

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LEONARDO DE MELO ABREU

**MODELAGEM CINEMÁTICA E DINÂMICA DE ROBÔS MODELO SCARA E
DESENVOLVIMENTO DE APLICATIVO PARA PLANEJAMENTO E CONTROLE
DE TRAJETÓRIA**

CAMPO MOURÃO

2022

LEONARDO DE MELO ABREU

**MODELAGEM CINEMÁTICA E DINÂMICA DE ROBÔS MODELO SCARA E
DESENVOLVIMENTO DE APLICATIVO PARA PLANEJAMENTO E CONTROLE
DE TRAJETÓRIA**

**Kinematic and dynamic modeling of robots model SCARA and application
development for planning and trajectory control**

Trabalho de conclusão de curso de graduação,
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Eletrônica da
Universidade Tecnológica Federal do Paraná
(UTFPR)

Orientador: Dr. Flávio Luiz Rossini.

Coorientador: Me. Luiz Fernando Pinto de Oliveira.

CAMPO MOURÃO

2022



Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

LEONARDO DE MELO ABREU

**MODELAGEM CINEMÁTICA E DINÂMICA DE ROBÔS MODELO SCARA E
DESENVOLVIMENTO DE APLICATIVO PARA PLANEJAMENTO E CONTROLE
DE TRAJETÓRIA**

Trabalho de conclusão de curso de graduação,
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Eletrônica da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 02/junho/2022

Gilson Junior Schiavon

Doutor

Universidade Tecnológica Federal do Paraná (UTFPR)

Leandro Castilho Brolin

Doutor

Universidade Tecnológica Federal do Paraná (UTFPR)

Flávio Luiz Rossini

Doutor

Universidade Tecnológica Federal do Paraná (UTFPR)

CAMPO MOURÃO

2022

AGRADECIMENTOS

Em primeiro lugar quero agradecer e expressar a minha eterna gratidão a minha mãe, Célia de Melo Abreu, que mesmo com incontáveis dificuldades e obstáculos conseguiu me apoiar, sendo a minha base por todos esses anos que se passaram durante a graduação, uma trabalhadora incansável, uma mulher que respeito e admiro desde criança e continuarei respeitando e admirando cada vez mais sua força e perseverança ao longo de toda a minha vida.

Agradeço também as minhas irmãs, Janaina de Melo Prigol e Juliana de Melo Abreu, ao meu pai, José de Abreu, e demais familiares por fazerem parte da minha vida e pelo suporte fornecido durante todo esse período.

Estendo ainda meus agradecimentos a todas as minhas amigas constituídas durante a graduação, sem dúvidas essas pessoas fizeram toda a diferença na minha trajetória, tanto dentro como fora da universidade.

E por fim, agradeço ao corpo docente da UTFPR, em especial ao meu orientador, professor Dr. Flávio Luiz Rossini, e ao coorientador, Me. Luiz Fernando Pinto de Oliveira, pelo suporte e orientações prestadas neste trabalho de conclusão de curso.

RESUMO

O presente trabalho consiste no desenvolvimento da modelagem cinemática e dinâmica de robôs modelo SCARA (*Selective Compliance Articulated Robot*) e desenvolvimento de aplicativo para planejamento e controle de trajetória. O mesmo visa facilitar a aprendizagem de robótica por meio das representações gráficas geradas pela aplicação desenvolvida, dada a crescente utilização de robôs manipuladores por diversos campos da indústria e a dificuldade em encontrar materiais didáticos que explorem a simulação de robôs manipuladores industriais. Apresenta-se primeiramente o desenvolver matemático para as modelagens da cinemática direta, cinemática inversa e geração de trajetória, por conseguinte, o algoritmo utilizado para a obtenção das equações dinâmicas e, por fim, elaborou-se uma proposta de controle particionado com base nas características dinâmicas do manipulador. A partir das equações obtidas e implementadas no *software* de simulação MATLAB® R2021a, desenvolveu-se o aplicativo no qual o usuário pode realizar a simulação de posições e geração de trajetórias para qualquer robô do tipo SCARA que tenham estrutura similar ao robô SCARA SR-6 iA. Após testes e simulações, conclui-se que o aplicativo desenvolvido cumpre com êxito a função de realizar os cálculos e representações gráficas de todas as etapas de desenvolvimento (cinemática, dinâmica e controle), sendo viável, sua utilização para fins educacionais e profissionais.

Palavras-chave: robôs manipuladores; cinemática; dinâmica; controle.

ABSTRACT

The present work consists in the development of kinematic and dynamic modeling of SCARA (*Selective Compliance Articulated Robot*) robots and application development for planning and trajectory control. The same aims to facilitate the learning of robotics through the graphic representations generated by the application developed, dada the increasing use of manipulative robots by various fields of industry and the difficulty in finding didactic materials that explore the simulation of industrial manipulator robots. The mathematical development for the modeling of direct kinematics, inverse kinematics and trajectory generation is presented, therefore, the algorithm used to obtain dynamic equations and, finally, a partitioned control proposal was elaborated based on the dynamic characteristics of the manipulator. From the equations obtained and implemented in the MATLAB® R2021a simulation *software*, the application was developed in which the user can perform the simulation of positions and generation of trajectories for any robot of type SCARA that have similar structure to the robot SCARA SR-6 iA. After tests and simulations, it is concluded that the developed application successfully fulfills the function of performing the calculations and graphic representations of all stages of development (kinematics, dynamics and control), being feasible, its use for educational and professional purposes.

Keywords: manipulative robots; kinematics; dynamics; control.

LISTA DE ILUSTRAÇÕES

Figura 1 – Tipos de juntas	14
Figura 2 – Manipulador articulado	15
Figura 3 – Manipulador cartesiano	16
Figura 4 – Manipulador cilíndrico	16
Figura 5 – Manipulador esférico	17
Figura 6 – Manipulador SCARA	17
Figura 7 – Possíveis trajetórias para juntas.....	21
Figura 8 – Sistema de controle	22
Figura 9 – Interface para desenvolvimento gráfico (GUI)	23
Figura 10 – Interface do ambiente de programação.....	24
Figura 11 – Modelo simplificado do robô.....	25
Figura 12 – Vista superior do manipulador.....	27
Figura 13 – Tela inicial do aplicativo (aba “Cinemática”)	35
Figura 14 – Segunda aba do aplicativo “Dinâmica e Controle”	37
Figura 15 – Resposta cinemática direta para as entradas $\theta_1 = 0^\circ$, $\theta_2 = 0^\circ$, $d_3 = 0$ mm e $\theta_4 = 0^\circ$	39
Figura 16 – Resposta cinemática direta para as entradas $\theta_1 = 180^\circ$, $\theta_2 = 0^\circ$, $d_3 = 50$ mm e $\theta_4 = 0^\circ$	40
Figura 17 – Resposta cinemática direta para as entradas $\theta_1 = 90^\circ$, $\theta_2 = 90^\circ$, $d_3 = 50$ mm e $\theta_4 = 90^\circ$	40
Figura 18 – Resposta cinemática direta para as entradas $x = 650$ mm, $y = 0$ m e, $z = 0$ mm	41
Figura 19 – Resposta cinemática direta para as entradas $x = -650$ mm, $y = 0$ m e, $z = 0$ mm	41
Figura 20 – Resposta cinemática direta para as entradas $x = 350$ mm, $y = 300$ m e, $z = 50$ mm	42
Figura 21 – Diagrama de simulação da geração de trajetória	43
Figura 22 – Geração da trajetória para tempo = 10 s e passos = 0,2 s.....	44
Figura 23 – Comportamento temporal junta 1	44
Figura 24 – Comportamento temporal junta 2	45
Figura 25 – Comportamento temporal junta 3	46
Figura 26 – Comportamento temporal junta 4	47
Figura 27 – Geração da trajetória para tempo = 10 s e passos = 0,2 s.....	48
Figura 28 – Comportamento temporal (primeira junta)	50
Figura 29 – Comportamento temporal (segunda junta)	48
Figura 30 – Comportamento temporal (terceira junta).....	51
Figura 31 – Comportamento temporal (quarta junta).....	52
Figura 32 – Diagrama de simulação do sistema de controle.....	53
Figura 33 – Simulação do controle para a primeira junta rotativa $k_p = 9$ e $k_v = 10$	54
Figura 34 – Simulação do controle para a segunda junta rotativa $k_p = 9$ e $k_v = 10$	55
Figura 35 – Simulação do controle para a junta prismática $k_p = 9$ e $k_v = 10$..	55

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	11
1.1.1	Objetivo geral	11
1.1.2	Objetivos específicos	11
1.2	Justificativa	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Robôs manipuladores	13
2.1.1	Juntas e elos	13
2.1.2	Graus de liberdade	14
2.1.3	Configurações	14
<u>2.1.3.1</u>	<u>Manipulador articulado</u>	<u>15</u>
<u>2.1.3.2</u>	<u>Manipulador cartesiano</u>	<u>15</u>
<u>2.1.3.3</u>	<u>Manipulador cilíndrico</u>	<u>16</u>
<u>2.1.3.4</u>	<u>Manipulador esférico</u>	<u>16</u>
<u>2.1.3.5</u>	<u>Manipulador SCARA</u>	<u>17</u>
2.2	Cinemática	17
2.2.1	Cinemática direta	18
2.2.2	Cinemática inversa	18
<u>2.2.2.1</u>	<u>Métodos para solução</u>	<u>19</u>
2.3	Dinâmica	19
2.4	Geração de trajetória - espaço de juntas	20
2.5	Controle	21
2.6	Ambiente de desenvolvimento (<i>App Designer</i>)	23
3	METODOLOGIA	25
3.1	Modelagem da cinemática direta	25
3.2	Modelagem cinemática inversa	26
3.3	Obtenção das equações dinâmicas	29
3.3.1	Tensores de inércia	29
3.3.2	Algoritmo iterativo de Newton-Euler	30
3.4	Geração de trajetória	32
3.5	Modelagem do sistema de controle	33

4	RESULTADOS E DISCUSSÕES	35
4.1	Aplicativo	35
4.2	Modelagem matemática e simulação cinemática direta	37
4.3	Simulação cinemática inversa.....	40
4.4	Simulação geração de trajetória	42
4.5	Simulação do controle	52
5	CONCLUSÃO	57
	REFERÊNCIAS.....	59
	APÊNDICE A - Vista lateral do robô.....	62
	APÊNDICE B - Equação cinemática direta	64
	APÊNDICE C - Equações dinâmicas.....	66

1 INTRODUÇÃO

Inúmeras inovações tecnológicas ocorreram no decorrer da história, sendo uma delas a invenção do motor a vapor, que gerou grande desenvolvimento e crescimento industrial, o que veio a se tornar um marco importante para os dias atuais. Esses tipos de avanços e inovações tecnológicas, tais como: o surgimento de válvulas, computadores, dispositivos semicondutores, possuem reflexos diretos na indústria, seja para melhorar o processo fabril ou para otimização de recursos (FENERICK; VOLANTE, 2020).

O termo “robô” é derivado da palavra tcheca *robotnik*, que pode significar trabalho forçado, uma espécie de sinônimo para escravo, e rotina de trabalho (servo). O termo teve sua primeira aparição registrada em uma peça teatral chamada Robôs Universais de Rossum (RUR), escrita pelo escritor tcheco Karel Čapek em 1921. Na obra fictícia, robôs são construídos em larga escala, os quais no decorrer dos anos vieram a substituir a mão e obra humana (LEAL, 2020; QUINTINO, 2015; ROSÁRIO, 2010).

Na década de 50 surge o primeiro robô industrial projetado pelo engenheiro norte-americano George Devol. O *Unimate* era um dispositivo do tipo braço robótico, responsável pela automatização de algumas tarefas na fábrica da *General Motors* localizada em Nova Jersey (GAMERO, 2018).

A partir de uma forma de entretenimento, os robôs desenvolvidos pela *Unimation*, empresa criada por Devol, logo passaram a ser desenvolvidos para a utilização na produção industrial em larga escala. Com isso, grandes empresas entraram no investimento da produção de robôs na década de 80 (MULTI COMERCIAL, 2020). A ideia inicial era substituir a mão de obra humana em ambientes com altos níveis de calor, substâncias tóxicas, intenso esforço físico, trabalhos repetitivos, entre outras situações, com maior velocidade e melhor otimização de recursos (ROSÁRIO, 2010).

A aplicação de motores instalados nas juntas ainda na década de 80 por Victor Motta, mudou a forma de movimentação dos robôs, o que veio a tornar os manipuladores mais velozes e precisos. Possibilitou dessa forma, que manipuladores realizassem funções mais elaboradas tais como: montagem e soldagem de equipamentos (GAMERO, 2018).

Atualmente a definição de robô atribuída pela *International Organization for Standardization* e seguida pela *International Federation of Robotics* (IFR), o robô industrial é definido como: “manipulador multifuncional reprogramável automaticamente controlado e programável com três ou mais graus de liberdade” (IFR, 2020, n.p, tradução nossa).

Os robôs industriais atualmente são projetados para serem multifuncionais, sendo em sua grande maioria controlados por computadores ou microprocessadores, o que possibilita a reprogramação para realizarem diversas funções em diferentes segmentos da indústria, ao realizar a modificação de algumas linhas de código ou mudança de seu efetuador (NIKU, 2020; QUINTINO, 2015).

Graças a essa versatilidade as aplicações mais comuns dos manipuladores industriais são: soldagem de componentes e peças, linhas de montagem, deslocamento de cargas, realização de testes, entre outras aplicações (GAMERO, 2018; ROMANO; DUTRA, 2002).

Robôs classificados como SCARA são robôs que possuem duas juntas rotativas que são paralelas, que fornecem conformidade em um plano (IFR, 2020). O manipulador SCARA, que conta com as três juntas rotativas paralelas, possui o benefício de que essas juntas não precisam suportar o peso do manipulador ou da carga. Outra vantagem da utilização de manipuladores SCARA está em seus atuadores que podem ser grandes, o que permite movimentos até 10 vezes mais rápidos que robôs articulados industriais (CRAIG, 2009).

O projeto e desenvolvimento de um robô abrange diferentes campos do conhecimento que buscam dentro da engenharia elétrica, engenharia mecânica, engenharia de *software*, matemática, física e entre outras áreas, a integração de técnicas e algoritmos, com finalidade de melhorar o desempenho dos manipuladores. Desta maneira, procura-se representar através de simulações os fenômenos físicos de maneiras mais próximos ao real possível, em ambientes de simulação (OLIVEIRA, 2016; OLIVEIRA; ROSSINI, 2018; OLIVEIRA *et. al.*, 2020).

Tendo em vista a necessidade da modelagem matemática e simulação de robôs industriais, o presente trabalho apresenta o desenvolvimento da modelagem cinemática direta, cinemática inversa, e da dinâmica, além da geração de trajetória e controle de um manipulador do tipo SCARA. A partir desses resultados, elaborou-se um aplicativo interativo capaz de mostrar graficamente os resultados do

planejamento e controle de trajetória, ao utilizar o *software* de simulação MATLAB® R2021a para implementação e análise desses resultados.

O aplicativo tem como finalidade ser uma aplicação genérica, de forma que o mesmo pode ser utilizado para realizar as modelagens e gerações de trajetória para qualquer manipulador do tipo SCARA, independente do fabricante. A partir da configuração dos parâmetros de tamanho dos elos, massa de cada corpo móvel, posição cartesiana do robô e valores como: posições desejadas para o robô, tempo de trajetória, quantidades de pontos para as simulações e valores dos ganhos do sistema de controle.

Para ilustrar os resultados apresentados neste Trabalho de Conclusão de Curso (TCC), usou-se como referência o robô SCARA SR-6 iA - desenvolvido e comercializado pela empresa Fanuc - para análise dos resultados obtidos nas simulações.

1.1 Objetivos

Nesta seção são apresentados o objetivo geral e os objetivos específicos que servirão como base para a elaboração do trabalho.

1.1.1 Objetivo geral

O objetivo geral é modelar a cinemática e a dinâmica do robô, a fim de construir um aplicativo para simulações para análises de trajetória e controle, a partir da ferramenta *App Designer* presente no software MATLAB® R2021a.

1.1.2 Objetivos específicos

- Modelar a cinemática direta;
- Obter a cinemática inversa;
- Implementar a geração de trajetória;
- Obter as equações dinâmicas;
- Propor um sistema de controle;
- Elaborar códigos no *software* MATLAB® R2021a;
- Projetar e construir o aplicativo;
- Analisar e validar os resultados numéricos e gráficos gerados.

1.2 Justificativa

Em 2021 no relatório apresentado pela IFR, foi apresentado um recorde de 3 milhões de robôs industriais em operação em todo o mundo. O número de novos robôs vendidos cresceu cerca de 0,5% em relação ao relatório anterior, mesmo com o cenário de pandemia global, foram distribuídas cerca de 384.000 unidades globalmente em 2020 (IFR, 2021a).

No Brasil, em 2020, estima-se um total de 16.117 unidades em funcionamento, um aumento de 13%, quando comparado com os números de 2015, com uma taxa de crescimento anual de 3% (IFR, 2021b).

Estudos mostram a dificuldade do Brasil em formar engenheiros em comparação a outros países. O contato com a robótica idealmente deveria começar antes da entrada no ensino superior, onde os alunos possam ter contato com áreas da engenharia. Diminuindo assim a distância entre contextos teóricos apresentados aos alunos, das práticas e simulações da engenharia. O que facilitaria a aprendizagem de disciplinas que envolvem conceitos matemáticos e físicos, ao apresentar exemplos que utilizam como base as áreas de eletrônica, mecânica, computação e entre outras (FREITA *et al.*, 2021).

Dado a grande versatilidade dos robôs do tipo SCARA e ao levar em consideração a crescente utilização de robôs no decorrer dos anos, com tendência de alta para os próximos anos, são necessárias mão de obra qualificada para a aplicação desses manipuladores na indústria.

Deste modo, é imprescindível facilitar o aprendizado da robótica, a partir do desenvolvimento matemático e aplicações nas quais o usuário pode checar e analisar resultados numéricos, assim como, simular trajetórias, analisar comportamentos estáticos, dinâmicos e controle de robôs em interfaces simples e interativas com o usuário.

