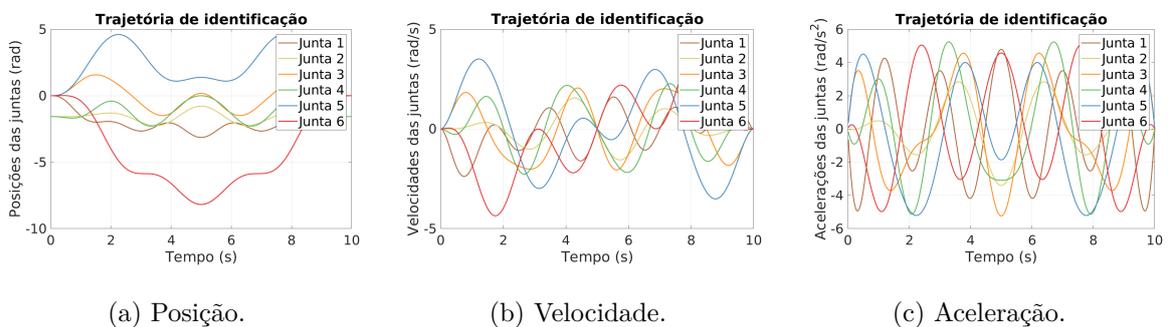


Figura 3.3: Trajetória de identificação 1.

Essas trajetórias foram passadas para o robô através de um programa de interface que será

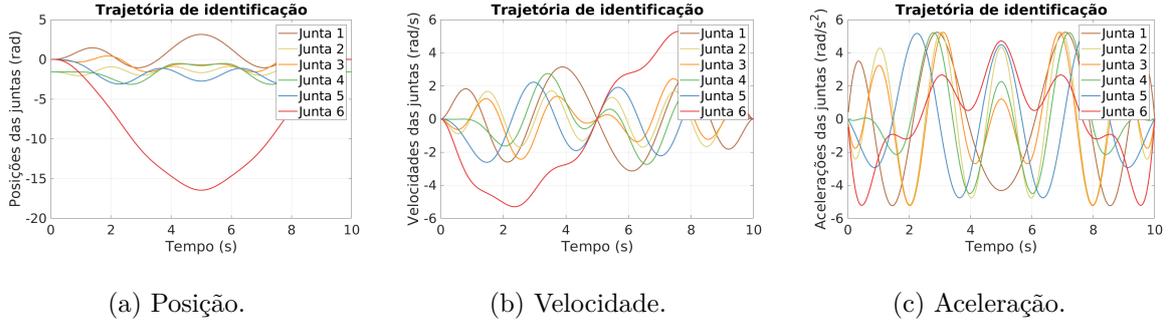


Figura 3.4: Trajetória de identificação 2.

discutido na Seção 3.5. Os dados da execução foram coletados em uma bag ROS.

De posse dos dados dos experimentos, derivou-se os sinais para se obter as acelerações e filtrou-se os torques para mitigar o ruído. Então foi feita a solução dos mínimos quadrados conforme a Equação (2.37).

### 3.4 Implementação do Filtro de Kalman Unscented

O Filtro de Kalman Unscented foi implementado no MATLAB, usando o algoritmo descrito na Seção 2.5.4. O estado  $x$  foi definido como as posições, velocidades, acelerações e os torques nas juntas, além da força no efetuador terminal. A medição foi definida como as posições, velocidades e torques nas juntas. A função de transição de estados foi definida da seguinte maneira

$$\begin{aligned}
 x_k &= F(x_{k-1}) \\
 \begin{bmatrix} q_k \\ \dot{q}_k \\ \ddot{q}_k \\ \tau_k \\ F_k \end{bmatrix} &= \begin{bmatrix} q_{k-1} + \dot{q}_{k-1} \cdot \Delta t \\ \dot{q}_{k-1} + \ddot{q}_{k-1} \cdot \Delta t \\ \ddot{q}_{fdyn}(q_{k-1}, \dot{q}_{k-1}, \tau_{k-1}, F_{k-1}) \\ \tau_{idyn}(q_{k-1}, \dot{q}_{k-1}, \ddot{q}_{k-1}, F_{k-1}) \\ F_{k-1} \end{bmatrix} \quad (3.8)
 \end{aligned}$$

com  $\ddot{q}_{fdyn}$  sendo a dinâmica direta (Equação (2.33)) e  $\tau_{idyn}$  a dinâmica inversa (Equação (2.30)).

A função de medição é dada por

$$y_k = H(x_k) = \begin{bmatrix} q_k \\ \dot{q}_k \\ \tau_k \end{bmatrix} \quad (3.9)$$

Assim, o espaço de estados fica definido como

$$x_k = F(x_{k-1}) + w \quad (3.10)$$

$$y_k = H(x_k) + v \quad (3.11)$$

em que  $w$  e  $v$  são os ruídos do processo e de medição. As matrizes de covariância  $Q$  e  $R$  associadas a esses ruídos foram calculadas como matrizes diagonais com as variâncias de cada sinal. Essas variâncias foram escolhidas da seguinte maneira:

- $Q$ : Foram escolhidas arbitrariamente variâncias pequenas para posição e velocidade, porque considera-se que as linearizações de 1ª ordem feitas na Equação (3.8) são suficientes. Para a aceleração e torque, utilizou-se as equações de dinâmica direta e inversa para calcular os valores de aceleração e torque, os quais foram subtraídos dos valores medidos (o torque medido filtrado e a aceleração derivada da posição), gerando sinais de “erro de modelagem”. A variância foi extraída desse sinal de erro. A variância da força no efetuador terminal foi escolhida arbitrariamente alta, uma vez que ela não está sendo modelada.
- $R$ : Aqui tomou-se as medições e aplicou-se sobre elas filtros para diminuir o ruído. Subtraiu-se os valores originais dos filtrados e calculou-se as variâncias dessa diferença.

Foi criado também um script para plotar os gráficos dos estados estimados.

Implementou-se também uma outra versão do UKF, que funciona em tempo real. Esse programa, ainda feito no MATLAB, recebe dados de um tópico ROS, executa as iterações do filtro e plota o estado estimado e as medições em tempo real. Mais detalhes de como isso foi feito serão explicados na próxima seção.

### 3.5 Comunicação e controle do UR3 pelo ROS

Para se comunicar com o robô e controlá-lo, criou-se através do ROS programas em C++. O código principal, que cuida de estabelecer uma conexão e canal de comunicação com o UR3, foi feito por outro aluno e modificado apenas na medida do necessário. Os códigos implementados foram nós para a leitura e envio da trajetória do robô, contida em um arquivo csv, e subamostragem dos dados do robô.

O UR3 não permite controle de baixo nível. O nível mais inferior de abstração que se consegue acessar é através da linguagem URScript. A estratégia de comando é enviar os programas, escritos em URScript, para o robô através de um socket TCP, usando o ROS para fazer este envio. O robô suporta tanto o envio de instruções individuais como o envio de programas inteiros.

Com base nisso, foi criado um programa em URScript que se conecta a um socket TCP no computador rodando o ROS e continuamente envia os dados de posição e velocidade das juntas e correntes nos motores nessa porta. Esse programa também lê do socket a referência de velocidade a ser seguida. Isso tudo é feito simultaneamente através de threads. A Figura 3.5 ilustra como é feita essa comunicação.

Um nó ROS, chamado de *interface*, cuida de enviar esse programa ao robô (que é executado prontamente) e fazer a leitura e envio dos dados. Os dados lidos são colocados em uma mensagem, obtendo antes disso os torques a partir das correntes através de uma multiplicação pelas cons-

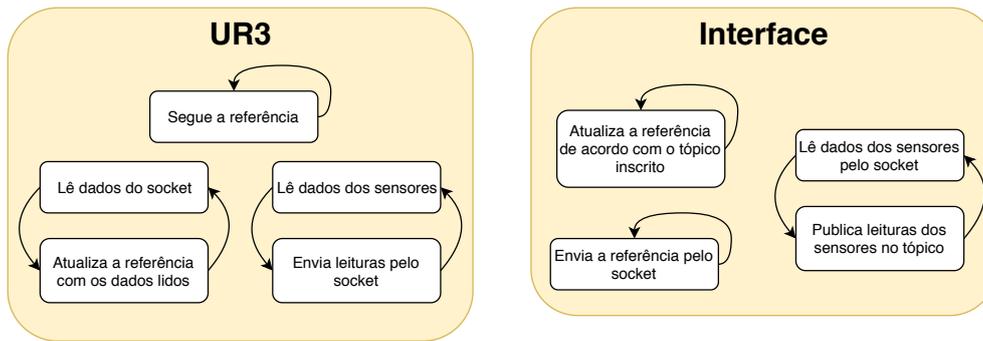


Figura 3.5: Diagrama mostrando o funcionamento da comunicação entre o computador (nó “Interface”) e o UR3

tantes de torque determinadas experimentalmente, que é publicada em um tópico apropriado. A referência de velocidade é lida de um tópico e enviada via socket TCP.

O nó que publica no tópico de referência de velocidade é o *send-trajectory*. Esse nó é chamado passando um arquivo csv e uma taxa de amostragem. Ele então lê os dados do csv, formata-os de maneira apropriada e os publica no tópico.

O último programa desenvolvido foi um subamostrador. Foi constatado que o Filtro de Kalman Unscented leva consideravelmente mais tempo para processar um estado do que o período de amostragem. Por causa disso, e para manter a sincronia temporal, optou-se por subamostrar os dados do robô na taxa que o UKF é capaz de processar. Entretanto, para não simplesmente descartar as amostras, decidiu-se utilizar uma média móvel exponencial. Dessa forma, alguma informação sobre as amostras passadas é preservada. Esse nó lê os dados na taxa do robô, atualiza a média exponencial e manda a média mais recente na taxa do UKF.

