



UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Dianny Pontes Machado

DESENVOLVIMENTO DE UM PROJETO DE CONTROLE DE UM ROBÔ PLANAR COM DOIS GRAUS DE LIBERDADE

Manaus
2015

Dianny Pontes Machado

**DESENVOLVIMENTO DE UM PROJETO
DE CONTROLE DE UM ROBÔ PLANAR
COM DOIS GRAUS DE LIBERDADE**

Trabalho de Conclusão de Curso submetido à Coordenação do curso de Engenharia de Controle e Automação da Universidade do Estado do Amazonas como parte dos requisitos necessários para a obtenção do grau de Engenheira.

Orientador Dr. Israel Mazaira Morales

Manaus
2015

Dianny Pontes Machado

**DESENVOLVIMENTO DE UM PROJETO DE
CONTROLE DE UM ROBÔ PLANAR COM DOIS
GRAUS DE LIBERDADE**

Trabalho de Conclusão de Curso submetido à Coordenação do curso de Engenharia de Controle e Automação da Universidade do Estado do Amazonas como parte dos requisitos necessários para a obtenção do grau de Engenheira.

Aprovada em 25 de junho de 2015.

BANCA EXAMINADORA

Dr. Israel Mazaira Morales
Orientador

Dr. Walter A. Vermehren Valenzuela
Presidente da banca

Dr. Miguel Angel Rodriguez Borroto
Convidado 2

Dr. Daniel Guzman Del Rio
Convidado 3

Manaus
2015

AGRADECIMENTOS

Mais uma etapa da minha vida concluída. A caminhada não foi fácil, porém só tenho a agradecer a Deus por tudo que me proporciona, aos meus pais, Juscelino e Lilian que sempre acreditaram e apoiaram para que eu chegasse até aqui, ao meu irmão Alex Diego, pelo cuidado e preocupação com sua irmã caçula e a toda minha família que sempre estão na torcida pelo meu sucesso.

Agradeço ao meu Anjo, Antonio Rafael, meu presente de Deus, o meu maior incentivo para continuar e superar cada desafio encontrado, a Ele dedico esta formação, pois sei que esse era seu grande desejo.

Agradeço também à Universidade do Estado do Amazonas, à coordenação de Engenharia de Controle e Automação e todos os seus professores, nos quais tenho a honra de tê-los como tutores. Agradeço especialmente ao meu orientador, Dr. Israel Mazaira Morales.

Agradeço a todos os meus amigos, sem exceção, que me deram forças para vencer esta batalha, onde falo com propriedade, meus amigos fizeram a diferença. Aqui não fiz amigos, mas sim irmãos os quais quero levar por toda vida!

RESUMO

Os softwares dos robôs disponíveis na Universidade do estado do Amazonas - EST-UEA não são livres, dificultando o estudo e a adequação a diferentes aplicações. A elaboração deste projeto, visa o desenvolvimento um projeto no programa MATLAB®, para controlar o posicionamento de um Robô Planar, utilizando o ambiente de simulação Simulink®, tendo como um dos objetivos principais a comunicação entre o computador e o sistema. Este possibilitará ao corpo discente a fácil modificação dos parâmetros de controle do robô, diferentes estratégias de controle e utilização de outras técnicas de controle contribuindo com o desenvolvimento acadêmico e didático da prática da Robótica. Com a intenção de aprimorar o conhecimento nas disciplinas Robótica Industrial, Controle e Automação, Eletrônica Analógica e Linguagem de Programação.

Palavras-chaves: Robô Planar. Modelagem. Controle. Posição.

ABSTRACT

The software robots available in the Amazonas State University - EST-UEA are not free, making it difficult to study and the adaptation to different applications. The development of this project is to develop a project in MATLAB® program to control the positioning of a robot Planar using the Simulink® simulation environment, having as a main objective communication between the computer and the system. This will enable the student body easy modification of robot control parameters, different control strategies and use of other control techniques contributing to the academic and educational development of the Robotics practice. In order to improve knowledge in the subjects Industrial Robotics, Automation and Control, Analog Electronics and Programming Language.