Assim, foi proposto por esse trabalho o projeto e desenvolvimento de um aplicativo configurável para robôs SCARA, com propósito de facilitar o ensino e aprendizagem de robótica, além da sua possível utilização em ambiente industrial para a realização de análises de trajetórias estáticas. Assim como, realizar análise de cada junta do robô ao longo da trajetória percorrida ao levar em consideração as suas características dinâmicas e possíveis alterações das constantes de erro de velocidade e posição no sistema de controle implementado.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos e as características físicas sobre robôs manipuladores, além de apresentar a teoria necessária para o desenvolvimento da cinemática direta e inversa, obtenção das equações dinâmicas, geração de trajetória e controle para robôs manipuladores.

2.1 Robôs manipuladores

A partir da definição de robôs adotada pela *Robotic Industries Association (RIA)*, pode-se inferir sobre as características dos robôs manipuladores, sendo as duas principais, realizar tarefas programadas e mover objetos (PAZOS, 2002).

A área de trabalho dos robôs manipuladores é o volume que delimita o espaço alcançável de seu efetuador. Esta área é definida a partir das propriedades das juntas e dos elos da estrutura. Os manipuladores geralmente são montados sobre uma base fixa, e dispõe de uma área de trabalho bem delimitada para a movimentação de todos os corpos, inclusive do efetuador (PAZOS, 2002).

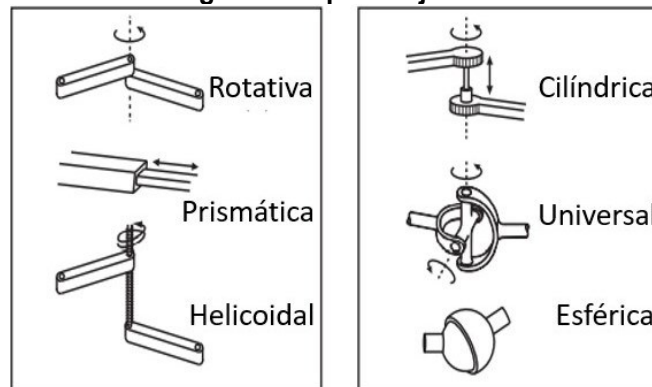
Segundo Niku (2020), é possível traçar e fazer uma comparação entre um guindaste e um robô manipulador, pois ambos possuem como função a manipulação de objetos dentro de seu espaço de trabalho. A principal diferença é que o guindaste é manipulado por ação humana, enquanto o robô possui um microprocessador ou computador que contém um programa responsável pelo controle do robô, esse programa pode ser alterado para a execução de diferentes tarefas.

2.1.1 Juntas e elos

A estrutura mecânica de um robô manipulador é composta essencialmente por justas e elos. A seguir são descritas as juntas típicas.

- a) Juntas: é comum na robótica a utilização de juntas prismáticas e rotativas, a combinação dessas juntas visa ocasionar um processo de produção mais simplificado (ROMANO; DUTRA, 2002). No entanto, conforme pode-se observar na Figura 1, existem outros tipos juntas que podem ser utilizadas além das prismáticas e rotativas, tais como: helicoidal, cilíndrica, universal e esférica.

Figura 1 - Tipos de juntas



Fonte: Adaptado de Lynch (2017)

- b) Elos: são os corpos responsáveis por ligar uma junta a outra. Pode ser de natureza estática ou dinâmica. Os elos são projetados para suportar forças de torção e flexão, sendo os materiais mais utilizados para a construção dessas estruturas o alumínio, aço, fibras de carbono e vidro (ROMANO; DUTRA, 2002).

2.1.2 Graus de liberdade

Pode ser contabilizado pelo total de juntas do manipulador. Há certa associação dos graus de liberdade com as funções que poderão ser executadas pelo robô. Para as funções mais complexas, são necessários mais graus de liberdade, o que dificulta o controle de movimentação a cada vez que se aumenta o número de elos do manipulador (PAZOS, 2002).

Por exemplo, um robô com 6 graus de liberdade possui três juntas responsáveis para o posicionamento do efetuador dentro da área de trabalho, as outras três restantes são para o efetuador segurar o objeto corretamente de acordo com a orientação desejada (PAZOS, 2002; SANTOS, 2004).

De acordo com Santos (2004), os graus de liberdade podem ser definidos simplesmente como o número de movimentos independentes que o manipulador pode realizar.

2.1.3 Configurações

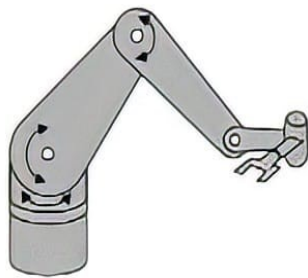
Como exemplo de um robô manipulador, tem-se o braço mecânico, que consiste em corpos rígidos ligados por juntas e elos, desta forma, permite a movimentação entre os corpos. Na teoria, pode-se ter inúmeras configurações para diferentes arranjos de juntas. Entretanto, na prática há poucas combinações que

são mais utilizadas, tais como: manipulador articulado (RRR), esférico (RRP), SCARA (RRP), cilíndrico (RPP) e cartesiano (PPP) (SPONG *et al.*, 2006).

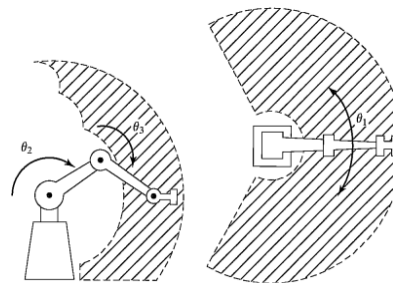
2.1.3.1 Manipulador articulado

Também chamado de cotovelo ou antropomórfico. Essa configuração consiste em uma junta rotativa na horizontal e outra junta rotativa posicionada na vertical, nas quais permitem o deslocamento tanto na horizontal como na vertical do manipulador, pode ainda possuir mais duas ou três juntas o que depende do robô e suas funções (CRAIG, 2009). Um exemplo de manipulador articulado e sua respectiva área de trabalho pode ser observado na Figura 2.

Figura 2 - Manipulador articulado
 (a) Vista lateral do manipulador (b) Área de trabalho do manipulador



Fonte: Niku (2020)

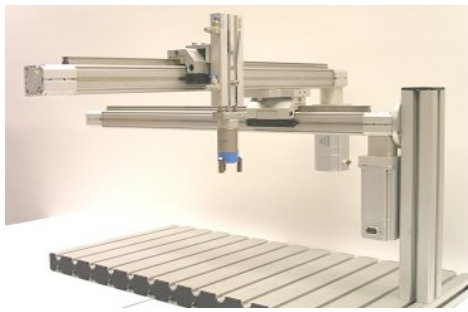


Fonte: Craig (2009)

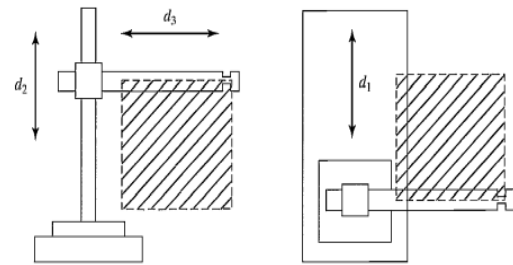
2.1.3.2 Manipulador cartesiano

Formado pelas três primeiras juntas prismáticas (PPP), seu movimento é orientado no plano cartesiano. Esse é o mais simples de todos os manipuladores (SPONG *et al.*, 2006). Um exemplo desse manipulador e sua respectiva área de trabalho pode ser visto na Figuras 3.

Figura 3 - Manipulador cartesiano
(a) Vista do manipulador **(b) Área de trabalho do manipulador**



Fonte: Festo (2020)



Fonte: Craig (2009)

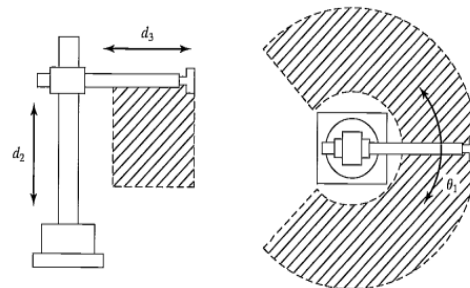
2.1.3.3 Manipulador cilíndrico

Neste tipo de robô, a primeira junta é rotativa, seguida por duas juntas prismáticas. A partir dessa configuração, a primeira junta permite que o primeiro elo consiga girar até 360° em torno do seu próprio eixo, que em conjunto com o restante da estrutura do robô, forma uma área de trabalho cilíndrica (PAZOS, 2002). Na Figura 4 ilustra-se um exemplo de manipulador cilíndrico e sua respectiva área de trabalho.

Figura 4 - Manipulador cilíndrico
(a) Vista do manipulador **(b) Área de trabalho do manipulador**



Fonte: Niku (2020)



Fonte: Craig (2009)

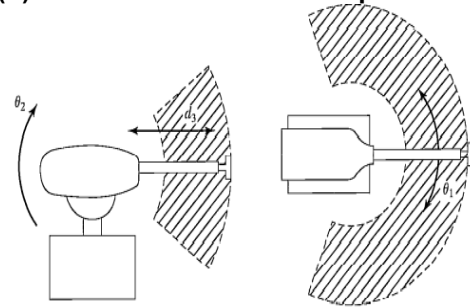
2.1.3.4 Manipulador esférico

Os eixos do robô formam um sistema de coordenadas polar, tem como base o centro de referência, faz uso de juntas na configuração (PRR), no qual forma uma área de trabalho que se aproxima ao formato de uma esfera (ROMANO; DUTRA, 2002). Na Figura 5, encontra-se um robô cilíndrico comercializado para a indústria com a representação de sua área de trabalho.

Figura 5 - Manipulador Esférico
(a) Vista do manipulador **(b) Área de trabalho do manipulador**



Fonte: Romano e Dutra (2002)



Fonte: Craig (2009)

2.1.3.5 Manipulador SCARA

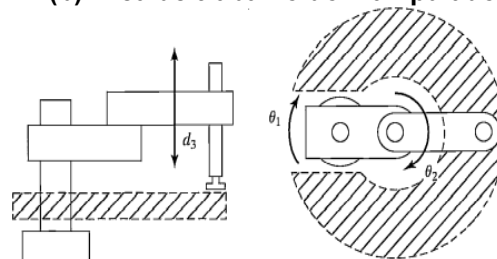
Para a configuração ilustrada na Figura 6, está presente o robô SCARA T3 da Epson. Nota-se que o manipulador possui duas juntas rotativas que rotacionam em um mesmo plano, além de possuir uma junta prismática no final do efetuador (CRAIG, 2009).

De acordo com Romano e Dutra (2002), esse tipo de manipulador é amplamente utilizado em funções de montagem de componentes de pequenas dimensões e placas em circuitos eletrônicos. Essa é a melhor escolha para soluções planares onde precisa-se de um robô rápido (CRAIG, 2009).

Figura 6 - Manipulador SCARA
(a) Vista do manipulador **(b) Área de trabalho do manipulador**



Fonte: Epson (2002)



Fonte: Craig (2009)

2.2 Cinemática

A partir da obtenção das equações cinemáticas, e a partir do conhecimento do valor de deslocamento de cada junta, é possível determinar o ponto no espaço onde a extremidade do efetuador do robô estará (NIKU, 2020).

2.2.1 Cinemática direta

A cinemática direta representa a obtenção dos valores da posição e orientação do efetuador. A obtenção destas coordenadas se dá pela aplicação dos ângulos desejados nas equações da cinemática direta e deslocamento quando utilizada a junta prismática, sendo assim possível ter conhecimento da posição do efetuador com relação a base (LYNCH; PARK, 2017; CRAIG, 2009).

A modelagem da cinemática direta se obtém por meio do produto das matrizes homogêneas de transformação que interligam à base do robô até o efetuador (LYNCH; PARK, 2017). Em Craig (2009), apresenta-se a cinemática direta, que relaciona todos os *links* calculados através das matrizes de transformações individuais, o que torna possível a obtenção da posição final do último sistema $\{N\}$ em relação ao sistema de referência $\{0\}$, expressa por:

$${}^0T_N = {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{N-1}T_N \quad (1)$$

Segundo Santos (2004), são necessários alguns passos para obtenção das equações da cinemática direta, sendo eles:

- (i) Definir uma posição inicial para o robô;
- (ii) Definir os sistemas de coordenadas de cada elo;
- (iii) Determinar às relações de rotação e translação;
- (iv) Determinar as matrizes de rotação e realizar a multiplicação das matrizes de transformação, conforme a Equação (1), obtém-se a relação entre o referencial do robô e efetuador.

2.2.2 Cinemática inversa

A partir da cinemática direta, obtém-se a posição final do efetuador do robô através da mudança dos valores nas juntas. No caso da cinemática inversa, o processo acontece ao contrário, isto é, através da posição final do efetuador se determinar a posição final das juntas (SANTOS, 2004).

2.2.2.1 Métodos para solução

Existem dois tipos de categorias de estratégias para encontrar a solução da cinemática inversa, sendo elas as estratégias de solução numérica (do inglês *Numerical solution*) e a estratégia de solução fechada (do inglês *Closed-form solution*) (CRAIG, 2009).

Métodos numéricos podem ser aplicados nas equações da cinemática inversa, isso pode ser uma solução alternativa, caso não possuam soluções analíticas. Podem vir a serem utilizados também com a finalidade de aprimorar a precisão de soluções (LYNCH; PARK, 2017).

Existem vários métodos iterativos para encontrar as raízes de uma equação. Como exemplo de algoritmos iterativos, tem-se a aplicação do método de Newton-Raphson para se encontrar raízes de equações ou sistemas de equações não lineares (LYNCH; PARK, 2017).

Soluções fechadas podem ser divididas em duas vertentes, no método geométrico e no método trigonométrico. Ambos os métodos são muito parecidos, pois nas aplicações do método geométrico faz-se uso de expressões algébricas. Como possível diferença entres os métodos, tem-se apenas a abordagem utilizada (CRAIG, 2009).

Essas soluções fechadas apresentam respostas mais rápidas que as soluções numéricas, nas quais suas implementações são muito utilizadas dentro do campo da cinemática. Outro ponto importante é que normalmente a cinética inversa possui inúmeras soluções, desse modo, o algoritmo de solução fechada permite o desenvolvimento de regras para a escolha de uma solução única dentre várias possíveis (SPONG *et al.*, 2006).

2.3 Dinâmica

Nesta parte de desenvolvimento são encontradas as equações dinâmicas utilizadas para descrever as forças e torque atuantes no manipulador (NIKU, 2020).

Para se obter as equações dinâmicas pode-se usar duas metodologias: a formulação de Newton-Euler, que é uma abordagem que se baseia no equilíbrio de forças e a formulação Lagrangiana que se baseia na energia (MURRAY; LI; SASTRY, 2017; CRAIG, 2009).

Para este trabalho foi desenvolvida a análise das equações de Newton-Euler em robôs manipuladores, como o objetivo é o ambiente de simulação, as equações foram descritas na forma matricial conforme a seguir.

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \quad (2)$$

sendo que $M(\theta)$ representa a matriz de inércias, $V(\theta, \dot{\theta})$, é o vetor centrífugo de Coriolis, $G(\theta)$ o vetor em termos da gravidade, θ os valores de posição, $\dot{\theta}$ os valores de velocidade, $\ddot{\theta}$ os valores de aceleração e τ o sinal de torque gerado (CRAIG, 2009).

2.4 Geração de trajetória: espaço de juntas

Para realizar o planejamento no espaço de juntas, é necessário ter conhecimento comportamental da evolução de cada junta ao longo do tempo. O objetivo é observar os valores de posição, velocidade e aceleração, durante o deslocamento da junta do ponto inicial até o final. Tal planejamento pode ser obtido através da seguinte equação:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (3)$$

sendo $\theta(t)$ o polinômio de 3° ordem genérico responsável pela geração da trajetória que calcula os valores de posição. Onde os parâmetros a_0 , a_1 , a_2 e a_3 são responsáveis por formarem uma trajetória “suave” para o robô, em determinado espaço de tempo, em que t representa o tempo em segundos que o manipular tem para chegar na posição final desejada (SANTOS, 2004; CRAIG, 2009).

Ao derivar a Equação (3), em relação ao tempo, obtém-se a equação de velocidade, representada por:

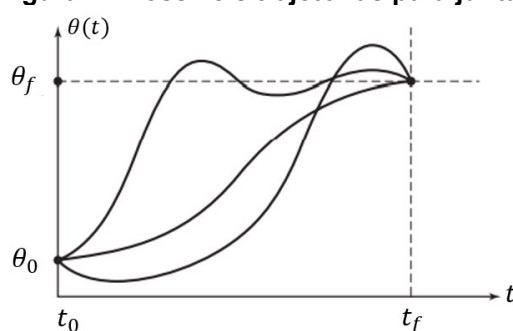
$$\dot{\theta}(t) = a_1 + 2a_2t + 3a_3t^2 \quad (4)$$

Por fim, ao realizar o mesmo procedimento de derivação em relação ao tempo para a Equação (4), tem-se a equação de aceleração a seguir.

$$\ddot{\theta}(t) = 2a_2 + 6a_3t \quad (5)$$

Frequentemente torna-se necessária a movimentação suave do manipulador, deste modo, é preciso que a função utilizada para obtenção dos valores de posição seja contínua, assim como sua derivada primeira e derivada segunda (CRAIG, 2009). Na Figura 7, pode-se notar alguns dos possíveis formatos de trajetórias para uma junta qualquer.