A toolbox de robótica do MATLAB é compatível com o ROS, o que possibilitou a implementação do UKF em tempo real. Conecta-se o programa do MATLAB com a rede ROS e inscreve-se no tópico contendo os dados.

# Capítulo 4

## Resultados

### 4.1 Identificação

Foram realizados três experimentos de identificação, usando as trajetórias geradas na Seção 3.3. As velocidades foram enviadas ao UR3 e os dados foram guardados em uma bag ROS. Devido às limitações mencionadas com respeito ao controle em baixo nível do UR3, foi necessário usar as funções do URScript para comandar o robô. Essas funções não permitem o controle total sobre posição, velocidade e aceleração, o que fez com que o perfil de aceleração das trajetórias fosse diferente do planejado, o que ocasionou um *drifting*, tanto na trajetória 1 como na trajetória 2. Esse comportamento faz sentido, uma vez que as trajetórias planejadas foram derivadas analiticamente, enquanto que as executadas de fato dependem da dinâmica de controle do robô. As Figuras 4.1 e 4.2 mostram esse fenômeno para a trajetória da junta 1. Todas as juntas apresentaram esse comportamento, mas escolheu-se mostrar apenas a junta 1 para ilustrar o efeito.

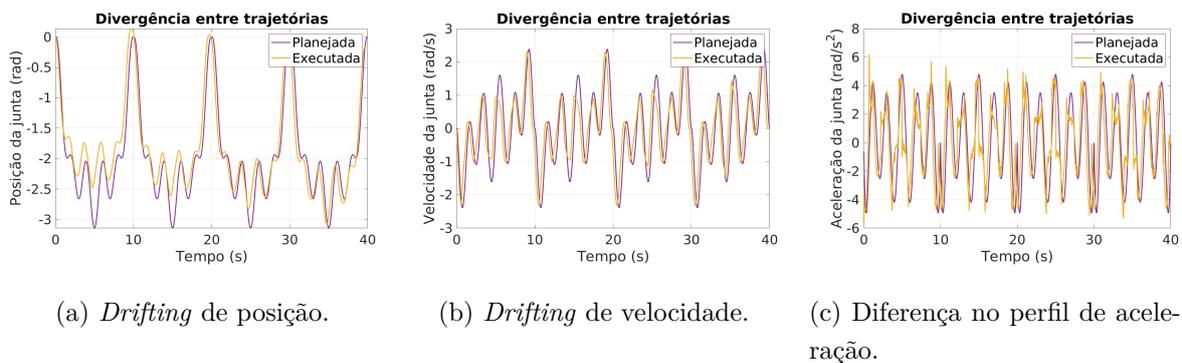


Figura 4.1: Diferenças entre a trajetória 1 planejada e a executada pelo UR3 para a junta 1.

As principais consequências dessa divergência são a alteração no número de condição da trajetória e perda da periodicidade no tempo. A primeira não tem um grande impacto, uma vez que os números de condição das trajetórias reais ainda possuem a mesma ordem de grandeza (como pode ser visto na Tabela 4.1, comparando com os valores teóricos na Seção 3.3). Entretanto, a segunda consequência faz com que não seja possível tomar uma média entre os períodos para diminuir o

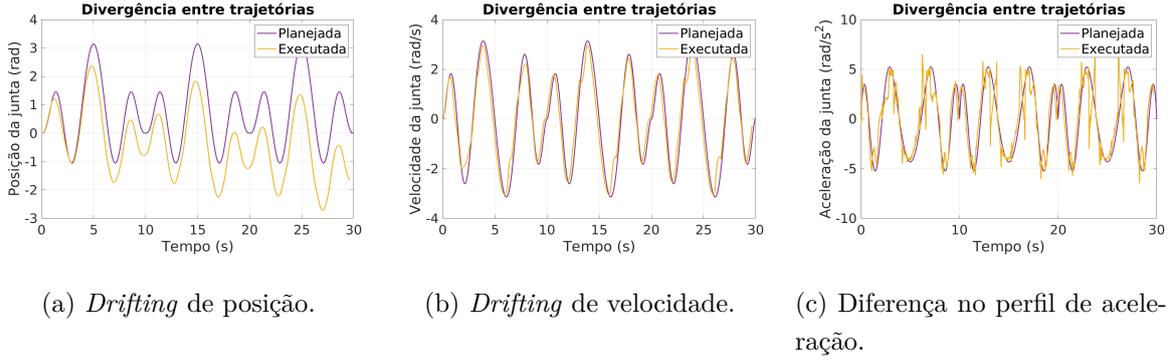


Figura 4.2: Diferenças entre a trajetória 2 planejada e a executada pelo UR3 para a junta 1.

Tabela 4.1: Números de condição das trajetórias de identificação executadas

Trajетória	Número de condição
0	$1.8633 \cdot 10^9$
1	$5.9157 \cdot 10^5$
2	$7.1726 \cdot 10^4$

efeito do ruído, o que era uma das principais motivações para essas trajetórias.

Para reduzir o ruído das medições de torque, os sinais foram filtrados com um filtro Butterworth de segunda ordem. Para obter as acelerações, utilizou-se um filtro derivativo FIR. A estimação dos parâmetros foi feita seguindo os passos explicados na Seção 2.4.1.

Antes de apresentar os resultados da identificação, vale dizer que foram feitos dois conjuntos de identificação: o primeiro sem modelar o atrito nas juntas (desconsiderando o termo  $\tau_{fr}$  na Equação (2.30)) e o segundo modelando o atrito. Testes iniciais feitos sobre a Trajetória 0 não mostraram melhora no *fitness* do modelo ao considerar o atrito. Entretanto, as Trajetórias 1 e 2 inicialmente tiveram um ajuste ruim. Após considerar o ruído no modelo, os resultados obtidos foram significativamente melhores.

As Figuras 4.3, 4.4, 4.5, 4.6, 4.7 e 4.8 mostram os ajustes do modelo às Trajetórias de identificação. A Tabela 4.2 mostra os erros médios quadráticos entre os torques calculados com o modelo e os torques medidos, onde as linhas correspondem às trajetórias usadas na identificação e as colunas correspondem às trajetórias usadas na validação. A Tabela 4.5 mostra o conjunto de parâmetros base e os valores encontrados realizando a identificação, enquanto que a Tabela 4.6 mostra os desvios padrões dos erros de estimação.

Observando as Figuras 4.3 e 4.4, percebe-se que a modelagem do atrito não melhorou muito a qualidade do ajuste. De fato, como pode ser visto na Tabela 4.2, a diferença no *fitness* é negligível para validação com a própria trajetória de identificação e há uma pequena piora no erro para a validação com as outras trajetórias. Além disso, vê-se que o ajuste nas juntas 1, 2 e 3 é bom, enquanto que nas juntas 4, 5 e 6, apesar de seguir o comportamento geral do sinal medido, o ajuste é extremamente ruidoso.

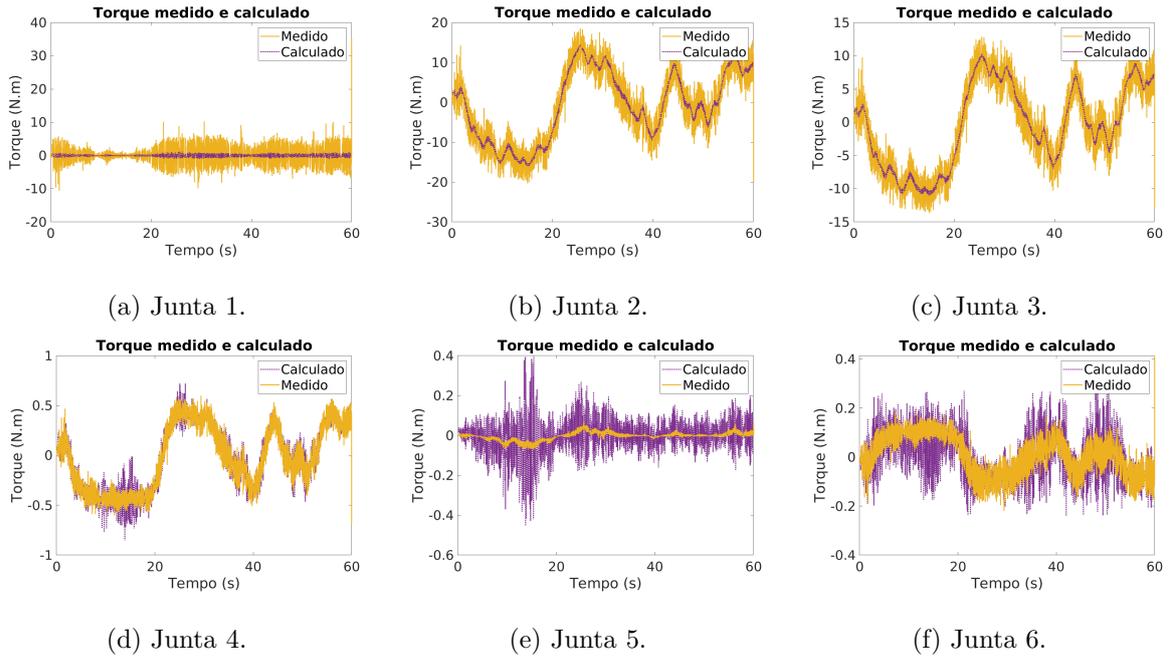


Figura 4.3: Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 0 sem modelar o atrito.