Key-words: Planar robot. Modeling. Control. Position

LISTA DE ILUSTRAÇÕES

Figura 1 – Braços com um (à esquerda) e dois graus de liberdade (à direita) . . .	16
Figura 2 – Os seis tipos possíveis de juntas do par inferior	17
Figura 3 – Relação fixa entre os dois eixos de juntas.	18
Figura 4 – Notação de Denavit-Hartenberg-Organização dos referenciais associados a um sistema articulado.	20
Figura 5 – Configuração do elemento terminal de um robô	24
Figura 6 – Utilização do modelo geométrico direto e da matriz Jacobiana inversa para determinar uma configuração ϑ^* correspondente a uma situação desejada x^*	25
Figura 7 – Pêndulo simples (1GL)	27
Figura 8 – Pêndulo duplo (2 GL)	27
Figura 9 – Controle de Posição de um robô no seu espaço operacional	28
Figura 10 – Estrutura de Controle de Robôs Manipuladores	31
Figura 11 – Principais partes de um Servomotor. Carcaça do Motor (1), Bobinagem do estator (2), Chapas do Estator (3), Rotor com ímãs (4), Resolver (5) e Freio (6)	32
Figura 12 – Servomotor AC Assíncrono	33
Figura 13 – Componentes do Robô Planar	34
Figura 14 – Projeto Proposto	35
Figura 15 – Projeto das juntas	36
Figura 16 – Conexão das juntas	36
Figura 17 – Robô Planar de 2 eixos	37
Figura 18 – Plano Cartesiano de um Robô Planar com dois graus de liberdade . . .	41
Figura 19 – Rótulo padrão do MR-TB20	46
Figura 20 – MR-TB20 Utilizado no projeto	46
Figura 21 – Servomotor HC-RFS503S	47
Figura 22 – Servoamplificador MR-J2S-500A	48
Figura 23 – Esquemático do cabo de comunicação RS-232	49
Figura 24 – Envio de trem de pulsos no modo posição	50
Figura 25 – Conversor tensão-frequência UD-C450	51
Figura 26 – Circuito Integrado	52
Figura 28 – CI CD4049UBC Hex Inverting Buffer	53
Figura 27 – Entrada diferencial - Modo Posição do MR-J2S-500A	53
Figura 29 – Interface de Comunicação	54
Figura 30 – Microcontrolador Arduino Mega 2560	55
Figura 31 – Algoritmo teste de controle do elo do Robô	57

Figura 32 – Ambiente de controle pelo usuário	58
Figura 33 – Janela de resultados	58
Figura 34 – Janela Trajetória	59
Figura 35 – Janela Ajuda	59
Figura 36 – Algoritmo Cinemática Inversa	60
Figura 37 – Erro da junta θ_1	61
Figura 38 – Erro da junta θ_2	62
Figura 39 – Estabilidade dos elos	62
Figura 40 – Torque θ_1	63
Figura 41 – Torque θ_2	64
Figura 42 – Posição inicial	64
Figura 43 – Trajetória	65
Figura 44 – Posição desejada	65
Figura 45 – Trajetória do movimento	66
Figura 46 – Algoritmo completo simulação de um manipulador com dois graus de liberdade	67

LISTA DE TABELAS

Tabela 1 – Parâmetros nominais do elo	36
Tabela 2 – Tabela de Parâmetros D-H	37
Tabela 3 – Principais especificações do Servoamplificador	48
Tabela 4 – Comandos de programação do MR-Configurator	50
Tabela 5 – Características principais do Arduino Mega 2560	55