Figura 7 - Possíveis trajetórias para juntas



Fonte: Craig (2009)

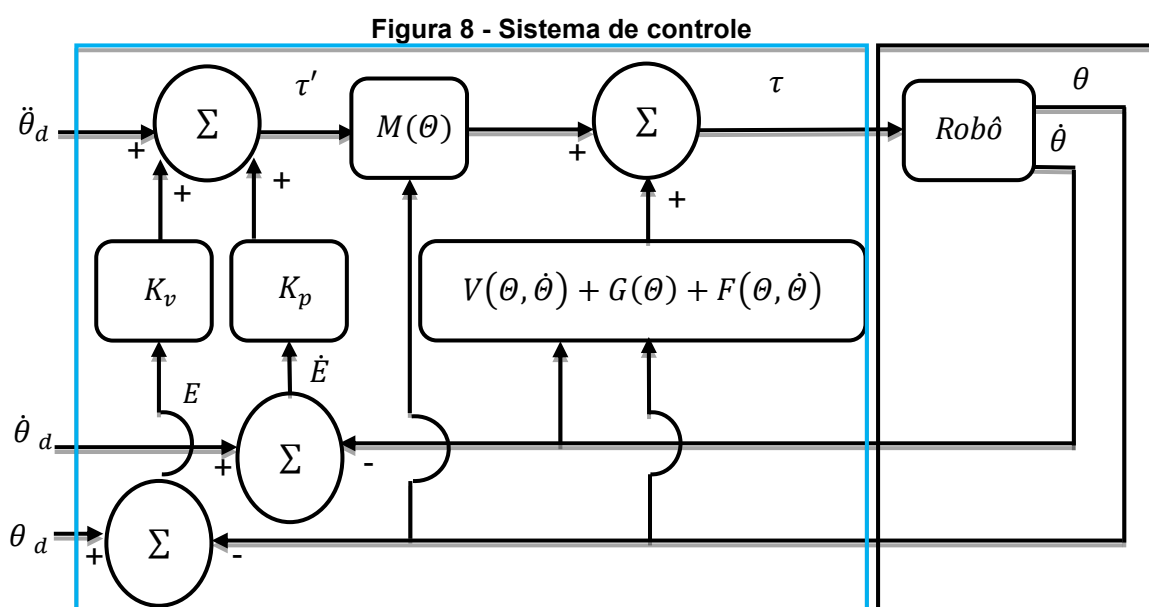
Segundo Craig (2009), movimentações bruscas pelo robô devido um elemento de descontinuidade pode causar desgastes excessivos no mecanismo físico, além de vibrações indesejadas.

2.5 Controle

Mesmo depois da implementação da cinemática e da dinâmica, existem problemas decorrentes da movimentação do robô. Por exemplo, ao imaginar que o controlador envie um sinal para o atuador presente na junta, isso faz com que a mesma adquira uma velocidade e aceleração na direção das coordenadas desejadas. Mesmo implementado um sinal de *feedback* ao sistema, as juntas envolvidas na movimentação podem ultrapassar a posição desejada, o que pode ocasionar em um movimento de vaivém nas juntas, até que atinjam a posição final desejada (NIKU, 2020).

No pior caso, com um sistema instável as oscilações aumentam, o que pode vir a causar o colapso do sistema. Como solução, pode-se adotar um sistema de controle que atue sobre o sinal de tensão ou corrente enviado para o atuador, a fim de evitar ou diminuir o impacto de *overshoot* no sistema (NIKU, 2020).

Na Figura 8, apresenta-se um sistema de controle particionado baseado em modelo e no espaço de juntas do manipulador. Nessa proposta de controle, as entradas do sistema são a posição, a velocidade e a aceleração no espaço de juntas. Localizado na região interior ao quadrado azul está controle particionado, que leva em consideração as características dinâmicas do manipulador (CRAIG, 2009). No interior do retângulo em preto tem-se o sistema físico (planta) do robô, que compõe o sistema de controle baseado em modelo para realização de simulações.



Fonte: Adaptado de Craig (2009)

Sendo os parâmetros da Figura 8 listados a seguir:

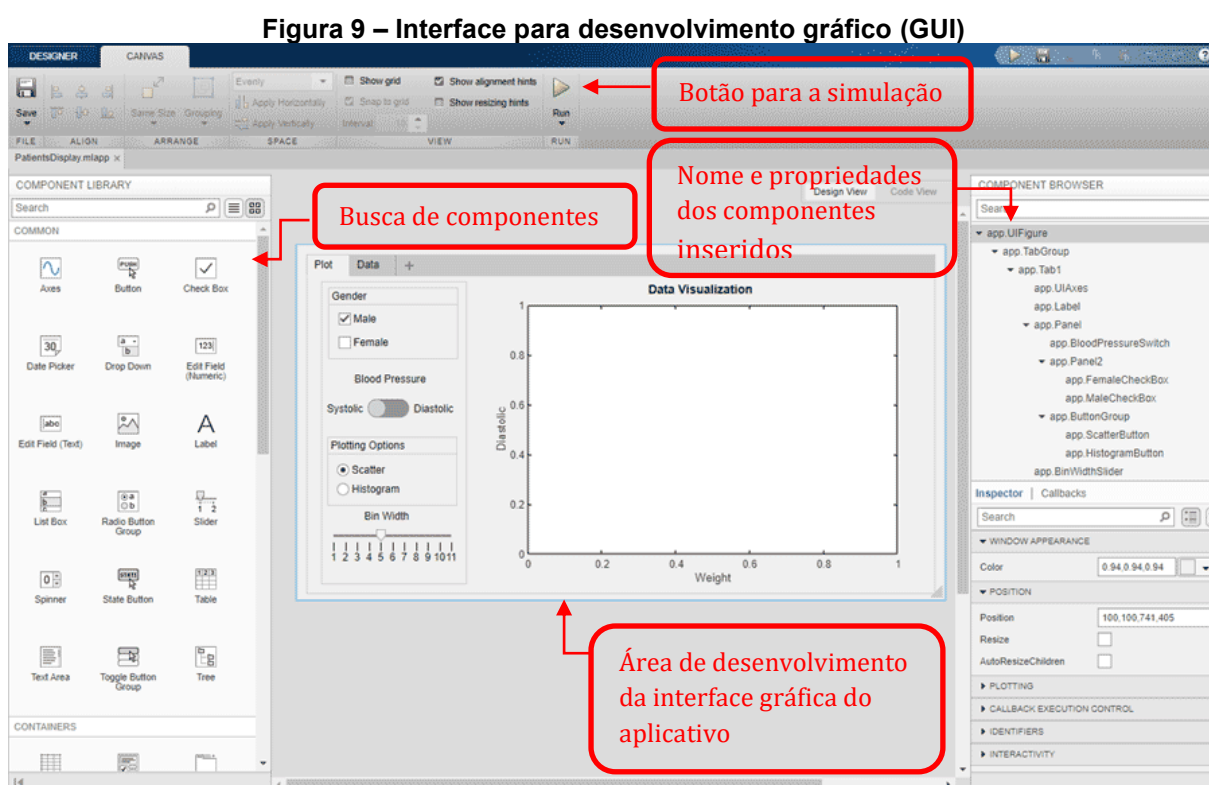
- $\ddot{\theta}_d$: é um vetor que contém os valores de aceleração final desejada para cada junta;
- $\dot{\theta}_d$: é um vetor que contém os valores de velocidade final desejada para cada junta;
- θ_d : é um vetor que contém os valores de posição final desejada para cada junta;
- \dot{E} : é um vetor que possui os valores do erro de velocidade para cada junta, dado por: $\dot{E} = \dot{\theta}_d - \dot{\theta}_a$, sendo $\dot{\theta}_a$ é um vetor com os valores das velocidades atual de cada junta do robô;

- E : é um vetor que possui os valores do erro de posição para cada junta dado por: $E = \theta_d - \theta_a$, sendo θ_a é um vetor com os valores das posições atual de cada junta do robô;
- K_v e K_p : os ganhos de velocidade e posição, respectivamente, do sistema de controle;
- *Robô*: planta física do manipulador (robô real);
- τ' : a lei do servo;
- τ : a equação em malha aberta.

2.6 Ambiente de desenvolvimento (*App Designer*)

O *App Designer* é uma ferramenta de desenvolvimento de aplicativos para execução em *desktop* ou aplicação em servidores *web* gerados com o *MATLAB Compiler™* e o *Simulink Compiler™* (MATHWORKS, 2022).

A ferramenta se divide em duas partes principais: A área de desenvolvimento da interface gráfica, *Graphical User Interface (GUI)* (MARTHA; LOPES; RANGEL, 2021); e a área de codificação, onde o desenvolvedor programa o comportamento da interface gráfica (MARTHA; LOPES; RANGEL, 2021). Na Figura 9, apresenta-se a interface GUI.

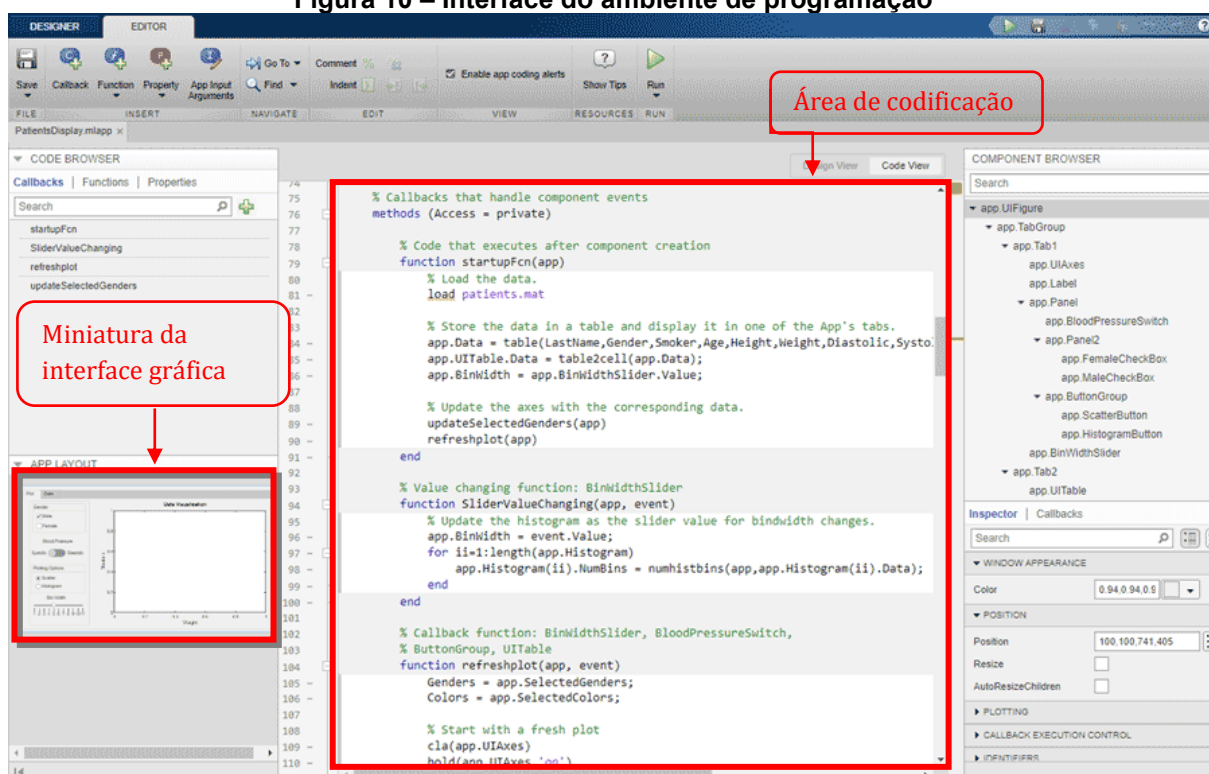


Fonte: Adaptado de Mathworks (2022)

Para adicionar um componente visual basta arrastar o componente desejado do campo *Component Library*, localizado à direita na Figura 9, para a área de *design*, localizada ao centro da Figura 9. Esse procedimento realizado, o *App Designer* automaticamente cria um código orientado a objeto que representa o *layout* do design do aplicativo de acordo com os componentes visuais inseridos na GUI. O aplicativo possui uma biblioteca de componentes padrões, na qual constam lâmpadas, botões, interruptores, caixas de texto, menus, entre outros componentes (MATHWORKS, 2022).

Com a parte gráfica definida, o projetista pode partir para uma segunda tarefa, estabelecer as relações e funcionalidades dos componentes da interface gráfica. Por meio do código orientado a objeto, o programador consegue configurar cada elemento visual inserido na etapa anterior (MARTHA; LOPES; RANGEL, 2021). Na Figura 10, ilustra-se a imagem com um exemplo desse ambiente utilizado para programação.

Figura 10 – Interface do ambiente de programação



Fonte: Mathworks (2022)

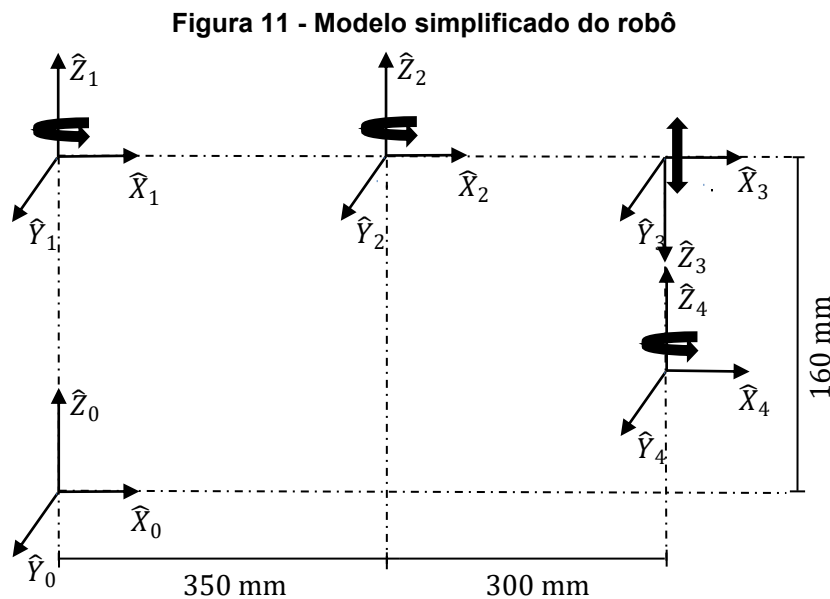
A partir da simulação do código, pode-se analisar os comportamentos descritos para os componentes utilizados na GUI (MATHWORKS, 2022).

3 METODOLOGIA

O desenvolvimento do projeto se deu no *software* de simulação MATLAB® R2021a, no qual foram implementadas rotinas para a obtenção, simulação e análise das equações cinemáticas, dinâmicas, geração de trajetórias e controle desenvolvidas para um robô de configuração SCARA. Neste capítulo é apresentada a metodologia aplicada para obtenção das equações de cada etapa de desenvolvimento do trabalho.

3.1 Modelagem da cinemática direta

De acordo com os passos definidos por Santos (2004), primeiramente foi definida, uma posição inicial para o robô e fixado os sistemas de coordenadas de cada elo, conforme ilustrado na Figura 11.



Fonte: Autoria própria (2022)

Posteriormente sucedeu a transcrição dos pontos no espaço e matrizes de rotação entorno do eixo Z, de cada um dos sistemas de coordenadas, em relação ao sistema anterior, com $i = 0, 1, 2$ e 3.

$${}^i P_{i+1} = [p_x \ p_y \ p_z]^T \quad (6)$$

sendo p_x o valor ao longo do eixo x , p_y o valor ao longo do eixo y e p_z valor ao longo do eixo z , portanto, ${}^iP_{i+1}$ representa o ponto onde está localizado no sistema de referência $\{i + 1\}$, com relação ao sistema de referência anterior $\{i\}$ do manipulador.

Outro passo importante está no cálculo das matrizes de rotação. Essas matrizes representam a movimentação do sistema $\{i + 1\}$, em análise com relação ao sistema anterior $\{i\}$ e z o eixo de rotação do sistema $\{i + 1\}$. Foi definido na Figura 11, que os sistemas de rotação para o robô SCARA giram em torno do eixo z . Desta forma, as matrizes de rotação para o manipulador SCARA podem ser obtidas pela equação:

$${}_{i+1}R_z(\theta) = \begin{bmatrix} \cos \theta & -\text{sen } \theta & 0 \\ \text{sen } \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

sendo θ o ângulo de rotação e $i = 0, 1, 2$ e 3 .

Com o vetor de pontos (Equação (6)) e as matrizes de rotação (Equação (7)), concatenou-se as matrizes de rotação de cada sistema com a sua respectiva posição na forma:

$${}_{i+1}T = \begin{bmatrix} {}_{i+1}R & {}^iP_{i+1} \\ 0 & 1 \end{bmatrix} \quad (8)$$

sendo ${}_{i+1}T$ a matriz de transformação homogênea do sistema de referência $\{i + 1\}$, para o sistema anterior $\{i\}$.

Dessa forma, com $i = 0, 1, 2$ e 3 , obtém-se as matrizes de transformação homogêneas de cada sistema para o sistema anterior. Ao aplicar esses resultados na Equação (1), tem-se a transformação equivalente dada por:

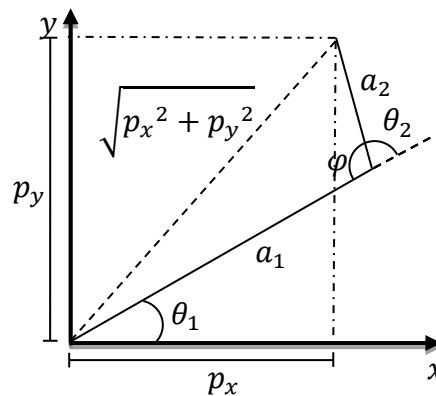
$${}^0T = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 \quad (9)$$

sendo que a matriz 0T contém os valores de posição final do efetuador em relação a base, ou seja, a cinemática direta do robô.

3.2 Modelagem cinemática inversa

Por meio das possibilidades de movimentação das juntas e elos no espaço cartesiano de robôs do tipo SCARA, foi possível obter as equações da cinemática inversa através de análises trigonométricas. A solução trigonométrica para esse manipulador se deu pela análise da Figura 12.