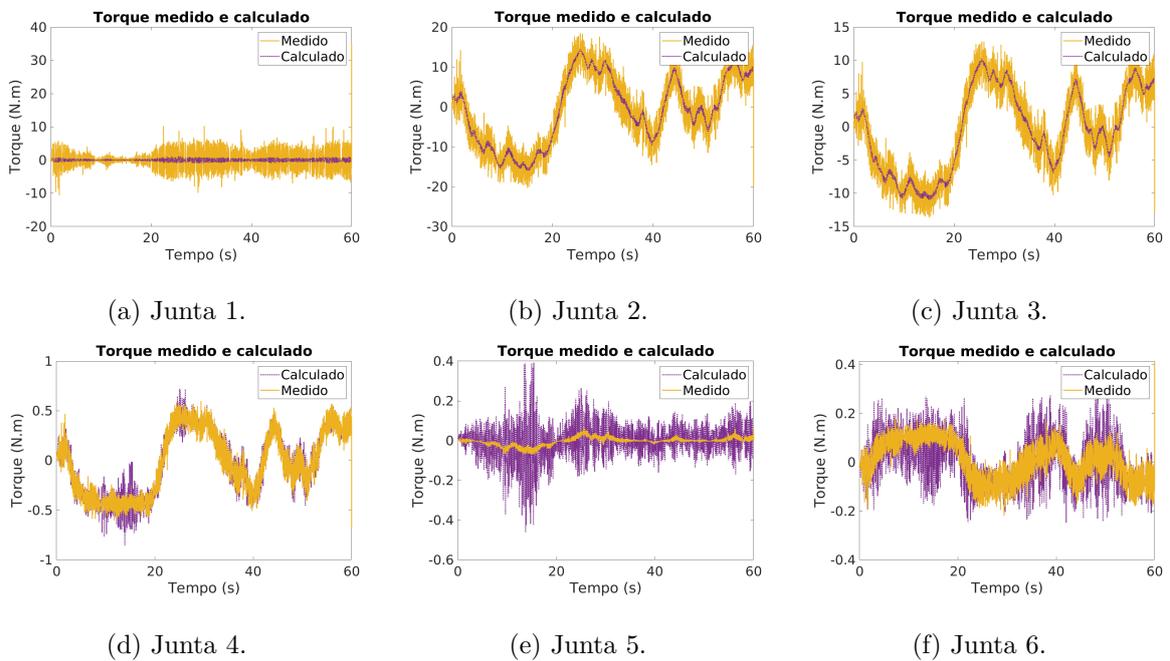
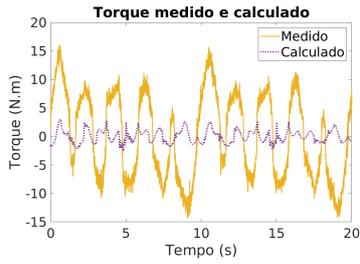
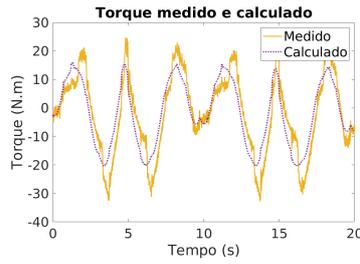


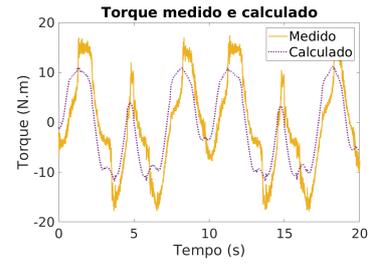
Figura 4.4: Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 0 modelando o atrito.



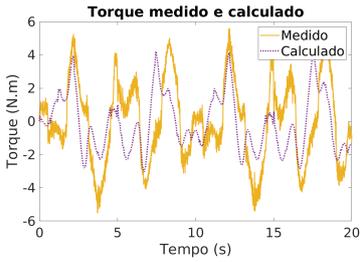
(a) Junta 1.



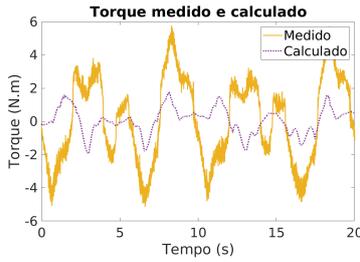
(b) Junta 2.



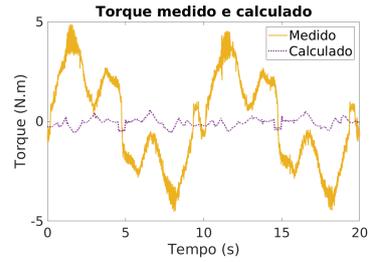
(c) Junta 3.



(d) Junta 4.

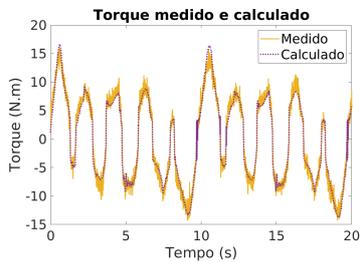


(e) Junta 5.

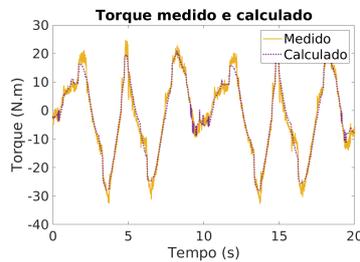


(f) Junta 6.

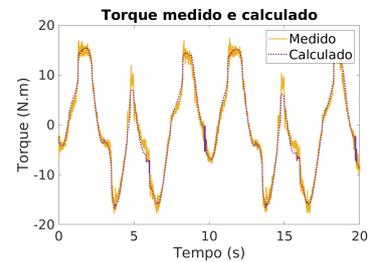
Figura 4.5: Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 1 sem modelar o atrito.



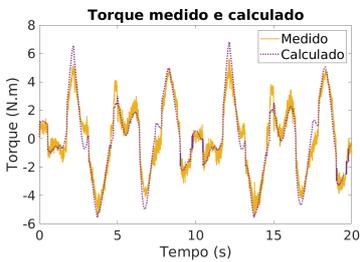
(a) Junta 1.



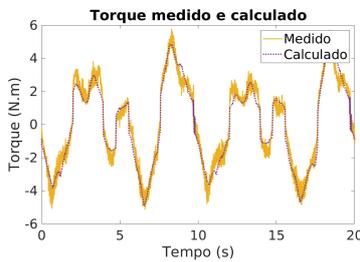
(b) Junta 2.



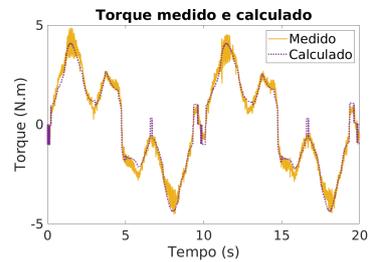
(c) Junta 3.



(d) Junta 4.

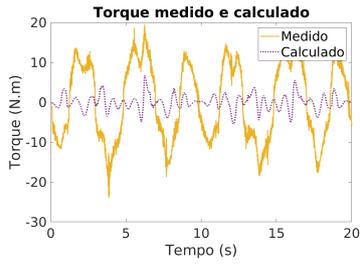


(e) Junta 5.

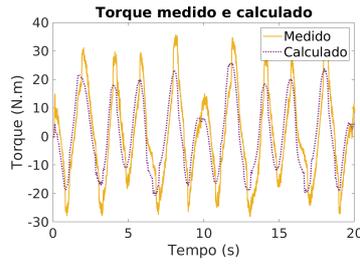


(f) Junta 6.

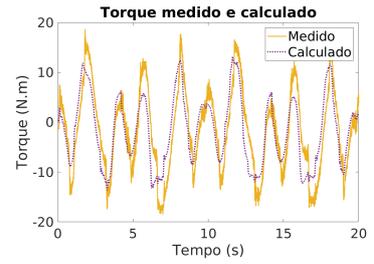
Figura 4.6: Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 1 modelando o atrito.



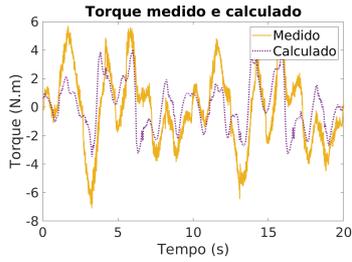
(a) Junta 1.



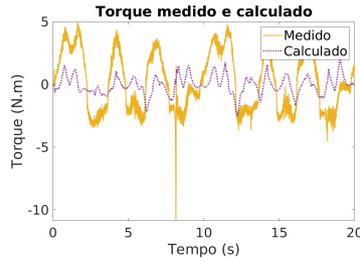
(b) Junta 2.



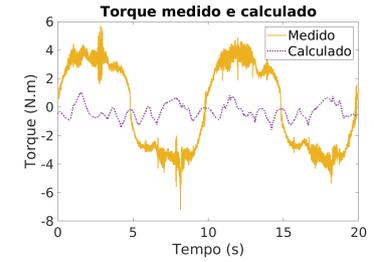
(c) Junta 3.



(d) Junta 4.

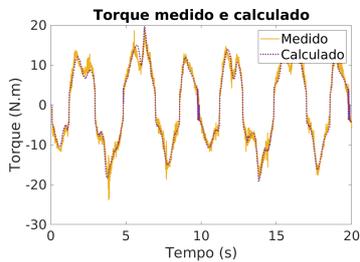


(e) Junta 5.

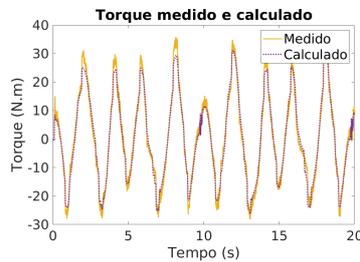


(f) Junta 6.

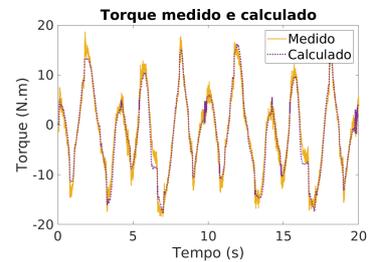
Figura 4.7: Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 2 sem modelar o atrito.