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Formulação do Problema	11
1.2	Motivação	11
1.3	Justificativa	12
1.4	Objetivos	12
1.5	Metodologia	12
1.6	Organização do Trabalho	13
2	TRABALHOS RELACIONADOS	14
2.1	Desenvolvimento de um robô manipulador industrial	14
2.2	Simulação do modelo cinemático inverso de um robô através do uso de redes neurais artificiais: um complemento ao ensino de robótica	14
2.3	Robôs cartesianos	15
3	REFERENCIAL TEÓRICO	16
3.1	Cinemática dos Manipuladores	16
3.1.1	Descrição de elos dos manipuladores	17
3.1.2	Método de Denavit–Hartenberg	18
3.1.3	Algoritmo para obtenção de coordenadas para o elo	21
3.1.4	Obtenção da Matriz de Transformação Homogênea A_i^{i-1}	22
3.1.5	Matriz Transformada T	23
3.1.6	Cinemática Inversa	24
3.1.7	Descrição Matemática de um robô com N graus de liberdade	25
3.1.8	Matriz Jacobiana	26
3.1.9	Inversão da Matriz Jacobiana: Controle de Posição de um Robô	28
3.2	Dinâmica dos Manipuladores	29
3.2.1	Formulação Lagrangiana da Dinâmica dos Manipuladores	29
3.3	Controle dos Manipuladores	31
3.4	Servomotor	31
3.5	Microcontrolador	33
4	MATERIAIS E MÉTODOS	34
4.1	Projeto do Robô Planar	34
4.2	Modelagem	37
4.3	Modelagem Cinemática	37

4.4	Modelagem Dinâmica	40
4.5	Controle do Robô	42
4.6	Hardware	45
4.6.1	Ligação Elétrica	45
4.6.2	Funcionamento do sistema servomotor e servoamplificador	47
4.6.3	Testes de funcionamento do sistema de servo	49
4.7	Interface de comunicação	54
4.7.1	Microcontrolador Arduino Mega 2560	54
4.7.2	Interface Gráfica	57
5	RESULTADOS OBTIDOS	60
5.1	Algoritmo teste de controle do elo do Robô	60
5.2	Simulação de controle de posição do robô planar	61
5.2.1	Animação do movimento	64
5.3	Algoritmo completo simulação de um manipulador com dois graus de liberdade	66
6	CONCLUSÕES	68
6.1	Trabalhos Futuros	68
	Referências	70
	ANEXO A – ALGORITMO DE CONTROLE IMPLEMENTADO NO MATLAB®	72

1 INTRODUÇÃO

Pelo conceito de Bryan (2012), ao longo das últimas décadas a abertura de mercado e o processo de globalização da economia têm levado o setor industrial a um novo padrão de concorrência, onde as formas tradicionais de gestão e produção não são mais suficientes para garantir a lucratividade e sobrevivência no mercado que se apresenta cada vez mais competitivo e exigente em qualidade, custo e atendimento. Visando o aumento de produtividade e padronização na qualidade de produtos, muitas indústrias têm implementado a automatização de sua produção baseada na utilização de robôs manipuladores para realização de tarefas pré-determinadas e repetitivas.

O curso de Engenharia de Controle e Automação na Escola Superior de Tecnologia da Universidade do Estado do Amazonas (EST-UEA), possui um extenso leque de disciplinas que envolvem as áreas de mecânica, eletrônica e computação, onde por meio destas pode ser feito o controle e/ou a automatização de um sistema, porém na EST-UEA, esses conteúdos são ministrados de forma teórica, e dentre estes está a Robótica Industrial. Devido à inexistência do estudo prático da robótica, o presente projeto visa à melhoria para o desempenho acadêmico dos alunos desta instituição, por meio do desenvolvimento de um projeto de controle de posição que irá controlar o posicionamento de um robô planar, tendo como objetivo fazer a interface entre o computador e o Robô.

1.1 Formulação do Problema

Inexistência do estudo de prática da Robótica na área de controle de posicionamento de um robô planar, no laboratório da Escola Superior de Tecnologia da Universidade do Estado do Amazonas (EST-UEA), por falta de Interface Humano-Máquina (IHM).

É possível desenvolver um projeto para o controle de posição do robô planar, através de um servomotor e um servoamplificador, disponíveis no laboratório da EST-UEA, que possibilite a interface entre o computador e o Robô Planar.