Figura 12 - Vista superior do manipulador



Fonte: Autoria própria (2022)

Na Figura 12, nota-se um triângulo oblíquo, no qual ao aplicar as leis dos cossenos obtém-se a seguinte equação:

$$(\sqrt{p_x^2 + p_y^2})^2 = a_1^2 + a_2^2 - 2a_1a_2\cos\varphi \quad (10)$$

De acordo com a Figura 12 pode-se escrever que $\theta_2 + \varphi = 180^\circ$, logo, $\varphi = 180^\circ - \theta_2$. Portanto a Equação (12) pode ser descrita como:

$$p_x^2 + p_y^2 = a_1^2 + a_2^2 - 2a_1a_2(\cos(180^\circ - \theta_2)) \quad (11)$$

Ao utilizar identidades trigonométricas percebe-se que $\cos(\pi - \theta) = -\cos\theta$, portanto, ao substituir a identidade trigonométrica na Equação (11), e ao atribuir sua expressão a uma variável auxiliar D - com intenção de simplificar reescrituras posteriores - chega-se em:

$$D = \cos\theta_2 = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1a_2} \quad (12)$$

Ao realizar mais algumas manipulações tem-se que $\sin\theta = \sqrt{1 - \cos^2\theta}$ e $\tan\theta = \frac{\sin\theta}{\cos\theta}$, faz-se às devidas substituições da Equação (12), utilizando as identidades trigonométricas com o propósito de isolar θ_2 , e obtém-se a seguinte equação:

$$\tan\theta_2 = \left[\pm \frac{\sqrt{1 - D^2}}{D} \right] \quad (13)$$

A partir da Equação (13), encontra-se θ_2 pela equação:

$$\theta_2 = \tan^{-1} \left[\pm \frac{\sqrt{1 - D^2}}{D} \right] \quad (14)$$

sendo a Equação (14) a expressão que retorna o valor angular de θ_2 . Porém é possível constatar que se tem duas soluções para θ_2 , ambas corretas. Por conveniência, para a implementação no aplicativo foi utilizada a equação que considera o sinal positivo para solução de θ_2 .

Para encontrar a expressão de θ_1 (ângulo esse apresentado na Figura 12), foram analisadas as relações trigonométricas entre o ângulo formado entre o eixo x e o elo a_1 , e o ângulo formado entre a linha pontilhada que sai da base do sistema de referência até ao efetuador. Ao realizar as devidas manipulações, tem-se a equação para θ_1 :

$$\theta_1 = \tan^{-1} \left[\frac{p_y}{p_x} \right] - \tan^{-1} \left[\frac{a_2 \sin\theta_2}{a_1 + a_2 \cos\theta_2} \right] \quad (15)$$

Com as equações que determinam os valores de θ_1 e θ_2 , a solução para θ_4 pode ser obtida através da seguinte equação:

$$\theta_4 = \theta_1 + \theta_2 - \tan^{-1} \left[\frac{r_{22}}{r_{21}} \right] \quad (16)$$

sendo r_{22} e r_{21} os coeficientes provenientes da matriz de rotação 0_4R (SPONG *et al.*, 2006).

3.3 Obtenção das equações dinâmicas

Para o estudo sobre o comportamento dinâmico são necessários conhecimentos sobre o vetor de torque nas juntas, obtidos através da Equação (2).

Dessa forma, o comportamento dinâmico pode ser analisado com a aplicação de diferentes valores de torques nas juntas do mecanismo.

3.3.1 Tensores de Inércia

Como passo inicial, antes da implementação do algoritmo de Newton-Euler, precisa-se conhecer os valores dos tensores de inércia em relação ao centro de gravidade (CRAIG, 2009), da forma:

$${}^c I_i = \begin{bmatrix} {}^c I_{xx_i} & 0 & 0 \\ 0 & {}^c I_{yy_i} & 0 \\ 0 & 0 & {}^c I_{zz_i} \end{bmatrix} \quad (17)$$

sendo os parâmetros ${}^c I_{xx_i}$, ${}^c I_{yy_i}$ e ${}^c I_{zz_i}$ denominados de momentos de inércias das massas de cada corpo móvel de acordo com seu próprio sistema de referência, obtidas pelo conjunto de integrais triplas computadas por meio do centro de gravidade para cada corpo móvel, calculadas da forma:

$${}^c I_{xx_i} = \iiint_V (y^2 + z^2) \rho dv \quad (18)$$

$${}^c I_{yy_i} = \iiint_V (x^2 + z^2) \rho dv \quad (19)$$

$${}^c I_{zz_i} = \iiint_V (x^2 + y^2) \rho dv \quad (20)$$

sendo ρ é a constante de densidade do material que é constituído o elo, padronizado para o valor $\rho = 1$ no código do aplicativo.

3.3.2 Algoritmo iterativo de Newton-Euler

A partir dos resultados da cinemática direta e inversa calculados nas seções anteriores, assume-se que os valores de posição, velocidade e aceleração (θ , $\dot{\theta}$ e $\ddot{\theta}$) são conhecidos. O que torna possível estimar o torque necessário para mover cada junta através do algoritmo de Newton-Euler dividido em duas etapas, “iterações para fora” e “iterações para dentro” (CRAIG, 2009).

A primeira etapa é constituída de iterações para fora, com $i = 0, 1, 2$ e 3 . Expressa pelas equações apresentadas a seguir.

Para a velocidade angular:

$${}^{i+1}\omega_{i+1} = {}^{i+1}R {}^i\omega_i + \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (21)$$

sendo ω a velocidade angular; R a matriz de rotação, $\dot{\theta}$ a velocidade e \hat{Z} o vetor que representa o eixo de rotação para as juntas rotativas e o deslocamento para juntas prismáticas (CRAIG, 2009).

Para a aceleração angular:

$${}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}R {}^i\dot{\omega}_i + {}^{i+1}R {}^i\omega_i \times \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (22)$$

sendo $\dot{\omega}$ a aceleração angular, R a matriz de rotação, ω é a velocidade angular, $\dot{\theta}$ a velocidade e \hat{Z} o vetor que representa o eixo de rotação para as juntas rotativas e o deslocamento para juntas prismáticas (CRAIG, 2009).

Para a aceleração linear para juntas rotativas:

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}R [{}^i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^i\dot{v}_i] \quad (23)$$

sendo \dot{v} é a aceleração linear, R a matriz de rotação, $\dot{\omega}$ a aceleração angular, P a localização do sistema de referência do corpo móvel e ω a velocidade angular (CRAIG, 2009).

Para a aceleração linear (juntas prismáticas):

$${}^{i+1}\dot{v}_{i+1} = {}^{i+1}R [{}^i\dot{\omega}_i \times {}^iP_{i+1} + {}^i\omega_i \times ({}^i\omega_i \times {}^iP_{i+1}) + {}^i\dot{v}_i] + \dots$$

$$\dots + 2^{i+1} \omega_{i+1} \times \dot{d}_{i+1} {}^{i+1}\hat{Z}_{i+1} + \ddot{d}_{i+1} {}^{i+1}\hat{Z}_{i+1}] \quad (24)$$

sendo \dot{v} a aceleração linear, R a matriz de rotação, $\dot{\omega}$ a aceleração angular, P a localização do sistema de referência do corpo móvel, ω a velocidade angular, \dot{d} a velocidade, \ddot{d} a aceleração e \hat{Z} o vetor que representa o eixo de deslocamento da junta prismática (CRAIG, 2009).

Para a aceleração do centro de gravidade:

$${}^{i+1}\dot{v}_{C_{i+1}} = {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{i+1}) + {}^{i+1}\dot{v}_{i+1} \quad (25)$$

sendo \dot{v}_c a aceleração do centro de gravidade, $\dot{\omega}$ a aceleração angular, P_c a localização dos centros de gravidade dos corpos móveis, ω a velocidade angular, P a localização do sistema de referência do corpo móvel, \dot{v} a aceleração linear (CRAIG, 2009).

Para a força:

$${}^{i+1}F_{i+1} = m_{i+1} {}^{i+1}\dot{v}_{C_{i+1}} \quad (26)$$

sendo F a força, m a massa do corpo móvel e \dot{v}_c a aceleração do centro de gravidade (CRAIG, 2009).

Para o torque:

$${}^{i+1}N_{i+1} = {}^{c_{i+1}}I_{i+1} {}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{c_{i+1}}I_{i+1} {}^{i+1}\omega_{i+1} \quad (27)$$

sendo N o torque, cI o tensor de inércia do corpo, $\dot{\omega}$ a aceleração angular e ω a velocidade angular (CRAIG, 2009).

Com a primeira etapa concluída, tem-se a segunda etapa de cálculos “de fora para dentro”, onde $i = 4, 3, 2$ e 1 , representa nas equações:

Para a decomposição da força:

$${}^i f_i = {}_{i+1}^i R {}^{i+1} f_{i+1} + {}^i f_i \quad (28)$$

sendo f a força exercida sobre o elo e R a matriz de rotação (CRAIG, 2009).

Para a decomposição de torque:

$${}^i n_i = {}^i N_i + {}_{i+1}^i R^{i+1} n_{i+1} + {}^i P_{c_i} \times {}^i F_i + {}^i P_{i+1} \times {}_{i+1}^i R^{i+1} f_{i+1} \quad (29)$$

sendo n o torque exercido sobre o elo, N o torque, R a matriz de rotação, P_c a localização do centro de gravidade do corpo móvel e f a força exercida sobre o elo (CRAIG, 2009).

Para o torque em juntas rotativas:

$$\tau = {}^i n_i^T {}^i \hat{Z}_i \quad (30)$$

sendo τ o vetor de torque, n o torque exercido sobre o elo e \hat{Z} o vetor que representa o eixo de rotação (CRAIG, 2009).

Para o torque (juntas prismáticas):

$$\tau = {}^i f_i^T {}^i \hat{Z}_i \quad (31)$$

sendo τ o vetor de torque, f a força exercida sobre o elo e \hat{Z} o vetor que representa o eixo de deslocamento da junta prismática (CRAIG, 2009).

Chega-se assim, a um conjunto de equações, as quais serão isoladas e reescritas no formato da Equação (2), para simulação no *software* MATLAB® R2021a.

3.4 Geração de trajetória

Segundo Craig (2009), há quatro restrições para o polinômio de terceira ordem escolhido, sendo elas:

1. $\theta(0) = \theta_0$ a restrição referente ao valor inicial da posição;
2. $\theta(t_f) = \theta_f$ a restrição referente ao valor final da posição, sendo t_f é o tempo que a junta tem para chegar na posição desejada;
3. $\dot{\theta}(0) = 0$ a restrição referente ao valor inicial da velocidade;
4. $\dot{\theta}(t_f) = 0$ a restrição referente ao valor final da velocidade, sendo t_f é o tempo que a junta tem para chegar na posição desejada.

A combinação das restrições com o polinômio de terceira ordem, Equação (3), e isolando as equações em relação aos parâmetros a_i , com $i = 0, 1, 2$ e 3 , geram as equações:

$$a_0 = \theta_0 \quad (32)$$

$$a_1 = 0 \quad (33)$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) \quad (34)$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) \quad (35)$$

nas quais através das entradas dos valores de tempo, posição inicial e final, acerta-se o polinômio que interliga qualquer posição final e inicial dos ângulos das juntas do robô, responsável por gerar uma trajetória “suave” para cada junta do robô (CRAIG, 2009).

3.5 Modelagem do sistema de controle

O desenvolvimento do sistema de controle se deu pela utilização do modelo já discutido na Equação (2) e ilustrado na Figura 8. A estratégia de controle é baseado no controle particionado, sendo a equação característica de erro em malha fechada, expressa por:

$$\tau = \alpha\tau' + \beta \quad (36)$$

sendo $\alpha = M(\theta)$, $\beta = V(\theta, \dot{\theta}) + G(\theta) + F(\theta, \dot{\theta})$, sendo que $F(\theta, \dot{\theta})$ representa o modelo de atrito (CRAIG, 2009).

A equação para a lei do servo para o referido sistema de controle, representada por:

$$\tau' = \ddot{\theta}_d + K_v\dot{E} + K_pE \quad (37)$$

sendo τ' a lei de servo, $\ddot{\theta}_d$ a aceleração final desejada, K_v e K_p os ganhos do sistema, que por sua vez são matrizes diagonais, e E o erro de posição e \dot{E} o erro de velocidade (CRAIG, 2009).

Por fim a equação característica de erro em malha fechada:

$$\ddot{E} + K_v\dot{E} + K_pE = 0 \quad (38)$$

sendo \ddot{E} o erro de aceleração (CRAIG, 2009).

Os valores de K_v e K_p são ganhos constantes de erro de velocidade e posição respectivamente, logo, podem ser calculados por meio da Equação (38).

4 RESULTADOS E DISCUSSÕES

Neste capítulo serão descritos os resultados obtidos através das simulações realizadas no *software* MATLAB® R2021a. Inicialmente é descrita a interface do aplicativo e posteriormente os resultados de alguns cálculos e simulações realizados pela aplicação.

Vale ressaltar que todos os resultados são baseados no robô SCARA SR-6 iA, porém o aplicativo pode fazer a representação gráfica da resposta para qualquer manipulador SCARA independente do fabricante.

4.1 Aplicativo

O aplicativo possui duas abas sendo elas: “Cinemática” e “Dinâmica e Controle”. Na primeira aba são apresentados os resultados das cinemáticas direta e inversa além da geração de trajetória, a partir dos dados fornecidos pelo usuário.

A interface dessa primeira parte do aplicativo pode ser separada em três módulos principais: seleção de visualização, inserção de dados e representação gráfica. Na Figura 13, pode-se observar os três módulos da interface.

Figura 13 - Tela inicial do aplicativo (aba “Cinemática”)

The screenshot shows the 'Cinematic' tab of the application. It features a sidebar on the left with three main sections: 'Cinemática', 'Visualização', and 'Juntas'. The 'Cinemática' section has radio buttons for 'Direta' (selected), 'Inversa', and 'Trajetória'. The 'Visualização' section has radio buttons for 'Deslocamento' (selected), 'Velocidade', and 'Aceleração'. The 'Juntas' section has radio buttons for 'Junta 1' (selected), 'Junta 2', 'Junta 3', and 'Junta 4'. Below these is an 'Inf. Complementares' section. The main area is divided into three input sections: 'Comprimento dos Elos (mm)' with fields for E1, E2, and E3; 'Cinemática Direta' with fields for Theta 1, Theta 2, d3, and Theta 4; and 'Cinemática Inversa (mm)' with a position [x,y,z] field. Below these is a 'Geração de Trajetória (mm)' section with fields for initial and final positions and time/step. A legend identifies four links: Elo 1/Peça 1 (black), Elo 2/Peça 2 (yellow), Elo 3/Peça 3 (blue), and Elo 4/Peça 4 (red). A diagram shows a SCARA robot arm with joints E1, E2, E3, and E4. On the right, two empty plots with axes from 0 to 1 are titled 'Aguardando a inserção de parâmetros'.

Fonte: Autoria própria (2022)

Na Figura 13, encontra-se um retângulo de cor amarela à esquerda, o qual destaca a área de visualização, área onde o usuário pode escolher o tipo de análise que almeja realizar os cálculos e visualizar os resultados gráficos.

Na região central, retângulo de cor azul, demonstra-se o módulo de inserção de dados. Nesta área o usuário deve preencher os campos de acordo com as especificações do robô que se deseja analisar.

Mais à direita, no interior do retângulo de cor verde, pode-se observar dois gráficos que se alteram de acordo com a mudança do campo de visualização. O primeiro gráfico, localizado no campo superior direito, é responsável pela geração da imagem isométrica em três dimensões (3D) do robô, conforme as escolhas da cinemática e da área de visualização.

Já o segundo gráfico, localizado no lado esquerdo inferior, é responsável pela representação das respostas das juntas referente a geração de trajetória. Além de mostrar a posição de cada uma das juntas, apresenta os gráficos de velocidade e aceleração, também para cada uma das juntas.

Abaixo do retângulo em amarelo está posicionado o retângulo de cor roxa. No interior desse retângulo está a caixa de texto chamada “Inf. Complementares”. A função dessa caixa de texto é apresentar os valores de posição final do efetuador, ao realizar os cálculos para a cinemática direta. Para a cinemática inversa a caixa de texto apresenta os valores de posição final calculados dos ângulos para cada uma das juntas do manipulador.

Por meio desta caixa de texto o usuário consegue obter o valor de θ_4 , pois como a quarta junta representa apenas uma rotação da junta prismática em torno do seu próprio eixo, portanto, não é possível a sua visualização de deslocamento nas respostas gráficas geradas pelo aplicativo da mesma forma que são representadas as outras juntas do robô.

Na aba “Dinâmica e Controle”, pode-se computar e apresentar a resposta gráfica do sistema de controle particionado, aplicado juntamente ao sistema às características dinâmicas do manipulador. Na Figura 14, ilustra-se a segunda aba do aplicativo.

Figura 14 - Segunda aba do aplicativo “Dinâmica e Controle”

Fonte: Autoria própria (2022)

Em todas as simulações as cores utilizadas para representar os elos são:

- Cor preta: linha que representa o primeiro elo fixo a base;
- Cor amarela: equivale ao segundo elo conectado ao primeiro por intermédio da primeira junta rotacional;
- Cor azul: simboliza o terceiro elo interligado ao segundo através da segunda junta rotativa e possui linkado na sua outra extremidade a junta prismática;
- Cor vermelha: corresponde ao quarto elo (junta prismática) interligada ao terceiro elo.