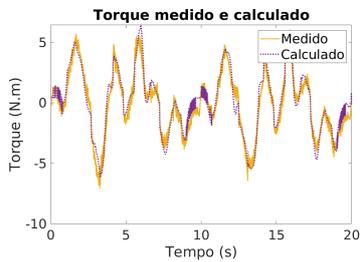
(a) Junta 1.



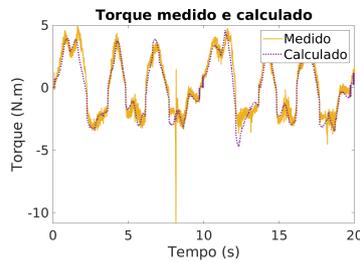
(b) Junta 2.



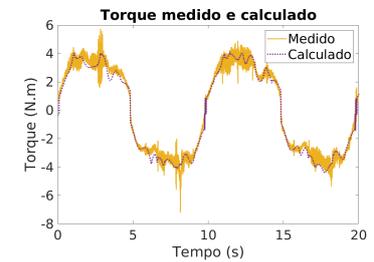
(c) Junta 3.



(d) Junta 4.



(e) Junta 5.



(f) Junta 6.

Figura 4.8: Torques medidos e preditos com os parâmetros identificados das juntas na Trajetória 2 modelando o atrito.

Tabela 4.2: Erro médio quadrático das validações

Tabela 4.3: Modelo sem atrito

		Validação		
		0	1	2
Identif.	0	1.5858	137.8218	149.4358
	1	96.8449	22.3090	41.0509
	2	83.2863	24.9872	31.9539

Tabela 4.4: Modelo com atrito

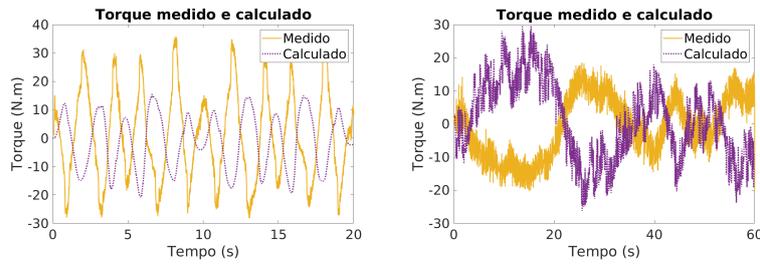
		Validação		
		0	1	2
Identif.	0	1.5856	138.6933	151.8229
	1	106.3038	1.6026	9.5169
	2	93.0282	4.2030	1.7341

Tabela 4.5: Valores dos parâmetros identificados com as Trajetórias 0 e 2

Parâmetro	Valor Traj 0	Valor Traj 2	Parâmetro	Valor Traj 0	Valor Traj 2
$m2$	-32.2695	7.7889	$m6 \cdot rx6 \cdot ry6$	0.0139	0.0016
$m3$	-9.1983	16.3656	$m6 \cdot ry6^2$	0.0194	-0.0063
$m4$	17.7233	-5.6170	$m2 \cdot rz2$	-0.6270	-0.5516
$m5$	-1.4915	14.5558	$m2 \cdot ry2 \cdot rz2$	-0.0509	-0.2434
$m6$	-9.7264	-8.9520	$m3 \cdot ry3 \cdot rz3$	-0.0210	-0.1199
$m1 \cdot rx1^2$	1.0294	0.1973	$m4 \cdot rx4 \cdot rz4$	0.1534	0.1417
$m2 \cdot rx2$	7.7459	-4.1335	$m4 \cdot ry4 \cdot rz4$	0.1247	-0.0585
$m3 \cdot rx3$	-0.4843	-2.7438	$m5 \cdot rz5$	0.7986	0.8175
$m4 \cdot rx4$	0.0018	-0.0392	$m5 \cdot rx5 \cdot rz5$	0.0087	0.0239
$m4 \cdot rx4^2$	-0.1588	-0.0079	$m5 \cdot ry5 \cdot rz5$	0.0712	-0.0761
$m5 \cdot rx5$	-0.0028	-0.0351	$m6 \cdot rx6 \cdot rz6$	-0.0179	0.0229
$m5 \cdot rx5^2$	0.1053	-0.0354	$m6 \cdot ry6 \cdot rz6$	-0.0566	0.0062
$m6 \cdot rx6$	-0.0004	0.0332	$\nu_{v1}$	-0.2390	-3.6460
$m6 \cdot rx6^2$	-0.0413	-0.0115	$\nu_{v2}$	0.0168	-4.7474
$m2 \cdot ry2$	-0.0021	-0.0506	$\nu_{v3}$	0.0939	-2.4645
$m2 \cdot rx2 \cdot ry2$	-0.6168	0.4615	$\nu_{v4}$	0.0883	-1.0033
$m2 \cdot ry2^2$	-1.2193	-0.1762	$\nu_{v5}$	0.0412	-0.8618
$m3 \cdot ry3$	-0.0005	-0.0201	$\nu_{v6}$	0.0649	-0.5752
$m3 \cdot rx3 \cdot ry3$	0.3556	-0.1650	$\nu_{c1}$	0.0281	-3.7471
$m3 \cdot ry3^2$	0.4499	-0.0256	$\nu_{c2}$	-0.0001	-3.6738
$m4 \cdot rx4 \cdot ry4$	0.0287	-0.0994	$\nu_{c3}$	-0.0107	-2.0971
$m5 \cdot ry5$	0.9024	-0.3129	$\nu_{c4}$	-0.0099	-0.9383
$m5 \cdot rx5 \cdot ry5$	-0.0399	-0.0170	$\nu_{c5}$	0.0000	-0.9102
$m6 \cdot ry6$	0.0090	0.0059	$\nu_{c6}$	-0.0055	-1.1142

Tabela 4.6: Desvios padrões dos erros de estimação dos parâmetros. As colunas “sa” correspondem ao modelo sem atrito e as colunas “ca” ao modelo com atrito.

Parâmetro	Desvio padrão escalado ( $1000\sigma_{\theta_i}$ )					
	Trajetória 0		Trajetória 1		Trajetória 2	
	sa	ca	sa	ca	sa	ca
$m2$	47.7758	47.7927	11.2233	2.4593	6.3042	1.2631
$m3$	18.6278	18.6313	9.8846	2.1646	8.7494	1.7558
$m4$	29.1833	29.1964	19.6891	4.3099	16.9151	3.3906
$m5$	29.1070	29.1164	19.6844	4.3083	17.7165	3.5510
$m6$	2.7666	2.7747	7.4351	1.6259	12.0042	2.4067
$m1 \cdot rx1^2$	1.0921	1.0925	0.6288	0.1378	0.4251	0.0852
$m2 \cdot rx2$	7.1370	7.1396	2.8346	0.6210	1.4869	0.2982
$m3 \cdot rx3$	3.9691	3.9705	0.8959	0.1964	0.7637	0.1531
$m4 \cdot rx4$	0.0074	0.0074	0.0504	0.0110	0.1116	0.0224
$m4 \cdot rx4^2$	0.2101	0.2102	0.1358	0.0297	0.1234	0.0247
$m5 \cdot rx5$	0.0074	0.0074	0.0465	0.0102	0.0827	0.0166
$m5 \cdot rx5^2$	0.0203	0.0204	0.0572	0.0125	0.0799	0.0160
$m6 \cdot rx6$	0.0025	0.0025	0.0357	0.0078	0.0576	0.0116
$m6 \cdot rx6^2$	0.0221	0.0221	0.0285	0.0062	0.0450	0.0090
$m2 \cdot ry2$	0.0031	0.0031	0.0667	0.0146	0.1275	0.0256
$m2 \cdot rx2 \cdot ry2$	0.4036	0.4037	0.2911	0.0637	0.3112	0.0624
$m2 \cdot ry2^2$	1.7394	1.7401	0.6682	0.1465	0.4876	0.0979
$m3 \cdot ry3$	0.0034	0.0034	0.0683	0.0149	0.1096	0.0220
$m3 \cdot rx3 \cdot ry3$	0.4002	0.4003	0.1814	0.0398	0.1565	0.0314
$m3 \cdot ry3^2$	0.8472	0.8475	0.1976	0.0433	0.1803	0.0362
$m4 \cdot rx4 \cdot ry4$	0.0153	0.0153	0.0942	0.0206	0.1288	0.0258
$m5 \cdot ry5$	2.4843	2.4851	1.5487	0.3390	1.3371	0.2679
$m5 \cdot rx5 \cdot ry5$	0.0244	0.0244	0.0433	0.0095	0.0661	0.0132
$m6 \cdot ry6$	0.0027	0.0027	0.0380	0.0083	0.0595	0.0121
$m6 \cdot rx6 \cdot ry6$	0.0192	0.0192	0.0211	0.0046	0.0358	0.0072
$m6 \cdot ry6^2$	0.0217	0.0217	0.0312	0.0068	0.0465	0.0094
$m2 \cdot rz2$	0.2173	0.2186	1.5195	0.3324	1.7947	0.3600
$m2 \cdot ry2 \cdot rz2$	0.0482	0.0482	0.5721	0.1252	0.6031	0.1208
$m3 \cdot ry3 \cdot rz3$	0.0266	0.0266	0.1731	0.0379	0.1891	0.0379
$m4 \cdot rx4 \cdot rz4$	0.1534	0.1535	0.1242	0.0272	0.1086	0.0218
$m4 \cdot ry4 \cdot rz4$	0.0195	0.0196	0.0961	0.0211	0.1366	0.0275
$m5 \cdot rz5$	0.2266	0.2273	0.6122	0.1339	0.9843	0.1974
$m5 \cdot rx5 \cdot rz5$	0.0114	0.0114	0.0321	0.0070	0.0527	0.0106
$m5 \cdot ry5 \cdot rz5$	0.0139	0.0139	0.0502	0.0110	0.0746	0.0150
$m6 \cdot rx6 \cdot rz6$	0.0118	0.0118	0.0281	0.0062	0.0495	0.0100
$m6 \cdot ry6 \cdot rz6$	0.0128	0.0128	0.0242	0.0053	0.0430	0.0086



(a) Modelo identificado na Trajetória 0 e aplicado sobre a Trajetória 2.  
 (b) Modelo identificado na Trajetória 2 e aplicado sobre a Trajetória 0.