1.2 Motivação

Devido a dificuldade encontrada durante a graduação, pela falta de estudo prático da robótica industrial, os alunos da universidade tem como alternativa utilizar ferramentas computacionais de simulação. Com isso, é difícil modelar um sistema que seja igual ao sistema real. Então, foi proposto o desenvolvimento do projeto de controle do robô para melhor conhecimento prático da robótica industrial.

1.3 Justificativa

Os softwares dos robôs disponíveis na EST-UEA não são livres, dificultando o estudo e a adequação a diferentes aplicações. O desenvolvimento do projeto proposto possibilitará ao corpo discente a fácil modificação dos parâmetros de controle do robô, diferentes estratégias de controle e utilização de outras técnicas de controle contribuindo com o desenvolvimento acadêmico e didático da prática da Robótica. Com a intensão de aprimorar o conhecimento nas disciplinas Robótica Industrial, Controle e Automação, Eletrônica Analógica e Linguagem de Programação.

1.4 Objetivos

Desenvolver um projeto de controle de posição de um robô planar com dois graus de liberdade, na ferramenta computacional MATLAB® e por meio do *software* Simulink®, realizar a simulação da trajetória dos elos do robô, a fim de obter os parâmetros necessários para aplicação futura. Este objetivo geral será alcançado quando todos seguintes objetivos específicos forem atingidos:

- Realizar modelagem cinemática e dinâmica do robô simulado;
- Realizar controle do robô simulado;
- Avaliar o comportamento do modelo matemático obtido em simulação;
- Acoplamento do servomotor e servoamplificador da Mitsubishi;
- Análise das características do controle de posição, velocidade e torque;
- Determinar os componentes auxiliares para atuação dos elos do robô;
- Interface de comunicação entre o servomotor e computador;
- Desenvolvimento do algoritmo para controle das juntas.

1.5 Metodologia

Inicialmente, pesquisas científicas foram realizadas para identificar, trabalhos relacionados com o tema proposto, e em que pontos este projeto se assemelha e se diferencia dos trabalhos já existentes.

Para realização deste projeto foram realizadas pesquisas bibliográficas nas áreas de robótica industrial, para obtenção da modelagem cinemática e dinâmica do robô, controle de manipuladores, para processar os sinais de entrada e converter estes sinais na movimentação do robô, isto é realizado por meio de um sistema de *software* e *hardware*.

Após o estudo desses conceitos foram realizadas as primeiras tentativas para construção do modelo do Robô planar, com as ligações elétricas do primeiro servomotor acoplado ao servoamplificador, representando a primeira junta do robô. Três testes foram realizados até alcançar o método mais viável para o teste de funcionamento do sistema.

As ligações elétricas do Servomotor foram efetuadas realizado os testes nos três modos de funcionamento do equipamento: posição, velocidade e torque. Os teste foram realizados com o software recomendado pelo manual Mitisubishi (2003), e assim os parâmetros de posição, velocidade e torque foram alterados para validar o funcionamento do Motor. Por meio do gerador de função, também foi realizado o teste no modo posição sendo enviado um trem de pulso de onda quadrada, variando a frequência e assim obtendo uma variação proporcional na velocidade do eixo do Servomotor, conforme descrito no manual do fabricante. E por fim, por meio do Simulink® foi feito um programa para controle de posição utilizando o modo de posição do Servoamplificador, onde o usuário pode ajustar o Setpoint de posição angular e, o microcontrolador servirá de interface recebendo dados do encoder e enviando sinal de atuação até obter o resultado desejado.

1.6 Organização do Trabalho

Este trabalho está dividido em seis capítulos para seu melhor entendimento. O capítulo 1 aborda uma breve introdução sobre controle de manipuladores, e conceitua o projeto por meio da justificativa e motivação. Também traz objetivos a serem alcançados e a metodologia que foi utilizada.

Já no capítulo 2 se dá a apresentação de trabalhos relacionados a essa monografia, encaixando-a no contexto científico, descrevendo as principais semelhanças e diferenças com os trabalhos já publicados.