4.2 Modelagem matemática e simulação cinemática direta

A partir da aplicação da Equação (7), para cada sistema de referência, obtém-se as matrizes de rotação 0R_1 , 1R_2 , 2R_3 , 3R_4 , descritas a seguir.

$${}^0R_z = \begin{bmatrix} \cos \theta_1 & -\text{sen} \theta_1 & 0 \\ \text{sen} \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (39)$$

$${}^1_2R_z = \begin{bmatrix} \cos \theta_2 & -\text{sen} \theta_2 & 0 \\ \text{sen} \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (40)$$

$${}^2_3R_z = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (41)$$

$${}^3_4R_z = \begin{bmatrix} \cos \theta_4 & -\text{sen} \theta_4 & 0 \\ \text{sen} \theta_4 & \cos \theta_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (42)$$

Ao utilizar as Equações (39) a (42) e a Equação (8), apresentam-se às transformações 0_1T , 1_2T , 2_3T e 3_4T . São expostas a seguir as equações que representam as relações entre os sistemas de referência do robô, a tomar em consideração a distribuição apresentada na Figura 11.

$${}^0_1T = \begin{bmatrix} \cos \theta_1 & -\text{sen} \theta_1 & 0 & 0 \\ \text{sen} \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (43)$$

$${}^1_2T = \begin{bmatrix} \cos \theta_1 & -\text{sen} \theta_1 & 0 & d_1 \\ \text{sen} \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (44)$$

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & d_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (45)$$

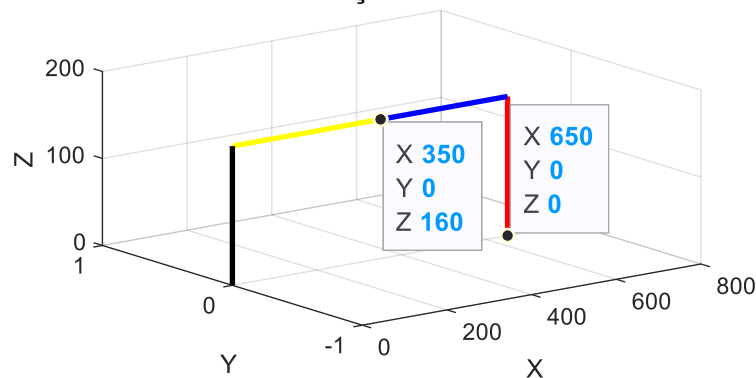
$${}^3_4T = \begin{bmatrix} \cos \theta_1 & -\text{sen} \theta_1 & 0 & 0 \\ \text{sen} \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_0 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (46)$$

A partir da substituição das matrizes de transformação na Equação (9), obtém-se a matriz 0_4T - Apêndice B - que retorna o valor da posição do efetuador em relação a base do manipulador. Com esse resultado e as Equações (43) a (46),

implementadas no *software* MATLAB® R2021a, foram realizadas às simulações para a cinemática direta.

Na Figura 15, tem-se o gráfico em 3D decorrente da inserção dos valores $E_1 = 160 \text{ mm}$, $E_2 = 350 \text{ mm}$ e $E_3 = 300 \text{ mm}$ (valores dos elos do robô SCARA SR-6 iA), no aplicativo, com a opção da cinemática direta selecionada. Essa simulação tem por objetivo analisar a posição inicial das juntas do robô, ao verificar se a imagem gráfica gerada pela aplicação desenvolvida condiz com a posição idealizada na Figura 11, com apenas o deslocamento em d_3 . No Apêndice A – Item 1, está localidade à vista laretal para os eixos x e z .

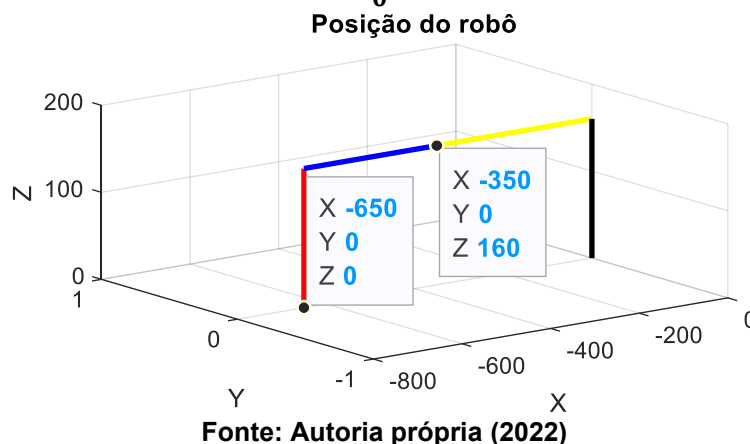
Figura 15 - Resposta cinemática direta para as entradas $\theta_1 = 0^\circ$, $\theta_2 = 0^\circ$, $d_3 = 0 \text{ mm}$ e $\theta_4 = 0^\circ$
Posição do robô



Fonte: Autoria própria (2022)

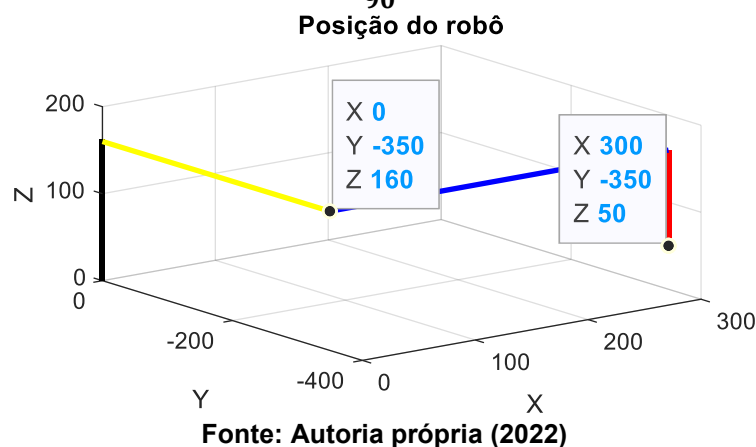
A segunda simulação para a cinemática direta, exposta na Figura 16, vem com intuito de verificar a movimentação do manipulador. Demonstra a representação do manipulador para a posição contrária a sua posição inicial no eixo x , ou seja, uma rotação de 180° da primeira junta. Para essa simulação foram mantidos os valores de $E_1 = 160 \text{ mm}$, $E_2 = 350 \text{ mm}$ e $E_3 = 300 \text{ mm}$, e no campo “Cinemática” foi selecionada a opção “Direta”, assim como a simulação anterior a vista lateral do manipulador se encontra no Apêndice A – item 2.

Figura 16 - Resposta cinemática direta para as entradas $\theta_1 = 180^\circ$, $\theta_2 = 0^\circ$, $d_3 = 0 \text{ mm}$ e $\theta_4 = 0^\circ$



A representação da terceira e última simulação realizada para verificação dos resultados obtidos na cinemática direta, encontra-se representada na Figura 17. A título de simulação, foram mantidos os valores para E_1 , E_2 e E_3 , e a marcação do campo “Cinemática” das simulações anteriores. O objetivo desta simulação foi analisar o comportamento da junta prismática, assim como, o comportamento de todas as juntas rotativas do robô, em uma mesma simulação.

Figura 17 - Resposta cinemática direta para as entrada $\theta_1 = 270^\circ$, $\theta_2 = 90^\circ$, $d_3 = 50 \text{ mm}$ e $\theta_4 = 90^\circ$



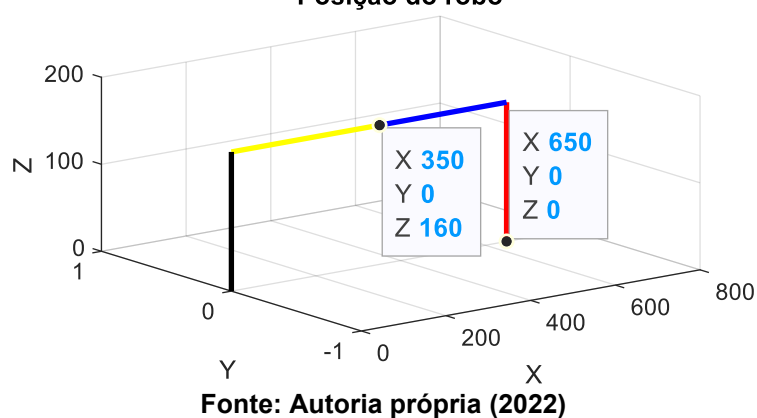
4.3 Simulação cinemática inversa

Para as simulações computacionais para a cinemática inversa, foram tomados como base os valores dos elos referentes ao robô SCARA SR-6 iA, $E_1 = 160 \text{ mm}$, $E_2 = 350 \text{ mm}$ e $E_3 = 300 \text{ mm}$. Conforme os dados utilizados na etapa de verificação das respostas obtidas para a cinemática direta. Porém, para as

simulações desta seção a opção selecionada na caixa cinemática foi a opção “Inversa”.

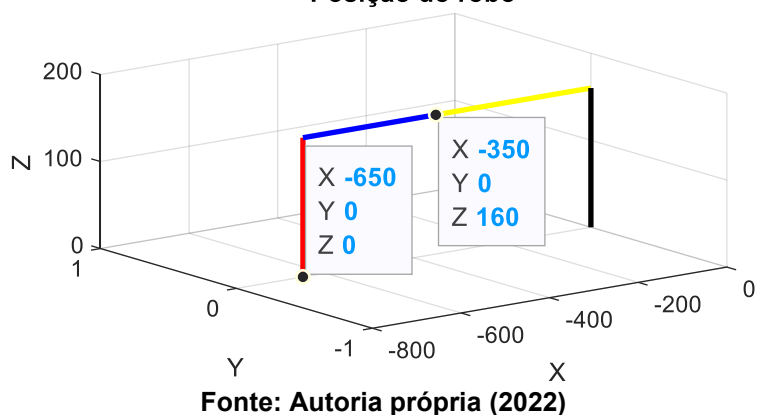
Na Figura 18, apresenta a mesma resposta para a posição final do efetuador obtida na Figura 15, contudo, obtida mediante a inserção do ponto cartesiano no algoritmo do aplicativo, e não mais os ângulos e deslocamento final da junta prismática. No Anexo A – Item 3, pode ser encontrada a imagem da vista lateral no plano x e z , para essa simulação.

Figura 18 - Resposta cinemática inversa para as entradas $x = 650\text{ mm}$, $y = 0\text{ mm}$ e $z = 0\text{ mm}$
Posição do robô



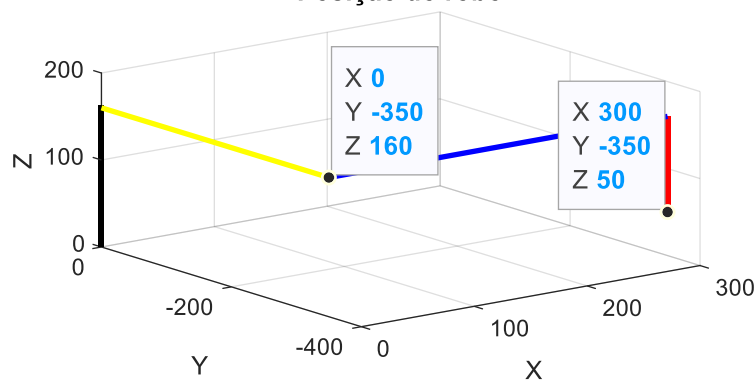
Na Figura 19, manteve-se o mesmo método da simulação anterior. Busca representar a mesma posição final para o efetuador, posição vista pela Figura 16, porém com a inserção do ponto cartesiano no aplicativo.

Figura 19 - Resposta cinemática inversa para as entradas $x = -650\text{ mm}$, $y = 0\text{ mm}$ e $z = 0\text{ mm}$
Posição do robô



Por fim a última simulação, apresentada na Figura 20, representa uma figura análoga à Figura 17, devido à inserção do ponto cartesiano, que representa a resposta exposta na Figura 17.

Figura 20 - Resposta cinemática inversa para as entradas $x = 300 \text{ mm}$, $y = -350 \text{ mm}$ e $z = 50 \text{ mm}$



Fonte: Autoria própria (2022)

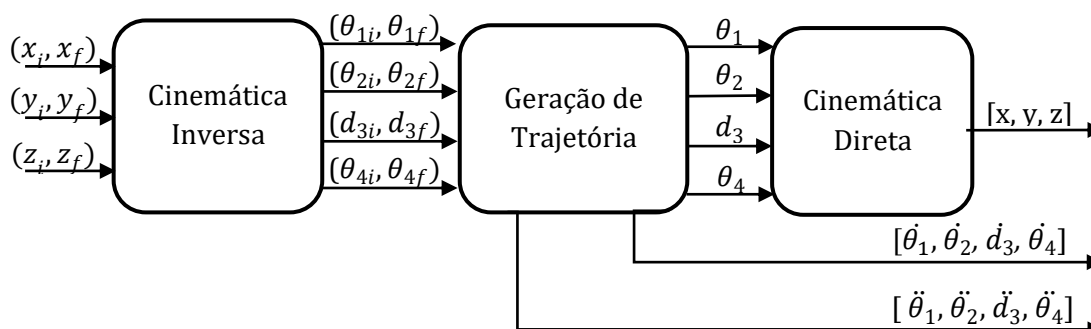
Ao final das simulações da cinemática direta e cinemática inversa, observa-se que os resultados de posição final do efetuador são iguais, quando comparadas as Figuras 15 e 18, as Figura 16 e Figura 19 e as Figuras 17 e 20. Isso porque foram inseridos propositalmente as mesmas posições em ambas às simulações, entretanto, na cinemática direta as entradas da simulação foram os ângulos θ_1 , θ_2 , θ_4 e a posição final de d_3 . Já na cinemática inversa foram inseridos diretamente os valores finais da posição do efetuador no espaço cartesiano x , y e z .

Deste modo, consegue-se analisar e constatar que as modelagens matemáticas para ambas às cinemáticas estão corretas, ou seja, estão coerentes com os resultados esperados, a partir da aplicação de regra da mão direita para análise das respostas obtidas, ao tomar como referente à posição inicial adotada para o manipulador definida na cinemática direta presente na Figura 11.

4.4 Simulação da geração de trajetória

Para a simulação da geração de trajetória foram utilizados os códigos gerados nas etapas anteriores, como apresentado no diagrama de blocos na Figura 21, de modo a configurar três etapas de processamento de dados.

Figura 21 - Diagrama de simulação da geração de trajetória



Fonte: Autoria própria (2022)

Primeiramente, entra-se com as posições inicial e final do manipulador na cinemática inversa que transporta os valores da posição inicial e final do plano cartesiano para o espaço de juntas, através do código desenvolvido no Item 4.1.

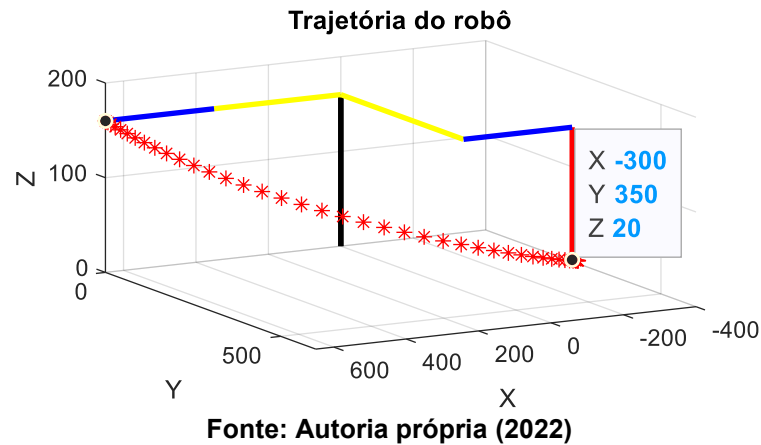
Por conseguinte, esses pontos vão para o algoritmo de geração de trajetória que aplica às Equações (3), (4), e (5), assim como às Equações (32) a (35). Como resultado desse bloco, geração de trajetória, tem-se a posição, a velocidade e a aceleração para cada junta do manipulador estudado.

Por fim, apenas a posição, no espaço de juntas, de cada junta, passa pela cinemática direta, com a finalidade de se obter os valores para a representação isométrica em 3D no espaço cartesiano da posição inicial e final de cada elo do manipulador e as posições do efetuador ao longo da trajetória realizada.

Para as simulações de trajetórias continua-se a usar como base os dados referentes ao manipulador SCARA SR-6 iA. Mantém-se os valores de $E_1 = 160 \text{ mm}$, $E_2 = 350 \text{ mm}$ e $E_3 = 300 \text{ mm}$, todavia, a opção escolhida no campo cinemática foi a opção “Trajetória”.

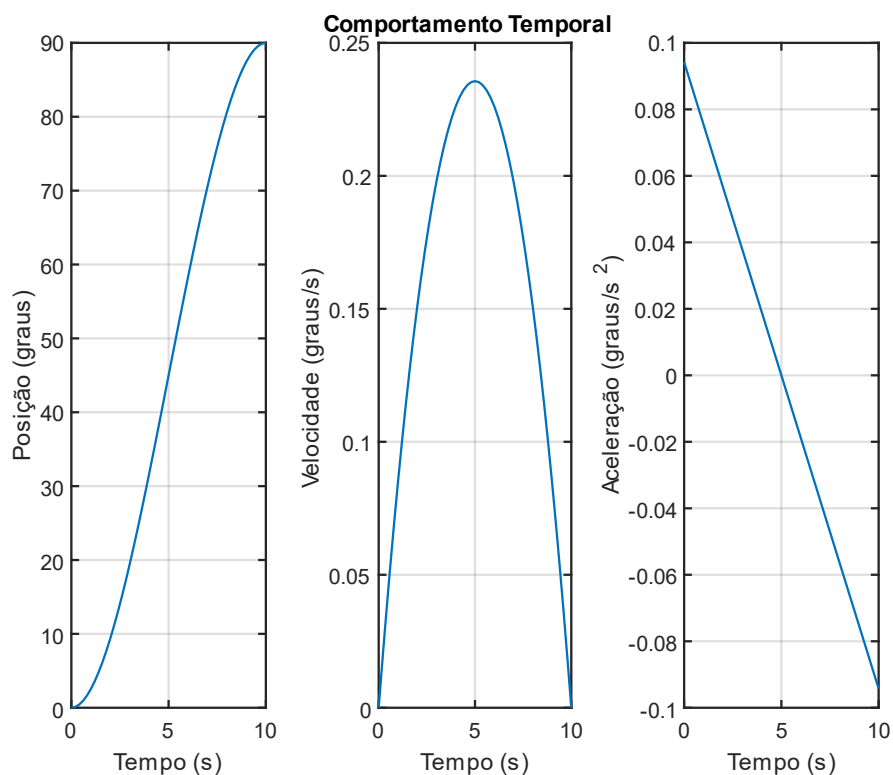
Na Figura 22 é apresentada a trajetória dado o ponto inicial $P_i = [650, 0, 160]^T$ e ponto final $P_f = [-300, 350, 20]^T$, ambos em milímetros. Além do valor inicial e final, foram escolhidos os valores do tempo de simulação igual a 10 s, com passos de 0,2 s, de maneira obter como resposta uma trajetória com 50 pontos de representação da posição final do efetuador ao longo da trajetória. O objetivo desta simulação é observar a trajetória gerada a partir de uma posição inicial inserida pelo usuário e verificar se as respostas gráficas obtidas representam de maneira coerente os resultados esperados decorrentes da implementação das Equações (3), (4) e (5).