Figura 4.9: Torques preditos opostos aos mensurados. Sinais da junta 1.

Ao analisar as Figuras 4.5, 4.6, 4.7 e 4.8, nota-se que a modelagem do atrito melhorou consideravelmente o ajuste das curvas. Essa percepção é confirmada pela Tabela 4.2, que mostra uma grande redução no erro médio quadrático entre as Trajetórias 1 e 2, embora haja um aumento no erro da validação com a Trajetória 0.

Considerando que as Trajetórias 1 e 2 são parametrizadas da mesma maneira, era esperado que produzissem resultados similares com relação à identificação. O aumento do erro de validação com a modelagem do atrito, observado tanto na Trajetória 0 com relação às trajetórias 1 e 2 quanto no contrário, pode ser devido a um overfitting, em que o modelo perde parte da capacidade de se ajustar a trajetórias genéricas.

Como a Trajetória 2 produziu uma identificação melhor que a 1 (Tabela 4.2), a partir desse ponto serão consideradas apenas as trajetórias 0 e 2, devido à sua diferente natureza.

A Tabela 4.5 mostra os valores dos parâmetros dinâmicos identificados segundo ambas trajetórias. A ausência dos parâmetros dos tensores de inércia é justificada de acordo com a Seção 2.4.2. Como foi discutido, os parâmetros identificáveis formam um conjunto linearmente dependente e, portanto, existem múltiplas maneiras de se escolher a base desse conjunto. Coincidentemente, a base escolhida não continha os parâmetros dos tensores de inércia. Entretanto, esses parâmetros podem ser obtidos por meio de uma combinação linear dos parâmetros da Tabela 4.5. Como o propósito deste trabalho não é a obtenção de tais parâmetros, eles foram ignorados, uma vez que os parâmetros base são suficientes para descrever a dinâmica do sistema.

Alguns valores obtidos para os parâmetros foram negativos. Isso é obviamente fisicamente impossível, e é produto de imprecisões nos procedimentos, tais como incertezas nas medições (ruídos), componentes dinâmicos não modelados, como flexibilidade dos links ou comportamento dos rotores dos motores [15]. Além disso, a dinâmica do controlador não foi modelada, a identificação foi feita em malha fechada com um modelo de malha aberta, o que muito provavelmente é uma grande fonte de erros. Uma consequência de parâmetros negativos é que, para algumas configurações, a matriz de massa  $D(q)$  do robô (Equação (2.26)) deixa de ser simétrica ou se torna negativa definida, o que afeta operações sobre ela, como a inversão [23].

Outro fato notável sobre os parâmetros identificados é que alguns são positivos na identificação

segundo a Trajetória 0, mas negativos na identificação segundo a Trajetória 2, e vice-versa. Esse fenômeno pode servir de justificativa para o comportamento estranho observado na validação entre as trajetórias, como mostra a Figura 4.9, onde o torque predito aparenta ser o oposto do torque real. Embora a figura mostre apenas os dados da junta 1, esse comportamento foi observado em grau similar também nas juntas 2 e 3.

Os desvios padrões do erro de estimação encontram-se na Tabela 4.6, onde os desvios foram escalados para grandezas mais palpáveis e as colunas “sa” e “ca” se referem, respectivamente, aos modelos sem atrito e com atrito. A modelagem do atrito diminuiu em cerca de 5 vezes os desvios padrões das estimações com base nas Trajetórias 1 e 2, enquanto que praticamente não alterou os desvios advindos da Trajetória 0. Tomando esses desvios padrões como uma medida de confiança no parâmetro estimado, constata-se que as estimativas vindas das Trajetórias 1 e 2 são mais precisas que as da Trajetória 0. Esse comportamento já era esperado, considerando que o número de condição da Trajetória 0 é ordens de magnitude maior que os demais.

## 4.2 Estimação de força

O Filtro de Kalman Unscented foi implementado em MATLAB em dois programas: um que lê os dados salvos de um experimento e outro que lê os dados diretamente dos tópicos ROS e faz a estimação em tempo real.

Ambos programas foram testados e aparentaram funcionar corretamente, havendo apenas alguns problemas notáveis com a estimação da aceleração, que serão abordados mais adiante. No entanto, o filtro em tempo real, devido ao tempo elevado necessário para executar uma iteração do UKF (causado pelo cálculo de muitos pontos sigma), não pôde trabalhar na frequência de amostragem do robô, 125 Hz, mas necessitou de receber dados amostrados a uma taxa menor. Essa subamostragem foi feita por um nó ROS dedicado a isso, como foi discutido na Seção 3.5. Essa limitação faz com que as estimativas sejam menos confiáveis, uma vez que há menos informação. A implementação em MATLAB é vantajosa pela praticidade de fazer mudanças e ajustes, mas é limitada no quesito de desempenho. É provável que, em uma versão final do filtro implementada em uma linguagem mais eficiente, como C++, possa-se executar o filtro em tempo real com a taxa de amostragem do robô.

Devido à ausência de instrumentos para medição das acelerações e forças externas, os filtros foram avaliados qualitativamente, com perturbações esporádicas na forma de empurrões. O filtro pareceu funcionar de maneira adequada.

Para ilustrar o comportamento do filtro, ele foi aplicado sobre as trajetórias 0 e 1, sem força externa, usando os parâmetros dinâmicos identificados por essas trajetórias. Os resultados obtidos podem ser vistos nas Figuras 4.10 a 4.15. Como o filtro seguiu bem as posições e velocidades, as imagens mostram apenas as acelerações, torques e esforços externos.

No teste da Trajetória 0, vê-se que as acelerações estimadas são extremamente ruidosas. No entanto, ao olhar mais de perto, na Figura 4.16a, é possível ver que no geral o filtro segue o valor

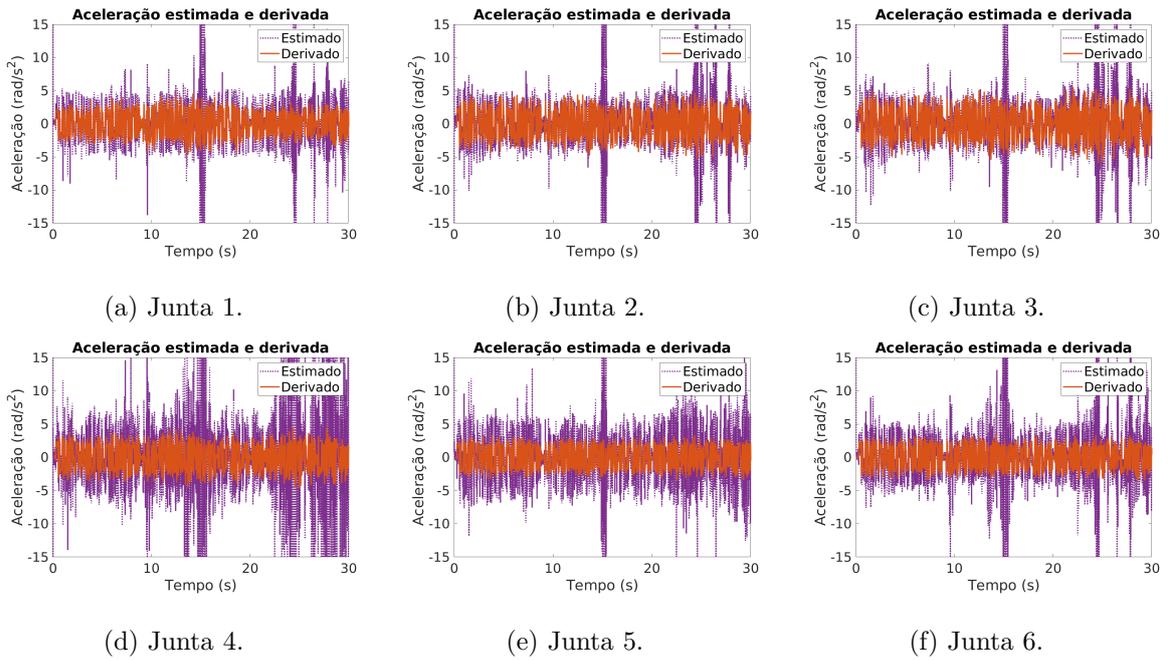


Figura 4.10: Acelerações estimadas pelo UKF na Trajetória 0.