Um referencial teórico, direto e aplicado, necessário ao desenvolvimento do projeto, é realizado no capítulo 3. Isso ocorrerá apresentando os tópicos relevantes por meio da visão de autores de obras consultadas e de trabalhos de pesquisa voltados para teorias e tecnologias aplicadas nesta monografia.

Em seguida, o capítulo 4 descreve como foi realizado o processo do desenvolvimento da proposta de controle de posição do robô planar, citando os materiais utilizados. Todas as etapas da metodologia são detalhadas, descrevendo-se o que foi necessário controle do manipulador em questão.

No capítulo 5 os resultados obtidos são mostrados e discutidos.

Por fim, no capítulo 6, os resultados são comparados, e uma conclusão é formulada, de acordo com o tema proposto, com a intenção de validar a proposta de controle de posição do robô planar de dois graus de liberdade.

2 TRABALHOS RELACIONADOS

Neste capítulo serão abordados alguns trabalhos científicos que, de alguma maneira, se assemelham com o tema desta monografia. O objetivo é encaixá-la no contexto científico, descrevendo as semelhanças e diferenças para os trabalhos já publicados.

2.1 Desenvolvimento de um robô manipulador industrial

O trabalho relata sobre o desenvolvimento de um robô manipulador industrial, que tem a finalidade de aplicação em linhas de produção, para otimizar peças. Para resumir a pesquisa serão descritas, de acordo com autor do projeto vantagens e desvantagens de sistemas robóticos, dentre outros aspectos que envolve este tema de robô manipulador.

Dentre as vantagens do sistema robótico no desenvolvimento do projeto da planta de produção pode-se citar a simplificação do sistema mecânico, redução do tempo e custo do desenvolvimento, facilidade de introduzir modificações e novas “capacidades” e flexibilidade para receber futuras modificações ou novas funcionalidades. De acordo com Campbell et al. (2007), quanto ao desenvolvimento do produto as vantagens são a flexibilidade de operação (programabilidade), inteligência (capacidade para sensoriamento e processamento das informações, se adaptando a diferentes condições de operação), automonitoramento e prevenção de acidentes, autodiagnóstico em caso de falhas, redução do custo de manutenção e elevado grau de precisão e confiabilidade. Outras vantagens do sistema robótico seriam o aumento da produtividade, melhoria e consistência na qualidade final do produto, redução da demanda de mão-de-obra especializada, contabilidade do processo, facilidade de programação, operação em ambientes hostis e perigosos ao homem e a realização de tarefas desagradáveis e repetitivas. Poucas desvantagens do sistema robótico podem ser citadas como, por exemplo, o custo inicial alto, sendo este amortizado durante seu tempo de uso, e a limitação do robô em relação à capacidade de aprendizado, memória, decisão e movimentos em relação ao ser humano.

2.2 Simulação do modelo cinemático inverso de um robô através do uso de redes neurais artificiais: um complemento ao ensino de robótica

Este trabalho despertou o interesse em relacionar ao projeto desta monografia, pelo foco no modelo cinemático inverso dos robôs manipuladores, porém a pesquisa possui aplicação em outro ramo, o que mostra a abrangência da Robótica industrial.

Segundo Camargo, Veraszto e Barreto (2003), o ensino de robótica passa, necessariamente, pelo estudo dos modelos cinemáticos dos robôs manipuladores. Por sua vez, a cinemática de um robô manipulador pode ser descrita através de seus modelos direto e inverso. O modelo cinemático inverso, através do qual obtém-se o estado das juntas em função da posição desejada para a ferramenta do robô, normalmente é descrito e ensinado de forma algébrica nas aulas de robótica. Contudo, a representação algébrica deste modelo é, frequentemente, de difícil obtenção. Assim, embora seja inquestionável a necessidade de determinação exata do modelo cinemático inverso de um robô, o uso de redes neurais artificiais (RNAs) na fase de projeto pode ser muito atraente, pois nos permite prever o comportamento do robô antes do desenvolvimento formal de seu modelo. Neste sentido, este trabalho apresenta uma forma relativamente rápida de se simular o modelo cinemático inverso de um robô, permitindo, assim, que o aluno possa ter uma visão global do modelo, vindo a identificar pontos que devem ser corrigidos ou que podem ser otimizados na estrutura de um robô.