Figura 22 – Geração da trajetória para o tempo = 10 s e passos = 0,2 s



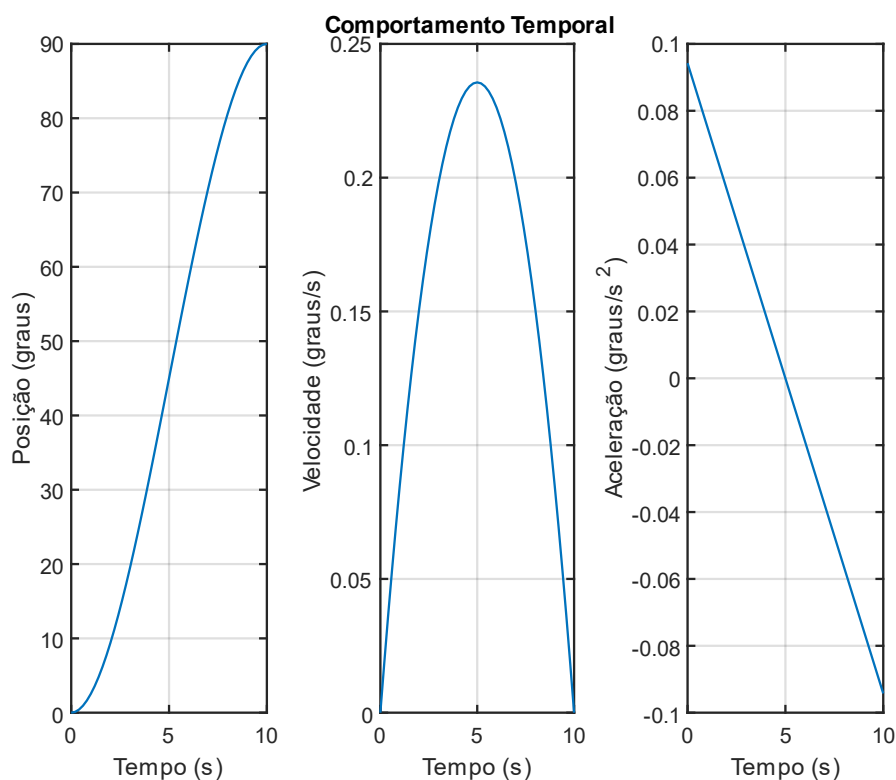
Na Figura 23, tem-se as representações gráficas para os valores calculados de posição, velocidade e aceleração, para a primeira junta rotativa, ao longo do tempo em segundos. Nota-se, o comportamento das respostas gráficas ao longo do tempo ocorre de maneira suave, sem grandes picos em curtos espaços de tempo. O que condiz com o resultado esperado pela implementação do polinômio de terceiro grau genérico e suas respectivas derivadas implementadas ao algoritmo de geração de trajetória.

Figura 23 - Comportamento temporal junta 1



A simulação presente na Figura 24, demonstra os valores de posição, velocidade e aceleração para a segunda junta rotativa. Ao observar a imagem percebe-se que os valores são idênticos aos retratados na Figura 23, sendo correto os valores de ambas as simulações, pois as duas juntas realizam um mesmo deslocamento no valor de 90° , em relação à posição inicial do robô definida na cinemática direta.

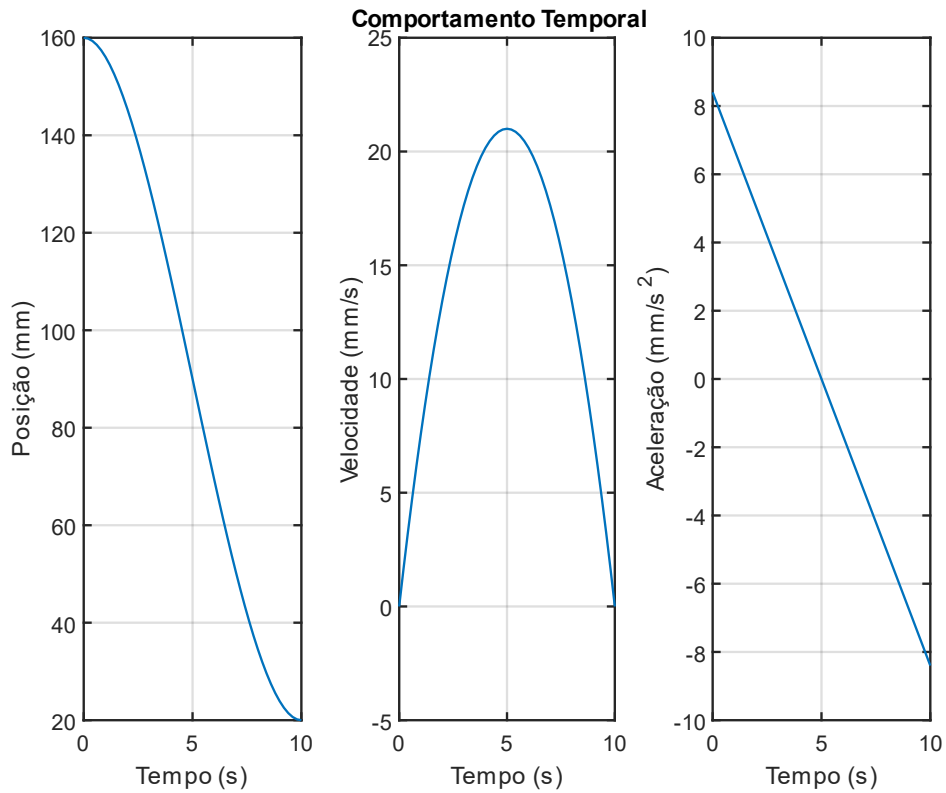
Figura 24 - Comportamento temporal junta 2



Fonte: Autoria própria (2022)

Na Figura 25 são exibidos os valores de posição, velocidade e aceleração para a terceira junta (junta prismática). Nota-se a posição inicial e final da juntas, assim como, os valores intermediários calculados para se chegar a posição idealizada.

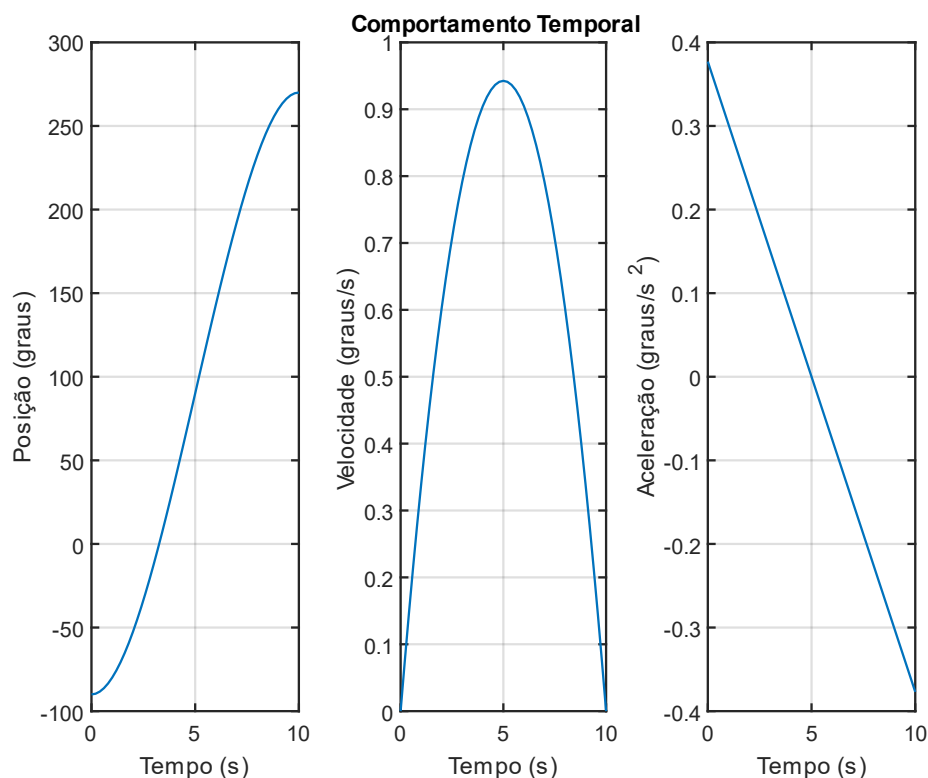
Figura 25 - Comportamento temporal junta 3



Fonte: Autoria própria (2022)

Por fim, tem-se para a Figura 26, onde estão evidenciados os resultados de posição, velocidade e aceleração, para a quarta junta (junta rotativa). Ao observar a junta 4, observa-se que a mesma realiza uma movimentação total de 360° .

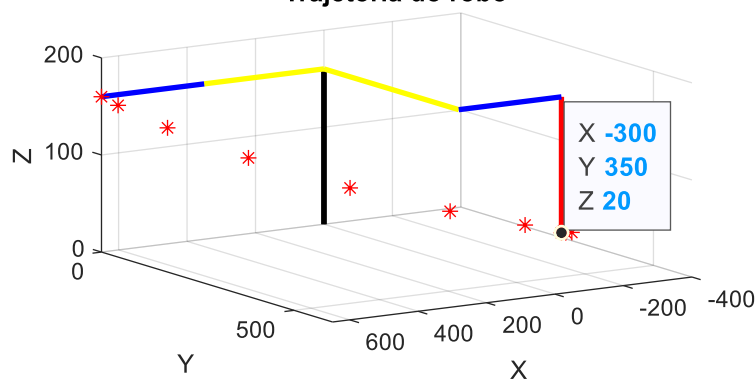
Figura 26 - Comportamento temporal junta 4



Fonte: Autoria própria (2022)

A Figura 27, apresenta os resultados para mais uma simulação para a realização da análise da geração de trajetória do robô. Para a simulação em questão, continuou-se a utilizar-se os mesmos valores de $E_1 = 160 \text{ mm}$, $E_2 = 350 \text{ mm}$ e $E_3 = 300 \text{ mm}$, referentes ao robô SCARA SR-6iA o ponto inicial $P_i = [650, 0, 160]^T$ e ponto final $P_f = [-300, 350, 20]^T$, ambos em milímetros. Todavia, foi modificado o valor de tempo de simulação, sendo alterado para 20 s e mudança no valor do passo para 2 s. O objetivo é analisar a influência do aumento dos valores do tempo de simulação e da variável passo nas respostas geradas pelo aplicativo.

Figura 27 - Geração em 3D da trajetória para o $tempo = 20 s$ e $passos = 2 s$
Trajétória do robô

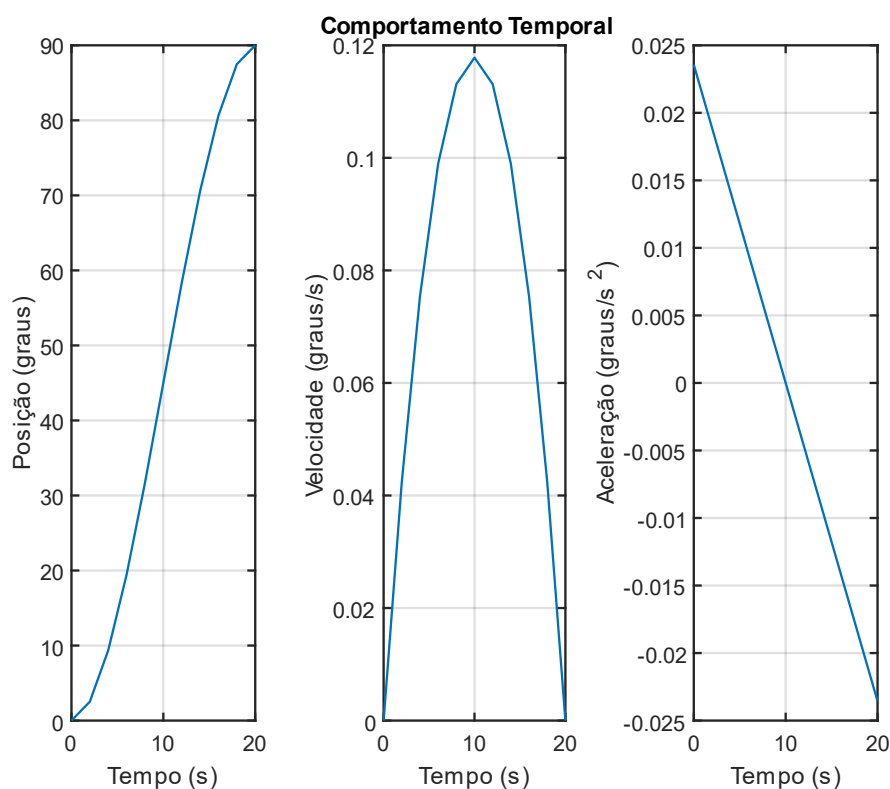


Fonte: Autoria própria (2022)

Ao comparar as respostas apresentadas nas Figuras 27 e 22, verifica-se uma diferença clara na quantidade de pontos apresentados entre as representações. Na Figura 27, tem-se um número menor de pontos representando a trajetória realizada pelo efetuador. Isso devido ao aumento significativo do valor da variável passo, o que faz com que uma menor quantidade de posições finais para o efetuador seja calculada pelo algoritmo de geração de trajetória. Desta forma, pode-se concluir que quanto menor o valor da variável passo melhor é a representação gráfica da trajetória percorrida pelo efetuador, porém o custo computacional é maior.

A simulação temporal da primeira junta rotativa é exposta na Figura 28. Nota-se uma leve diferença do formato dos gráficos, ao comparar a resposta obtida com a Figura 23. A diferença entre as formas se justifica pela diminuição na quantidade de pontos que o *software* possui para plotar a resposta gráfica, devido ao aumento do parâmetro passo.

Figura 28 - Comportamento temporal (primeira junta)



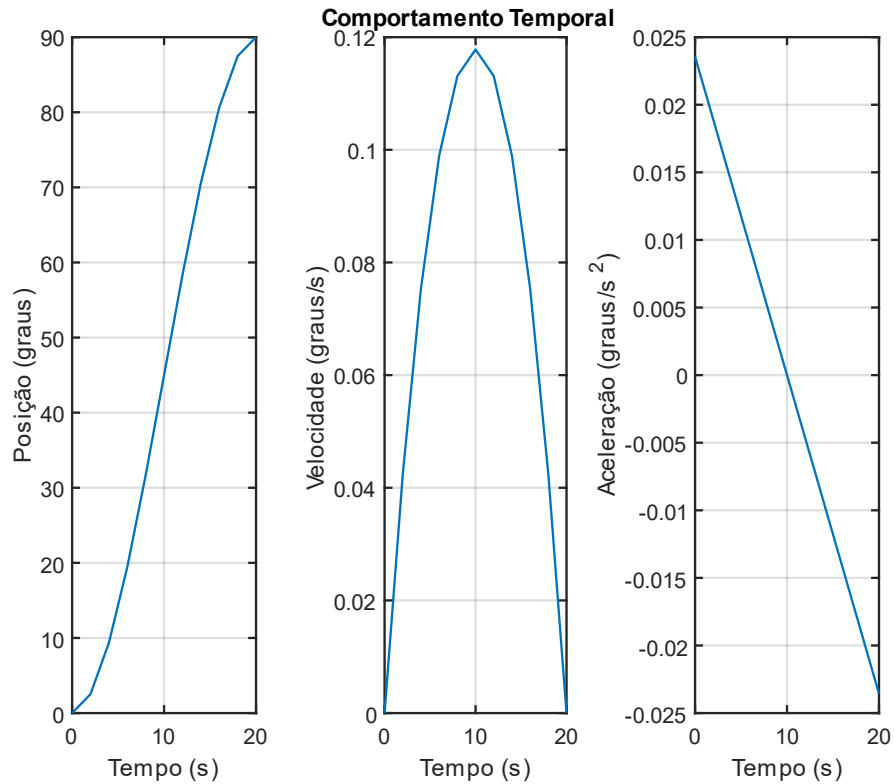
Fonte: Autoria própria (2022)

Nota-se ainda uma variação entre os valores de velocidade e aceleração verificados nas simulações. Para a velocidade há uma diminuição de 0,13 graus/s e para a aceleração uma redução de 0,075 graus/s² - para a resposta presente na Figura 28, em comparação a exposta na Figura 23 - nos valores de pico atingidos. A diminuição nos valores de velocidade e aceleração, são justificados pela alteração no tempo de simulação. Ao aumentar o valor da variável tempo, o mecanismo possui um tempo maior para chegar na posição final, logo, a simulação realizada na Figura 26 precisa de velocidade e aceleração menores para chegar a mesma posição final apresentada na Figura 23.

Ao comparar as simulações realizadas entre as Figura 24 e 29, observa-se formas gráficas com menores valores de velocidade e aceleração. Como já

mencionado, tanto a primeira junta rotativa, quanto a segunda, apresentam o mesmo deslocamento. Diante disso, tem-se exatamente a mesma análise realizada para a comparação entre as respostas apresentadas nas Figuras 23 e 28 para as respostas apresentadas nas Figuras 26 e 29.