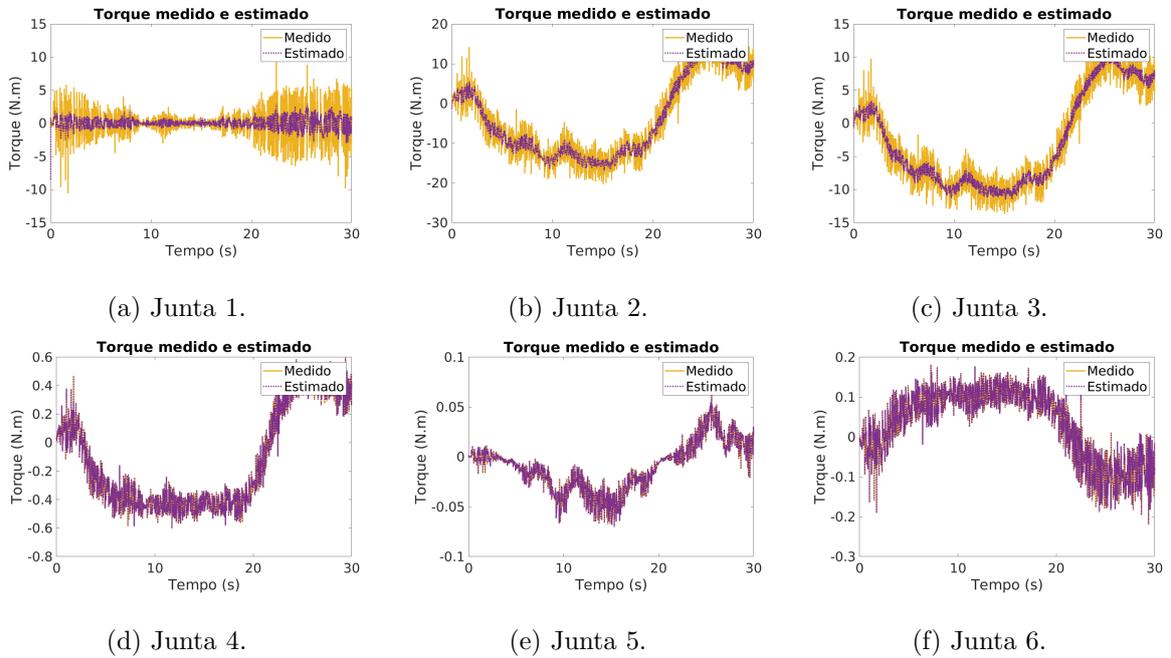


Figura 4.11: Torques estimados pelo UKF na Trajetória 0.

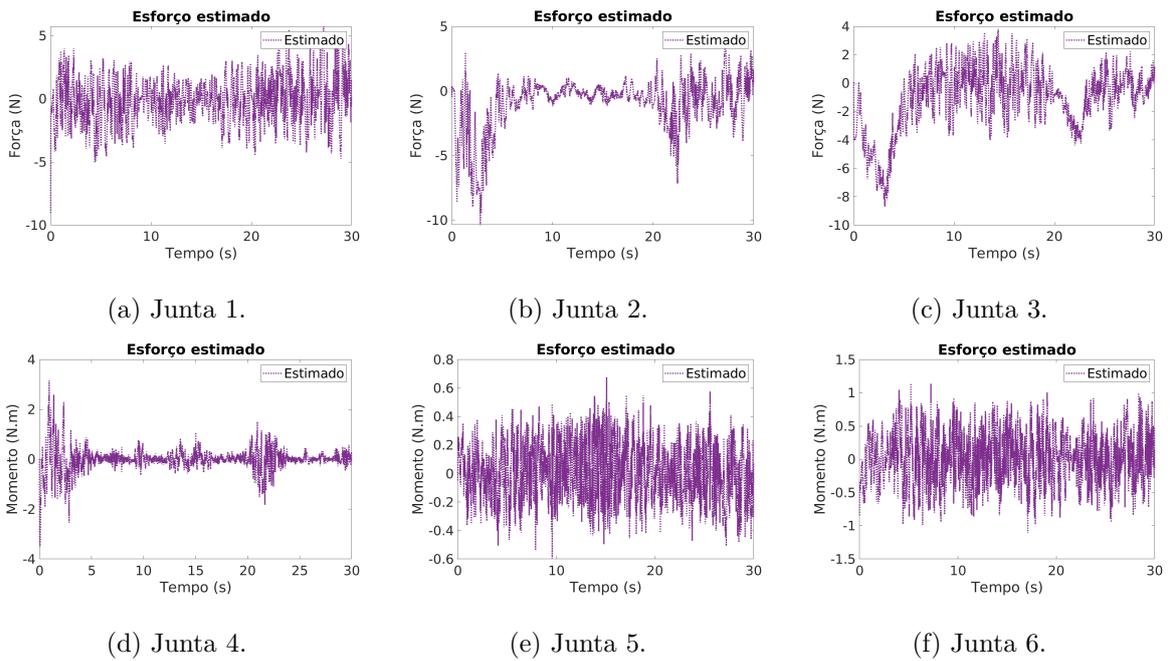


Figura 4.12: Forças estimadas pelo UKF na Trajetória 0.

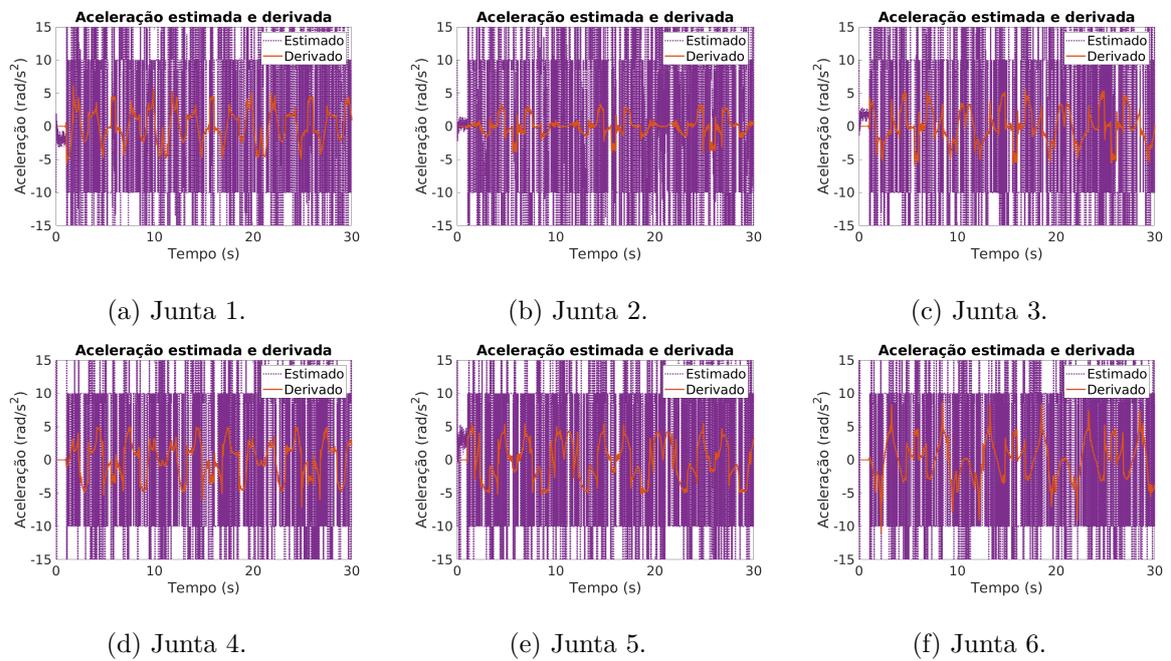


Figura 4.13: Acelerações estimadas pelo UKF na Trajetória 1.

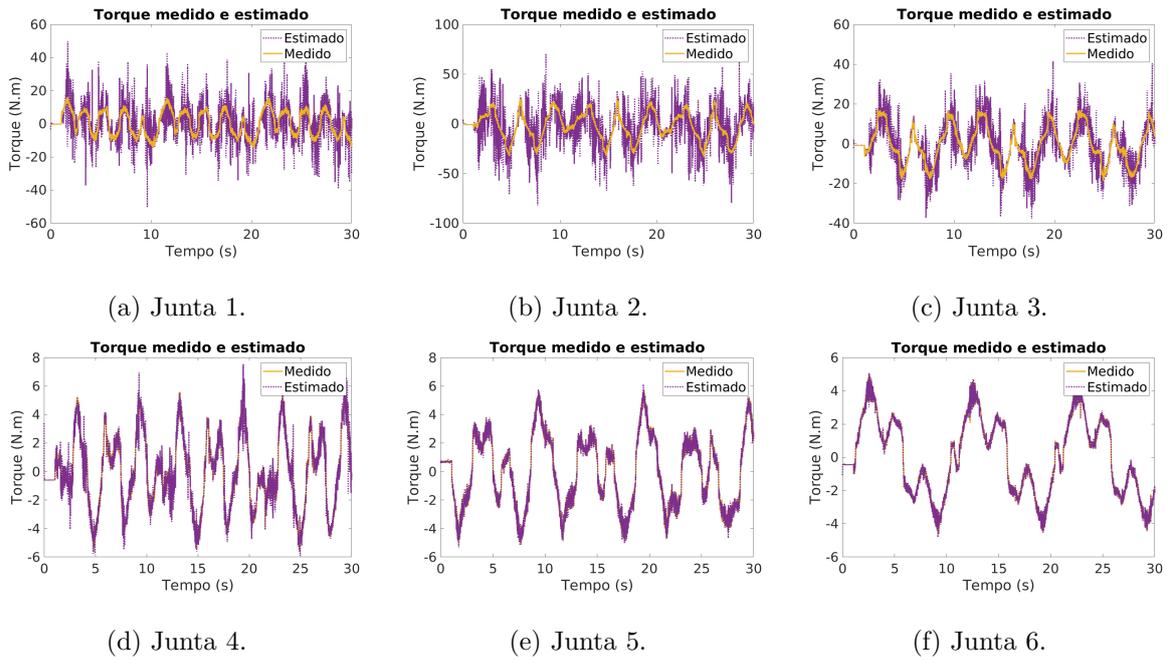


Figura 4.14: Torques estimados pelo UKF na Trajetória 1.

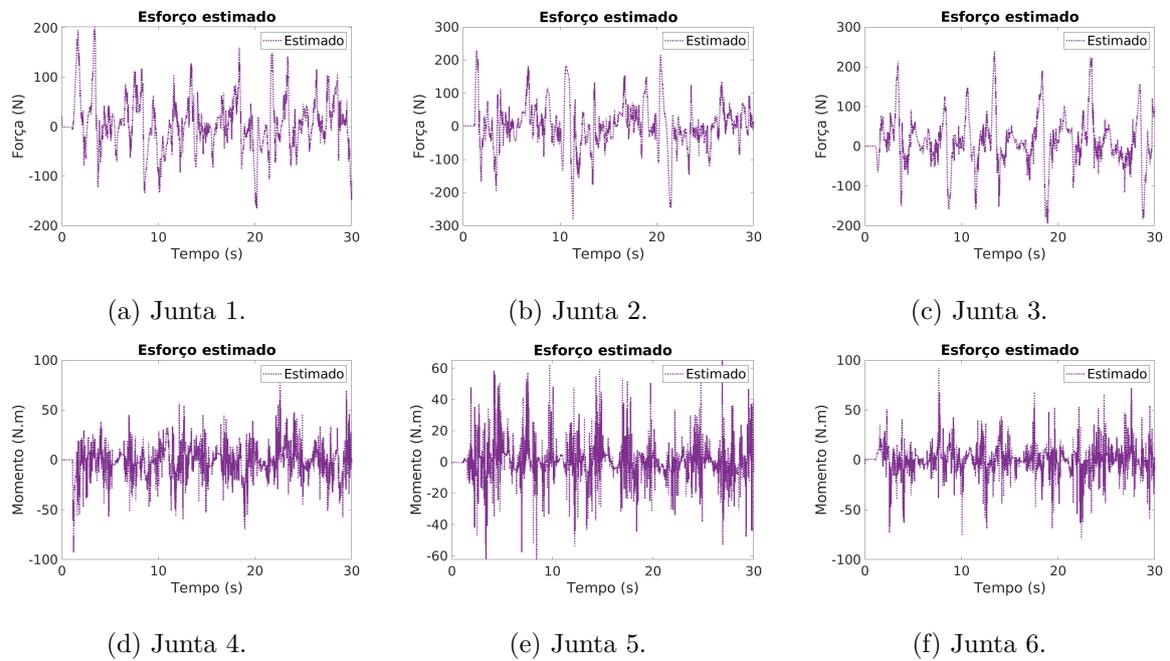


Figura 4.15: Forças estimadas pelo UKF na Trajetória 1.

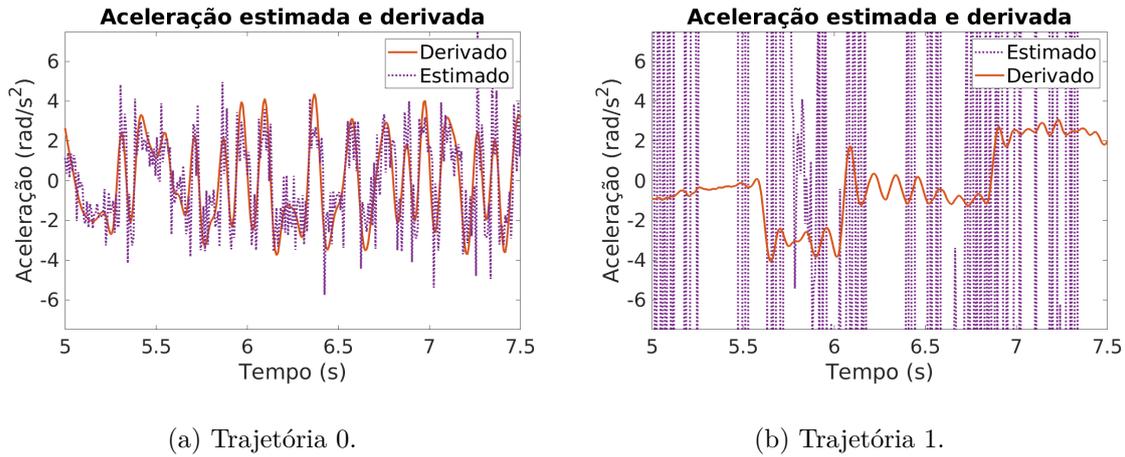


Figura 4.16: Visualização de uma pequeno trecho dos sinais de aceleração. Ambos são da junta 2, mas um é do experimento com a Trajetória 0 e outro com a Trajetória 1.

real. Além disso, as acelerações apresentam picos da ordem de  $10^4$ , que ocorrem devido à inversão da matriz  $D(q)$  durante o cálculo da dinâmica direta. Em certas configurações, a matriz  $D(q)$  assume valores muito próximos de zero, o que faz com que a matriz invertida seja absurdamente grande. Esse problema em particular pode ser mitigado buscando outros métodos de obtenção da dinâmica direta ou através de alguma espécie de saturação. A estimação ruim também pode ser atribuída a um ajuste ruim dos parâmetros do UKF e às imprecisões nos parâmetros dinâmicos identificados. Apesar desses problemas com as acelerações, os torques estimados são bons, seguindo os valores medidos, e os esforços no efetuator terminal estão distribuídos em torno do 0, conforme o esperado, e com pouca variação.

No teste da Trajetória 1, as mesmas considerações se aplicam. A estimação da aceleração aqui tem um erro ainda maior que a anterior. Olhando de perto, na Figura 4.16b, observa-se que, diferentemente do experimento na Trajetória 0, aqui o filtro não segue o valor real. Os torques estimados apresentam um ajuste bom, embora bem ruidoso. Os esforços externos, no entanto, embora ainda estejam distribuídos em torno do 0, são estimados sem precisão nenhuma, com variações da ordem de centenas de Newtons (ou N.m). Especula-se que esse modelo tenha tido um desempenho tão ruim em comparação ao modelo da Trajetória 0 por causa da pior estimação das acelerações e/ou porque os parâmetros dinâmicos obtidos não são adequadamente sensíveis a excitações do tipo esforços externos, o que gera uma grande incerteza na estimação.

Para uma avaliação mais formal, decidiu-se gerar artificialmente um sinal de força e aplicar a sua influência sobre os dados das trajetórias de identificação. Dessa forma, o valor esperado já é conhecido com precisão. Foram gerados pulsos de magnitude 10 (N no caso de força e N.m no caso de momento) entre 5s e 15s nas direções  $x$ ,  $y$  e  $z$  do efetuator terminal, sendo gerados ao total 6 simulações, cada uma com o esforço aplicado de uma maneira (forças nos 3 eixos e momentos em torno dos 3 eixos). A esses sinais foi adicionado um ruído gaussiano para simular algo mais realístico. Os torques nas juntas devido a esse esforço externo foram calculados de acordo com a Equação (2.29) e somados aos torques medidos nas trajetórias. Então aplicou-se o UKF sobre

esses dados e obteve-se a estimação de força.

Essa simulação foi feita para a Trajetória 1, mas a força estimada foi similar à Figura 4.15, sendo que qualquer força real existente estaria completamente perdida no meio das incertezas. Sendo assim, serão mostradas aqui apenas as estimações geradas com a Trajetória 0, que podem ser vistas na Figura 4.17. Reitera-se o fato de que cada um dos gráficos da Figura 4.17 corresponde a uma simulação separada, na qual o esforço externo foi aplicado apenas na respectiva componente.

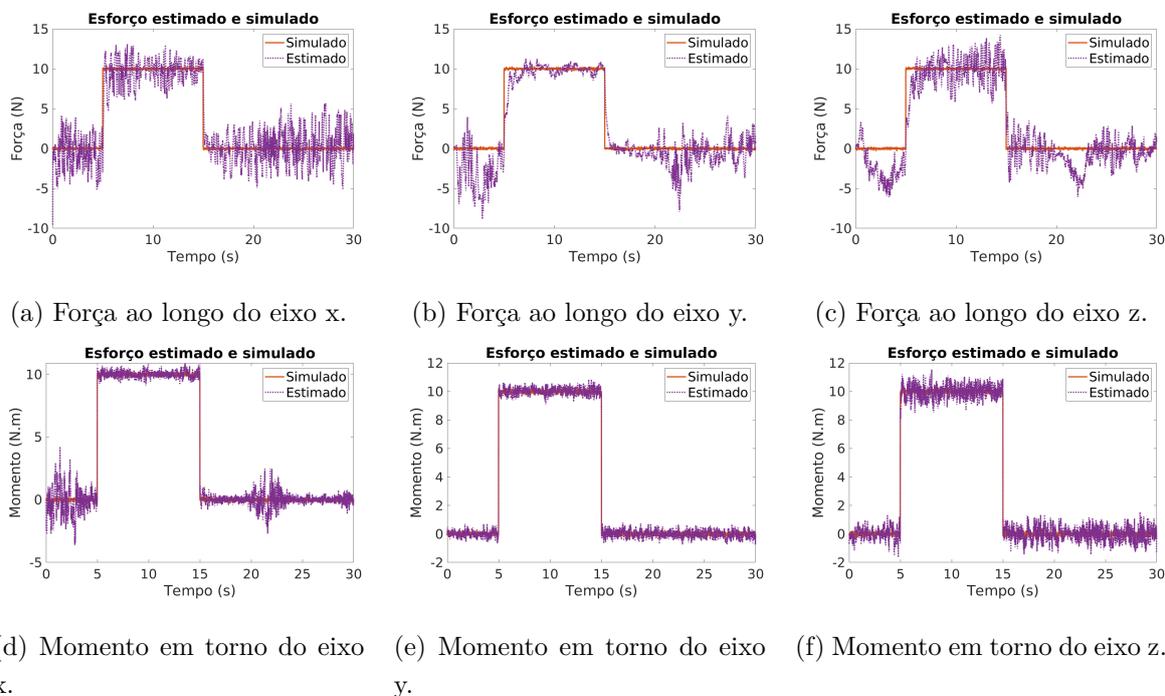


Figura 4.17: Forças estimadas pelo UKF na Trajetória 0.