Figura 29 - Comportamento temporal (segunda junta)

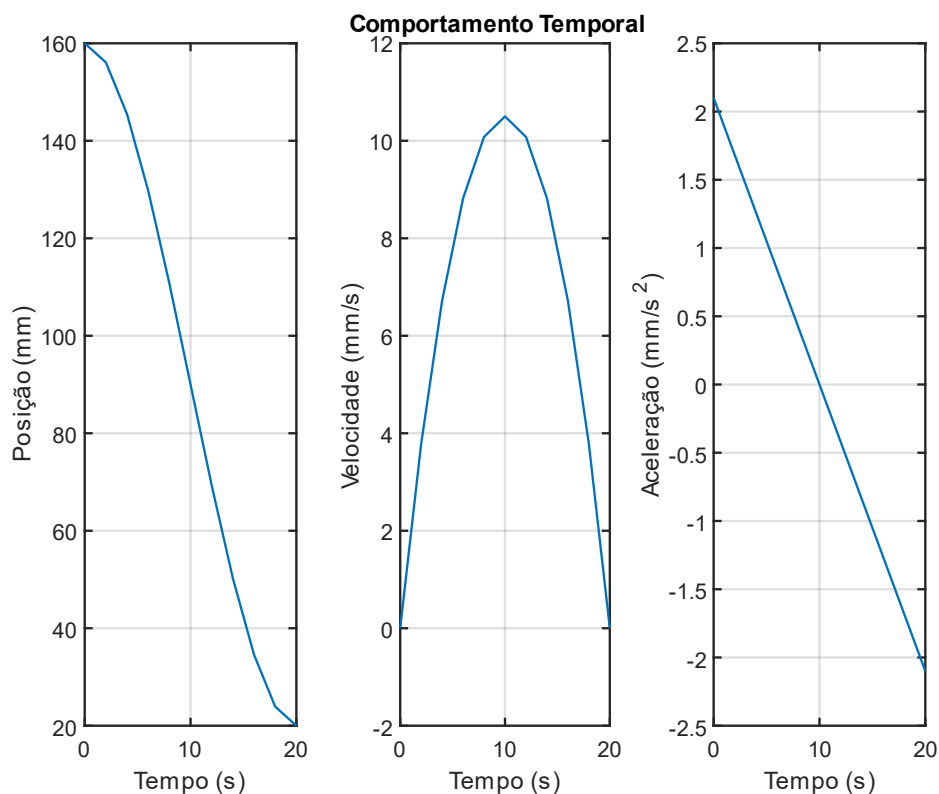


Fonte: Autoria própria (2022)

Ao analisar as representações gráficas expostas na Figura 30, em comparação a resposta apresentada na Figura 25, observa-se o aspecto de formas mais arredondadas na Figura 25, decorrente do menor valor atribuído à variável passo.

Identifica-se também a diminuição dos valores de velocidade e aceleração. Para a velocidade há uma diminuição de 9,7 mm/s e para a aceleração uma redução de 6,2 mm/s² - para a resposta presente na Figura 30, em comparação a exposta na Figura 25 - nos valores de pico atingidos. Logo, assim como nas duas juntas rotativas analisadas anteriormente, a junta prismática precisa de velocidade e aceleração menores para chegar a posição final, isso devido ao tempo maior para se deslocar até a posição final.

Figura 30 - Comportamento temporal (terceira junta)

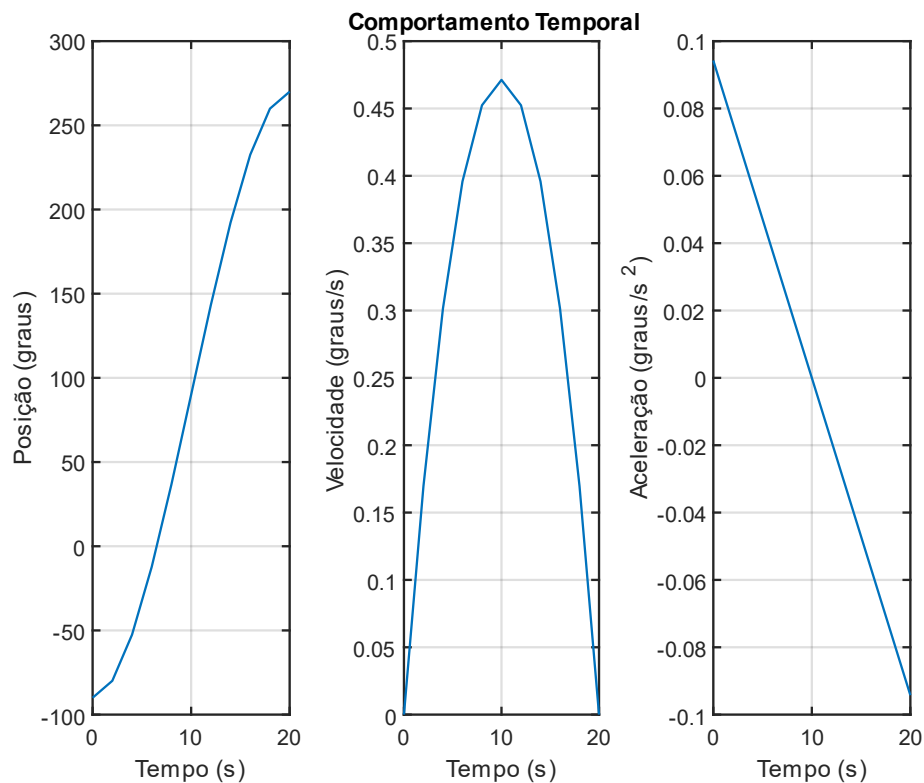


Fonte: Autoria própria (2022)

Finalmente, ao analisar a resposta da última junta presente na Figura 31, em comparação a resposta na Figura 26, verifica-se os mesmos aspectos relatados nas análises anteriores. Há variação nos valores de velocidade e aceleração, sendo menores na Figura 31. Para os valores de velocidade há uma diminuição de

aproximadamente 0,43 graus/s e para a aceleração uma redução de aproximadamente 0,228 graus/s² - para a resposta presente na Figura 31, em comparação a exposta na Figura 26 - nos valores de pico atingidos. Pode-se então concluir, que assim como nas outras juntas, houve a diminuição nos valores de velocidade e aceleração devido à alteração do valor de entrada da variável tempo.

Figura 31 - Comportamento temporal (quarta junta)



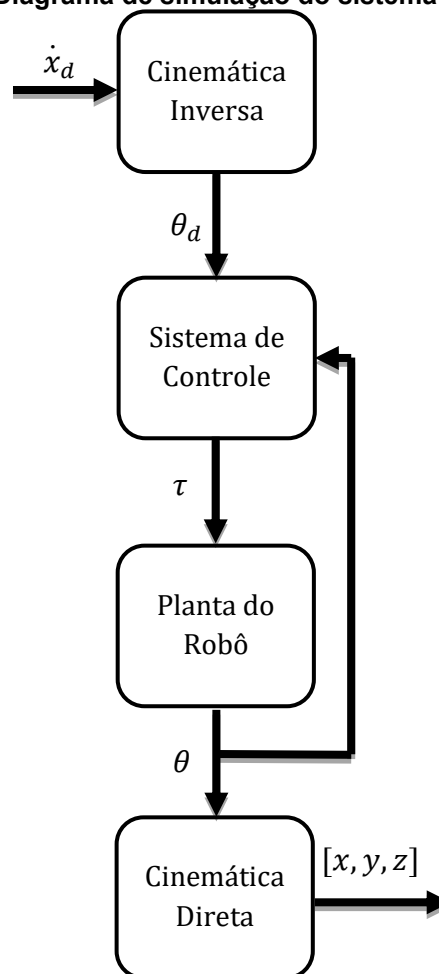
Fonte: Autoria própria (2022)

Ao final da comparação entre os resultados das simulações para a geração de trajetória, constata-se que estão de acordo com o esperado. Contudo, vale ressaltar que esses resultados não levam em conta as características dinâmicas do manipulador, características que são abordadas na seção subsequente.

4.5 Simulação do controle

Para a simulação do sistema de controle, usou-se do diagrama exposto na Figura 32. O diagrama em questão traz as adições dos blocos cinemática inversa à entrada e cinemática direta na saída dos valores para posição dos ângulos ao diagrama de controle descrito na Figura 8.

Figura 32 - Diagrama de simulação do sistema de controle



Fonte: Adaptado de Craig (2009)

Assim como a utilização dos valores presentes na Tabela 1, que contém os dados necessários para a realização dos cálculos das matrizes $M(\theta)$, $V(\theta, \dot{\theta})$ e $G(\theta)$, presentes no Apêndice – C. A partir da utilização da Equação (2), foram realizadas as simulações para a verificação dos resultados gerados pelo aplicado na etapa de controle.

Tabela 1 – Valores utilizados para a simulação do controle referentes ao robô SCARA SR-6 iA

Elos	Comprimento do Elos (m)	Comprimento em x (m)	Comprimento em y (m)	Comprimento em z (m)	Massa (kg)
E1	0,16	-	-	-	14
E2	0,35	0,35	0,18	0,05	6
E3	0,3	0,3	0,18	0,175	9
E4	0,21	0,01	0,01	0,21	1

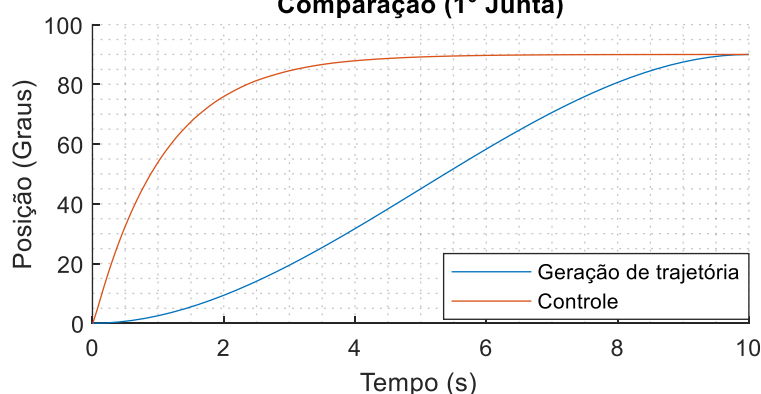
Fonte: Adaptado de FANUC (2017)

Na Figura 33, é apresentado o resultado da simulação para o $P_d = [-300, 350, 20]^T$, ponto desejado em milímetros, selecionada a opção de visualização “Junta 1”, $K_p = 9$, $K_v = 10$, $Tempo = 10 s$ e $Passo = 0,001 s$. Os

valores de K_p e K_v foram escolhidos pelo método de tentativa e erro, conforme a Equação (38) - a fim de se obter pólos localizados no eixo real e à esquerda do eixo imaginário $j\omega$ do plano complexo s - onde os valores K_p e K_v escolhidos representam os pólos reais ($x' = -1$ e $x'' = -9$), logo, é esperado que as respostas obtidas não apresentem níveis de *overshoot*.

O objetivo central dessa simulação é tornar possível a comparação das respostas gráficas de posição geradas pelo algoritmo de geração de trajetória, que leva em consideração apenas dados estáticos para os ensaios de geração de trajetória, e os resultados das posições obtidos através do sistema de controle implementado, que leva em consideração as características dinâmicas para o manipulador.

**Figura 33 - Simulação do controle para a primeira junta rotativa com $K_p = 9$ e $K_v = 10$
Comparação (1° Junta)**

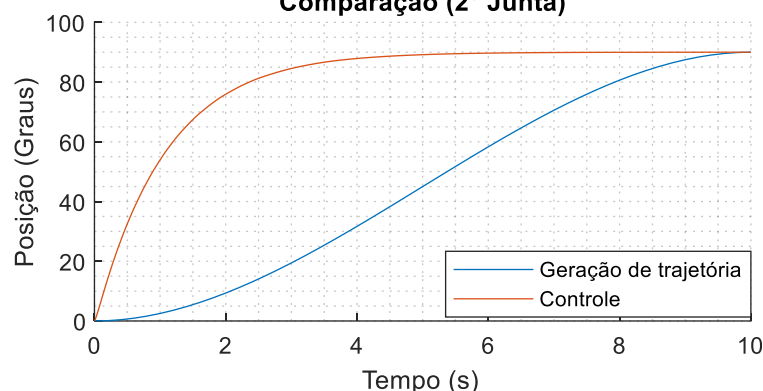


Fonte: Autoria própria (2022)

Ao analisar a Figura 33, identifica-se formatos diferentes para a curva que representa os valores de posição de θ_1 em relação ao tempo. Essa diferença se explica pela consideração das características dinâmicas do manipulador e a influência dos ganhos K_p e K_v , que possuem influência direta na trajetória que a junta 1 percorre ao longo do tempo.

Na Figura 34, verifica-se a mesma resposta apresentada para a Figura 33. Isso devido ao mesmo valor de movimentação da segunda junta em relação à posição inicial definida para o robô, exposta na Figura 11. Logo, os mesmos aspectos observados na Figuras 33, são observados para a Figura 34.

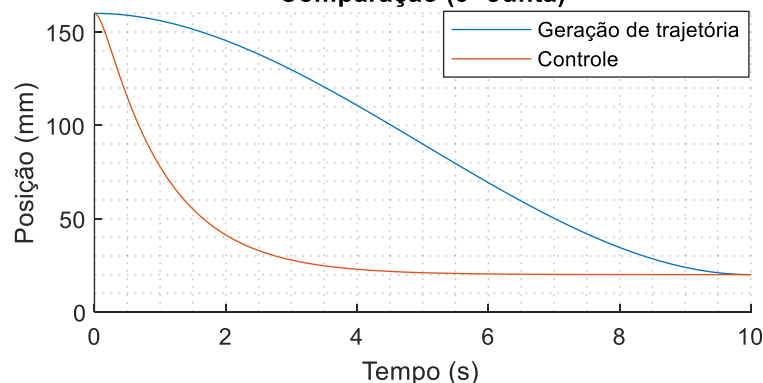
**Figura 34 - Simulação do controle para a segunda junta rotativa com $K_p = 9$ e $K_v = 10$
Comparação (2° Junta)**



Fonte: Autoria própria (2022)

A resposta para os valores atribuídos à trajetória da junta prismática está exposta na Figura 35. Ao fazer a comparação entre a curva em azul (Geração de trajetória) a curva em vermelho (Controle), nota-se que o curva em vermelho apresenta comportamento mais abrupto, assim como as respostas obtidas nas Figuras 33 e 34 para o sistema de controle, devido a inserção das características dinâmicas presentes no sistema de controle e a influência dos ganhos K_p e K_v .

**Figura 35 - Simulação do controle para a junta prismática com $K_p = 9$ e $K_v = 10$
Comparação (3° Junta)**



Fonte: Autoria própria (2022)

Realizadas às simulações de controle, conclui-se que o modelo utilizado para a Seção 4.2 - onde se tem a implementação por meio de polinômio de terceiro grau para a geração de trajetória - realiza um comportamento ideal, onde sempre será calculada uma trajetória suave para o deslocamento da junta, sendo o tempo de simulação a principal variável responsável por alterar essa trajetória.

Para a geração de trajetória por meio da implementação do sistema de controle particionado, a trajetória sofre influência das forças que atuam sobre manipulador e ainda dos valores de K_p e K_v . Onde os valores de K_p e K_v influenciam

nas posições dos pólos do sistema. Ao apresentar pólos complexos conjugados, têm-se respostas que apresentam *overshoot*, se os resultados das raízes forem positivas, o sistema se torna instável.

5 CONCLUSÃO

Realizou-se a implementação das equações desenvolvidas no decorrer do trabalho no *software* de simulação MATLAB® R2021a, deste modo, foi possível desenvolver um aplicativo capaz de realizar os cálculos e representações gráficas de todas as etapas de desenvolvimento do projeto (cinemática, dinâmica e controle), para qualquer robô do tipo SCARA que tenha estrutura similar ao do robô SCARA SR-6 iA.

A escolha do robô tipo SCARA para o TCC, ocorreu devido a sua grande aplicabilidade e versatilidade. Iniciou-se com a modelagem matemática da cinemática e da dinâmica do referido manipulador. Após estas etapas, usou-se a estratégia de controle particionado. Projetou-se um aplicativo para realização das parametrizações e simulações do robô. Por fim, analisaram-se os resultados gerados pelo aplicativo e compararam-se as respostas, estáticas e dinâmicas, do referido mecanismo.

Na cinemática direta, tem-se como entradas os valores dos ângulos de cada juntas (em graus) e deslocamento da junta prismática (em mm). A aplicação apresenta a posição do efetuador, tal como, a representação gráfica de todos os elos do manipulador no espaço cartesiano de acordo com os cálculos realizados através da aplicação dos valores inseridos pelo usuário nas matrizes homogêneas de transformação.

Para a cinemática inversa foi implementada a solução fechada na qual o aplicativo necessita como entrada a posição final do efetuador e a partir desse ponto inserido o algoritmo calcula os ângulos e o deslocamento da junta prismática necessários para que tal posição seja alcançada pelo efetuador. As simulações desta etapa apresentaram resultados conforme o esperado.

A geração de trajetória incorpora em sua representação tanto a cinemática direta como a cinemática inversa. Isso ocorre, pois a geração de trajetória escolhida para ser implementada acontece no espaço de juntas, como as entradas são a posição inicial e final do robô no espaço cartesiano, essas informações são transferidas para o espaço de juntas onde são calculados o deslocamento, a velocidade e a aceleração de cada junta, do manipulador. Logo, até o momento tem-se a posição das juntas no espaço de juntas, sendo necessária a conversão novamente para o espaço cartesiano, a fim de obter a representação gráfica clara da

posição final do efetuador, por meio de uma representação 3D de toda a trajetória gerada pelo robô, sendo esse processo realizado com êxito pelo aplicativo.

As equações dinâmicas no robô foram implementadas junto ao sistema de controle. O sistema de controle possibilita uma análise mais realista do movimento do robô, ao comparar com o modelo cinemático, em que esse último é estático. Ao simular o sistema de controle, nota-se a relação direta com os ganhos do controlador e que os mesmos podem ser alterados conforme as especificações do projeto. A partir das simulações no aplicativo, pode-se analisar e projetar sistemas robóticos para a indústria local, tal como para a área de Engenharia Biomédica.