Considerando a Figura 4.17 é possível perceber que o UKF é capaz de estimar os esforços de interação sobre o efetuator terminal. Em todas as curvas é possível delinear a forma do pulso aplicado, embora nas Figuras 4.17a, 4.17b e 4.17c haja uma distorção maior na forma do pulso quando comparadas com as Figuras 4.17d, 4.17e e 4.17f. Esse comportamento é esperado, uma vez que a aplicação direta de um torque sobre o efetuator terminal impacta mais diretamente os torques das juntas do que a aplicação de uma força.

# Capítulo 5

## Conclusão

O objetivo deste trabalho era a estimação da força de interação sobre o efetuador terminal no braço robótico UR3.

A princípio, foram consideradas motivações para alcançar esse objetivo, sendo a principal a possibilidade de desenvolver um sistema de colaboração entre robôs e seres humanos, que é um problema importante e suas soluções são aplicadas na vanguarda do cenário industrial. Também foram mencionados métodos de estimação de força utilizados por outros autores, com o intuito de fornecer uma visualização dessa área de pesquisa.

Seguindo essa introdução, foi feita uma apresentação dos fundamentos teóricos necessários para a realização e compreensão deste trabalho.

A primeira parte da fundamentação consistiu na exposição da derivação do modelo dinâmico de um manipulador utilizando a formulação de Lagrange. Foram abordados conceitos adjacentes, como a cinemática direta e a cinemática diferencial. Dentro dessa derivação, foram apresentados os passos necessários para o cálculo das equações. Também foram adicionados a este modelo as influências do atrito e dos esforços externos, sendo este último o próprio objetivo deste trabalho. Mais tarde percebeu-se que não foi levada em conta a dinâmica de controle do robô. Embora a arquitetura relativamente fechada e a possibilidade de programar o robô apenas em alto nível dificulte a obtenção dos parâmetros dos controladores das juntas, seria possível obter ao menos valores estimados, que certamente tornariam o modelo dinâmico mais acurado e possibilitariam uma melhor identificação.

Dando continuidade à modelagem dinâmica, explicou-se em seguida o procedimento de identificação dos parâmetros dinâmicos. Uma vez que estes parâmetros não são conhecidos para o UR3, a forma mais prática de obtê-los é através de uma identificação. Foram expostos os fundamentos relacionados, que vão desde o ajuste do modelo dinâmico à análise estatística dos parâmetros identificados, passando antes pela obtenção de uma boa trajetória de identificação.

De posse do modelo dinâmico identificado, prosseguiu-se para a abordagem do estimador de estados, o Filtro de Kalman Unscented. Iniciando pela apresentação do Filtro de Kalman simples (linear), expandiu-se gradualmente a complexidade do filtro até chegar aos Filtro de Kalman

Unscented de Raiz Quadrada, explicando a cada passo dessa evolução as razões por trás da nova formulação.

A fundamentação foi concluída com uma apresentação do ROS e do UR3. Os conceitos básicos do ROS necessários para o entendimento deste trabalho foram explicados de maneira sucinta, enquanto que as características relevantes do UR3 foram exploradas.

A próxima parte do trabalho tratou da exposição dos procedimentos tomados durante o desenvolvimento do trabalho, relacionando-os com a teoria relevante apresentada no capítulo anterior. Estes procedimentos seguiram basicamente a mesma ordem de apresentação dos conceitos teóricos, e eles levaram ao próximo capítulo.

Os resultados obtidos foram apresentados no Capítulo 4. Eles consistiram basicamente em dois grupos: resultados da identificação e resultados da estimação de força.

Quanto aos resultados da identificação, obteve-se bom ajuste às Trajetórias de identificação. Foi mostrado que a modelagem do atrito é fundamental na obtenção de um modelo preciso. Através do uso de trajetórias diferentes foi possível observar que as identificações forneceram apenas um ajuste àquela trajetória em particular, faltando uma adequação mais geral do modelo.

Devido à dificuldade em realizar as medições necessárias para uma análise completa, os resultados da estimação de força apresentados tiveram um teor mais qualitativo. Foi possível perceber que o modelo dinâmico direto é fonte de grandes dificuldades, mas que mesmo assim o Filtro de Kalman Unscented é capaz de realizar estimações relativamente boas. Foi demonstrado através de uma simulação que a técnica implementada é capaz de estimar a força externa mesmo sem medições. No entanto, as limitações de um modelo dinâmico que não foi bem identificado implicam em grandes prejuízos nessa estimação.

Os resultados obtidos apontam para um caminho claro para melhoria. Uma vez que foi estabelecido que a qualidade da estimação do UKF depende fortemente do modelo dinâmico, o próximo passo seria realizar uma identificação melhor, conseguindo capturar as características de todos os parâmetros dinâmicos. Para isso seria essencial levar em conta a dinâmica de controle do robô. No processo de identificação, poderia-se utilizar novas técnicas, como identificação sequencial, ou aprimorar as técnicas utilizadas aqui, gerando uma trajetória mais otimizada, por exemplo.

Outra fonte de problemas foi a equação de dinâmica direta, que fez com que as acelerações assumissem valores absurdamente grandes devido a singularidades na trajetória. Esse problema deve ser resolvido buscando outra maneira de se obter a dinâmica direta ou encontrando uma maneira de contornar essa situação, expandindo o modelo dinâmico direto de modo a torná-lo resiliente a essas situações.

Resolvendo esses dois problemas, teria-se em mãos um excelente modelo dinâmico do robô. A partir daí poderia-se ajustar os parâmetros do UKF de modo a se obter uma estimativa mais precisa. Além disso, seria interessante buscar formas de controlar o robô através do ROS de uma maneira que garanta maior controle e melhor qualidade dos dados lidos.

Finalmente, o estimador de força externa seria implementado de maneira eficiente, de preferência em uma linguagem como o C++, fazendo com que seja possível e viável a execução em

tempo real. Esse estimador poderia ser naturalmente conectado à rede do ROS, possibilitando um acesso fácil à força estimada.

# REFERÊNCIAS BIBLIOGRÁFICAS

- [1] KUFIETA, K. Force estimation in robotic manipulators: Modeling, simulation and experiments. *Department of Engineering Cybernetics NTNU Norwegian University of Science and Technology*, 2014.
- [2] Universal Robot UR3. <https://www.universal-robots.com/products/ur3-robot/>. Acesso: 2019-08-18.
- [3] Unimate - The First Industrial Robot. <https://www.robotics.org/joseph-engelberger/unimate.cfm>. Acesso: 2019-07-01.
- [4] BICCHI, A.; PESHKIN, M. A.; COLGATE, J. E. Safety for physical human–robot interaction. *Springer handbook of robotics*, Springer, p. 1335–1348, 2008.
- [5] About Compliant Mechanisms. <https://www.compliantmechanisms.byu.edu/about-compliant-mechanisms>. Acesso: 2019-07-01.
- [6] VUKOBRATOVIC, M.; TUNESKI, A. Contact control concepts in manipulation robotics/spl minus/an overview. *IEEE Transactions on industrial electronics*, IEEE, v. 41, n. 1, p. 12–24, 1994.
- [7] STOLT, A. et al. Force controlled robotic assembly without a force sensor. In: IEEE. *2012 IEEE International Conference on Robotics and Automation*. 2012. p. 1538–1543.
- [8] DAMME, M. V. et al. Estimating robot end-effector force from noisy actuator torque measurements. In: IEEE. *2011 IEEE International Conference on Robotics and Automation*. 2011. p. 1108–1113.
- [9] MURAKAMI, T.; YU, F.; OHNISHI, K. Torque sensorless control in multidegree-of-freedom manipulator. *IEEE Transactions on Industrial Electronics*, IEEE, v. 40, n. 2, p. 259–265, 1993.
- [10] COLOMÉ, A. et al. External force estimation during compliant robot manipulation. In: IEEE. *2013 IEEE International Conference on Robotics and Automation*. 2013. p. 3535–3540.
- [11] SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. *Robot Modeling and Control*. : Wiley, 2005.
- [12] KHALIL, E. D. W. *Modeling Identification and Control of Robots*. : Butterworth-Heinemann, 2004.

- [13] WAN, E. A.; MERWE, R. V. D. The unscented kalman filter for nonlinear estimation. In: IEEE. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. 2000. p. 153–158.
- [14] MERWE, R. V. D.; WAN, E. A. The square-root unscented kalman filter for state and parameter-estimation. In: IEEE. *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*. 2001. v. 6, p. 3461–3464.
- [15] RADKHAH, K.; KULIC, D.; CROFT, E. Dynamic parameter identification for the crs a460 robot. In: IEEE. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007. p. 3842–3847.
- [16] GAUTIER, M.; KHALIL, W. Exciting trajectories for the identification of base inertial parameters of robots. *The International journal of robotics research*, Sage Publications Sage CA: Thousand Oaks, CA, v. 11, n. 4, p. 362–375, 1992.
- [17] PARK, K.-J. Fourier-based optimal excitation trajectories for the dynamic identification of robots. *Robotica*, Cambridge University Press, v. 24, n. 5, p. 625–633, 2006.
- [18] HAUGEN, F. A. *State estimation with Kalman Filter*. <http://techt teach.no/fag/seky3322/0708/kalmanfilter/kalmanfilter.pdf>. Material do prof. Haugen para o curso de Cibernética 2 na Oslo and Akershus University College.
- [19] SIMON, D. *Optimal state estimation: Kalman, H-infinity, and nonlinear approaches*. : Wiley-Interscience, 2006.
- [20] ROS/Introduction - ROS Wiki. <http://wiki.ros.org/ROS/Introduction>. Acesso: 2019-08-18.
- [21] UNIVERSAL ROBOTS. *The URScript Programming Language*.
- [22] Parameters for calculations of kinematics and dynamics. <https://www.universal-robots.com/how-tos-and-faq/faq-ur-faq/parameters-for-calculations-of-kinematics-and-dynamics-45257/>. Acesso: 2019-11-20.
- [23] KOVINCIC, N. et al. Dynamic parameter identification of the universal robots ur5.