O trabalho consegue atingir os objetivos de desenvolver um aplicativo genérico no qual o usuário pode realizar a simulação de robôs do tipo SCARA de maneira rápida. A aplicação disponibiliza acesso às representações gráficas ao usuário, às quais são de extrema importância para tornar mais nítido o entendimento dos resultados, que em muitas vezes só possuem apenas variáveis algébricas e em alguns livros e trabalhos na área, valores numéricos, que exigem um determinado nível de abstração de quem realiza tais tentativas para aprender desenvolver projetos para a robótica industrial.

Desta forma, pode-se concluir que o aplicativo desenvolvido para *desktop*, pode ser uma ferramenta importante para melhorar a curva de aprendizagem de robótica, para alunos e para profissionais que estão afastados da esfera acadêmica a algum tempo, que desejam aprender ou aplicar conceitos robóticos em ensaios para análise de comportamentos cinemáticos e dinâmicos juntamente ao sistema de controle para robôs do tipo SCARA.

Ao pensar em aprimoramentos, para ser uma ferramenta de aprendizagem mais completa, em trabalhos futuros o aplicativo pode vir a receber o desenvolvimento de outros manipuladores além do SCARA, tais como: Manipulador cartesiano, manipulador articulado, manipulador esférico, entre outros (buscando ampliar o horizonte de análises para diferentes tipos de manipuladores), evolução da interface com o usuário e aplicação do aplicativo em servidores *web*.

REFERÊNCIAS

CRAIG, J. J. **Introduction to robotics: mechanics and control**, 3. ed. Pearson Education India, 2009.

FREITAS, O. E. V., *et al.* Como despertar o engenheiro no jovem através da robótica com software e hardware livre. **Anais do salão internacional de ensino, pesquisa e extensão**, v. 13, n. 2, 2021.

DIAS, B. **Ranking dos 14 maiores fabricantes mundiais de robôs industriais - ServNews Robótica**. 2017. Disponível em: <https://servnews.com.br/site/ranking-dos-14-maiores-fabricantes-mundiais-de-robos-industriais/>. Acesso em: 1 abr. 2022.

EPSON. **Robôs SCARA all-in-one Epson synthis T3**, Epson Brasil. [S.l.]. Disponível em: <https://epson.com.br/Para-empresas/Rob%C3%B3tica/Rob%C3%B4s-SCARA/Rob%C3%B4s-SCARA-All-In-One-Epson-Synthis-T3/p/RT3-401SS>. Acesso em: 5 out. 2021.

FANUC. **Ultra high-speed SCARA robot: robot SR - 3iA/SR - 6iA**. FANUC América Corporation. 2017. Disponível em: https://aidex.com/wp-content/uploads/2018/01/FANUC-SR-Series-SCARA-Robot-Product-Info-Sheet_262.pdf. Acesso em: 29 abr. 2022.

FENERICK, J. A.; VOLANTE, C. R. A evolução das indústrias, os benefícios da automação e as perspectivas do mercado da robótica no brasil e no mundo. **Revista Interface Tecnológica**, v. 17, n. 1, p. 734-745, 2020.

FESTO. **Manipulador eléctrico cartesiano de tres ejes (desplazamiento de carros)**. [S.l.], 2021. Disponível em: <https://www.festo-didactic.com/es-es/productos/manipulador-electrico-cartesiano-de-tres-ejes-desplazamiento-de-carros.htm?fbid=ZXMuZXMuNTQ3LjE0LjE4Ljc2OS40NTMw>. Acesso em: 3 out. 2021.

GAMERO, I. **Robôs industriais: tudo o que você precisa saber!** 28 mar 2018. Disponível em: <https://pollux.com.br/blog/robos-industriais-tudo-o-que-voce-precisa-saber/>. Acesso em: 27 mar. 2022.

IFR. **Executive summary world robotics 2021 industrial robots**. 2021a. Disponível em: https://ifr.org/img/worldrobotics/Executive_Summary_WR_Industrial_Robots_2021.pdf. Acesso em: 2 abr. 2022.

IFR. **IFR presents world robotics 2021 reports**. 2021b. Disponível em: <https://ifr.org/ifr-press-releases/news/robot-sales-rise-again>. Acesso em: 21 mar. 2022.

IFR. **Robôs industriais**. 2020. Disponível em: <https://ifr.org/industrial-robots>. Acesso em: 1 abr. 2022.

LEAL, L. **R.U.R. e a origem do termo “robô”**. Editora Madrepérola. 23 jun. 2020. Disponível em: <https://www.editoramadreperola.com/r-u-r-e-a-origem-do-termo-roboto/>. Acesso em: 27 mar. 2022.

LYNCH, K. M.; PARK, F. C. **Modern robotics**. Cambridge *University Press*, 2017.]

MARTHA, L. F.; LOPES, P. C.; RANGEL, R. L. **O ambiente App Designer do MATLAB**. 2021. Disponível em: https://web.tecgraf.puc-rio.br/ftp_pub/lfm/CIV2801-AmbienteAppDesigner.pdf. Acesso em: 4 jun. 2022.

MATHWORKS. **MATLAB App Designer**. 2022. Disponível em: <https://www.mathworks.com/products/matlab/app-designer.html>. Acesso em: 4 jun. 2022.

MULT COMERCIAL. **Tipos de robô: conheça e entenda sobre robótica industrial**. 20 nov. 2020. Disponível em: <https://blog.multcomercial.com.br/tipos-de-roboto-conheca-e-entenda-sobre-robotica-industrial/>. Acesso em: 29 mar. 2022.

MURRAY, R. M.; LI, Z.; SASTRY, S. S. **A mathematical introduction to robotic manipulation**. CRC press, 2017.

NIKU, S. B. **Introduction to robotics: analysis, control, applications**. 3. ed. Hoboken: John Wiley & Sons, 2020.

OLIVEIRA, L. F. P. **Modelação, simulação e implementação de padrões de locomoção para robôs hexápodes**. 2016. Dissertação de mestrado.

OLIVEIRA, L. F. P.; ROSSINI, F. L. Modeling, simulation and analysis of locomotion patterns for hexapod robots. **IEEE Latin America Transactions**, v. 16, n. 2, p. 375-383, 2018.

OLIVEIRA, L. F. P. *et al.* Modeling, Simulation and Implementation of Locomotion Patterns for Hexapod Robots. *In: 2020 IEEE Congresso Biental da Argentina (ARGENCON)*. IEEE, 2020. p. 1-1.

PAZOS, F. **Automação de sistemas & robótica**. Rio de Janeiro: Axcel Books, 2002.

QUINTINO, L. F. *et al.* **Automação robótica em indústrias—evolução tecnológica e competitividade**. *In: XXII Congresso nacional de estudantes de engenharia mecânica*. 2015, Campo dos Goytacazes. Disponível em: https://www.researchgate.net/profile/Cesar-Augusto-Della-Piazza/publication/305680422_AUTOMACAO_ROBOTICA_EM_INDUSTRIAS_-_EVOLUCAO_TECNOLOGICA_E_COMPETITIVIDADE/links/5798d50508aeb0ffcd08bc2e/AUTOMACAO-ROBOTICA-EM-INDUSTRIAS-EVOLUCAO-TECNOLOGICA-E-COMPETITIVIDADE.pdf. Acesso em: 01 abr. 2022.

ROMANO, V.; DUTRA, M. **Introdução a robótica industrial**. Robótica Industrial: Aplicação na Indústria de Manufatura e de Processo, São Paulo: Edgard Blücher, 2002.

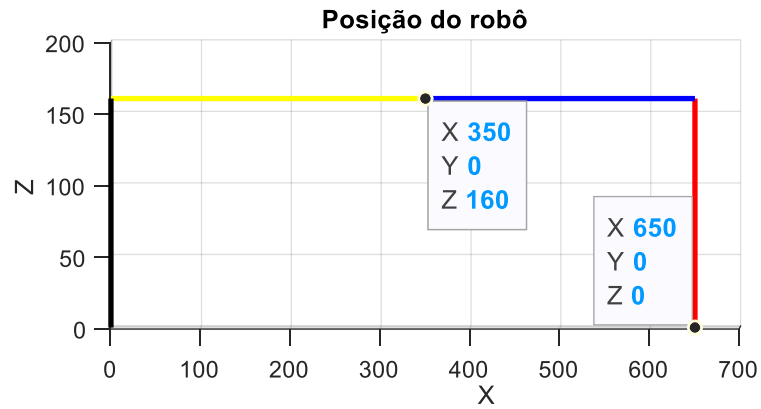
ROSÁRIO, J. M. **Robótica industrial modelagem, utilização e programação**. Editora Baraúna, 2010.

SANTOS, V. M. F. **Robótica industrial**. Universidade de Aveiro - Departamento de Engenharia Mecânica, [S.l], 2004.

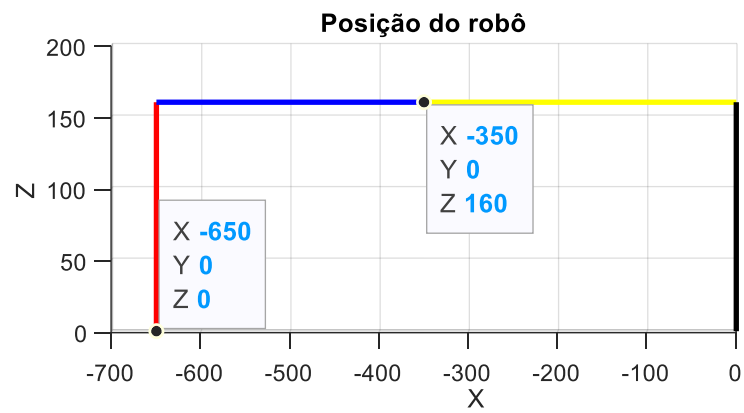
SPONG, M. W. *et al.* **Robot modeling and control**. New York: Wiley, 2006.

APÊNDICE A – Vista lateral do robô

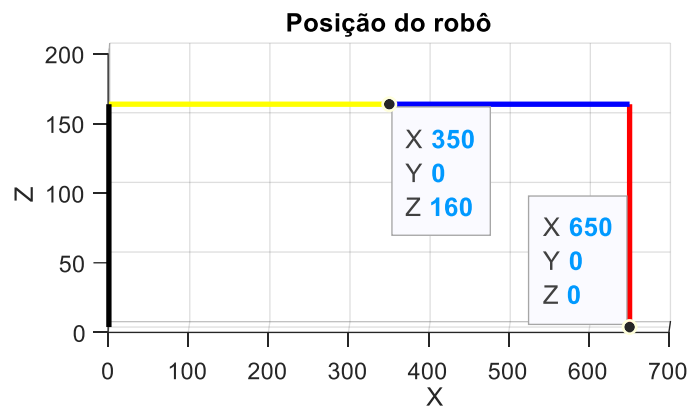
1)



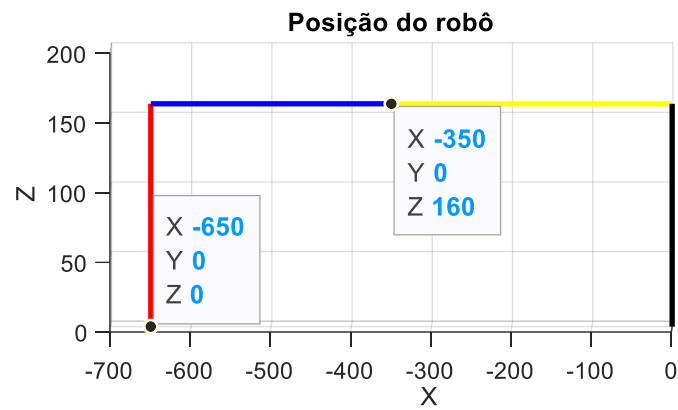
2)



3)



4)



APÊNDICE B – Equação cinemática direta

Como solução para a cinemática direta, tem-se a matriz:

$${}^0T_4 = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix}$$

sendo os elementos da matriz 0T_4 :

$$T_{11} = \cos \theta_1 (\cos \theta_2 \cos \theta_4 - \sin \theta_2 \sin \theta_4) - \sin \theta_1 (\cos \theta_2 \sin \theta_4 + \cos \theta_4 \sin \theta_2)$$

$$T_{12} = -\cos \theta_1 (\cos \theta_2 \sin \theta_4 + \cos \theta_4 \sin \theta_2) - \sin \theta_1 (\cos \theta_2 \sin \theta_4 - \sin \theta_2 \sin \theta_4)$$

$$T_{13} = 0$$

$$T_{14} = \cos \theta_1 (\cos \theta_1 (d_1 + d_2 \cos \theta_2) - d_2 \sin \theta_1 \sin \theta_2)$$

$$T_{21} = \cos \theta_1 (\cos \theta_2 \sin \theta_4 + \cos \theta_4 \sin \theta_2) + \sin \theta_1 (\cos \theta_2 \sin \theta_4 - \sin \theta_1 \sin \theta_4)$$

$$T_{22} = \cos \theta_1 (\cos \theta_2 \cos \theta_4 - \sin \theta_2 \sin \theta_4) - \sin \theta_1 (\cos \theta_2 \sin \theta_4 + \cos \theta_4 \sin \theta_2)$$

$$T_{23} = 0$$

$$T_{24} = \sin \theta_1 (d_1 + d_2 \cos \theta_2) + d_2 \cos \theta_1 \sin \theta_2$$

$$T_{31} = \sin \theta_1 (d_1 + d_2 \cos \theta_2) + d_2 \cos \theta_1 \sin \theta_2$$

$$T_{32} = 0$$

$$T_{33} = -1$$

$$T_{34} = d_0 - d_3$$

$$T_{41} = 0$$

$$T_{42} = 0$$

$$T_{43} = 0$$

$$T_{44} = 1$$

sendo:

d_i = Comprimento do elo em análise;

θ_i = O ângulo do sistema de referência em análise.

APÊNDICE C – Equações dinâmicas

Como representação das inércias, tem-se a matriz M dada por:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

sendo os elementos da matriz M descritos por:

$$M_{11} = \frac{m_1 L_1^2}{4} + I_{cz_1} + I_{cz_2} + L_2 s_2 (L_2 m_2 s_2 + L_2 m_3 s_2) + c_2 \left(m_2 \frac{L_2}{2} + L_2 c_2 \right) + \dots$$

$$\dots + m_3 (L_3 + L_2 c_2) + \frac{L_3 m_3 (L_3 + L_2 c_2)}{2} + \frac{L_2 m_2 \frac{L_2}{2} + L_2 c_2}{2}$$

$$M_{12} = I_{cz_2} + \frac{L_2^2 m_2}{4} + \frac{L_3^2 m_3}{2} + L_2 c_2 \frac{L_2 m_2}{2} + L_3 m_3$$

$$M_{13} = 0$$

$$M_{21} = I_{cz_2} + \frac{(L_3 m_3 (L_3 + L_2 c_2))}{2} + \frac{L_2 m_2 \frac{L_2}{2} + L_2 c_2}{2}$$

$$M_{22} = \frac{m_2 L_2^2}{4} + \frac{m_3 L_3^2}{2} + I_{cz_2}$$

$$M_{23} = 0$$

$$M_{23} = 0$$

$$M_{31} = -g m_3$$

$$M_{32} = -g m_3$$

$$M_{33} = -m_3 (g - 1)$$

Sendo:

L = Comprimento do elo;

m = massa do elo;

I_{cz} = A inércia;

$c_1 = \cos \theta_1$;

$c_2 = \cos \theta_2$;

$s_1 = \sin \theta_1$;

$s_2 = \sin \theta_2$.

O vetor centrífugo de Coriolis é representado por:

$$V = \begin{bmatrix} V_{11} \\ V_{21} \\ V_{31} \end{bmatrix}$$

sendo os elementos do vetor V descritos por:

$$\begin{aligned} V_{11} = & I_{cz_2} \dot{d}_3 + L_2 (c_2 (m_3 (L_3 \dot{d}_3 - \frac{L_3 (\dot{\theta}_1 + \dot{\theta}_2)}{2} + L_2 \dot{\theta}_1^2 s_2) + m_2 (L_2 s_2 \dot{\theta}_1^2 + \frac{L_2 \dot{d}_3}{2})) - \dots \\ & \dots - s_2 (m_2 (\frac{L_2 (\dot{\theta}_1 + \dot{\theta}_2)^2}{2} + L_2 \dot{\theta}_1^2 c_2) + m_3 (\frac{3L_3 \dot{\theta}_1 + \dot{\theta}_2^2}{2} + L_2 \dot{\theta}_1^2 c_2)) + \dots \\ & \dots + I_{cz_3} (\dot{\theta}_1 + \dot{\theta}_2) + \frac{(L_3 m_3 (L_3 \dot{d}_3 - \frac{L_3 (\dot{\theta}_1 + \dot{\theta}_2)}{2} + L_2 \dot{\theta}_1^2 s_2) + L_2 m_2 (L_2 s_2 \dot{\theta}_1^2 + \frac{L_2 \dot{d}_3}{2}))}{2} \end{aligned}$$

$$\begin{aligned} V_{21} = & I_{cz_2} \dot{d}_3 + I_{cz_3} (\dot{\theta}_1 + \dot{\theta}_2) + (L_3 m_3 (L_3 \dot{d}_3 - \frac{L_3 (\dot{\theta}_1 + \dot{\theta}_1)}{2}) + \frac{\dot{\theta}_1^2 + \frac{L_2 \dot{d}_3}{2}}{2} + \dots \\ & \dots + (L_2^2 m_2 s_2 \frac{\dot{\theta}_1^2 + \frac{L_2 \dot{d}_3}{2}}{2} \end{aligned}$$

$$V_{31} = 0$$

sendo:

L = Comprimento do elo;

m = massa do elo;

I_{cz} = A inércia do elo analisado;

$c_1 = \cos \theta_1$;

$c_2 = \cos \theta_2$;

$s_1 = \sin \theta_1$;

$s_2 = \sin \theta_2$.

Por fim o vetor que representa os termos da gravidade, expresso por:

$$G = \begin{bmatrix} 0 \\ 0 \\ -gm_3 \end{bmatrix}$$

sendo:

g = Gravidade;

m = Massa do elo.