2.3 Robôs cartesianos

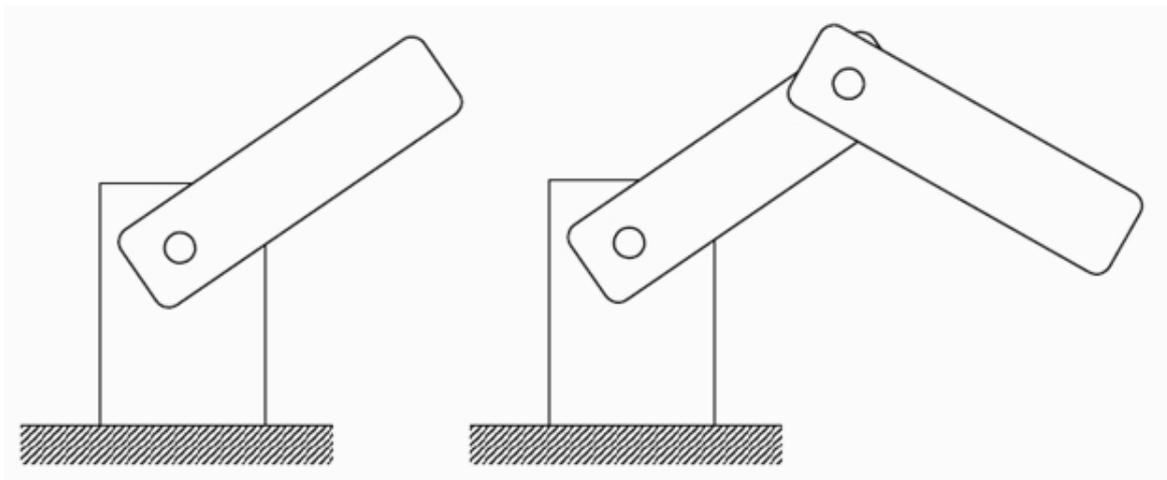
O trabalho em questão, relata sobre diversos tipos de robôs cartesianos, desde de sua estrutura de *hardware* até a comunicação com o computador, assim como a estrutura de controle de movimentos do robô industrial.

O robô industrial nasce da união de uma estrutura mecânica articulada e um sistema eletrônico de controle que se integra a um computador. Isto permite a programação e o controle dos movimentos. Cada servomotor é controlado por um servodrive que recebe o monitoramento de posição dos encoders acoplados a cada eixo. Desta forma, o sistema de controle sabe a localização exata dos eixos. O computador recebe estes dados de posição e, obedecendo ao programa do usuário, executa as operações de movimento de cada eixo, para atingir a posição programada. O programa é inserido pela unidade de programação, também chamada de *teach pendant*, embora também possa ser editado no teclado do computador. Isto varia de acordo com o modelo e o fabricante do robô.

3 REFERENCIAL TEÓRICO

Robô planar é um robô manipulador, onde sua área de atuação é em apenas um plano, sendo de 1 grau de liberdade-gdl, 2gdl ou mais, como na figura 1. Serão utilizados para realização desta pesquisa, um servomotor acoplado em um servoamplificador, que tem a função de driver do motor. Para isso será necessário o estudo da cinemática, dinâmica e controle de robôs.

Figura 1 – Braços com um (à esquerda) e dois graus de liberdade (à direita)



Fonte:<http://www.profelectro.info>

3.1 Cinemática dos Manipuladores

Para CRAIG (2012), Cinemática é a ciência do movimento que trata do assunto sem considerar as forças que o causam. É estudada a posição, a velocidade, a aceleração e todas as derivadas de ordem mais elevadas das variáveis de posição. Contudo, o estudo da cinemática dos manipuladores refere-se a todas as propriedades do movimento que sejam geométricas e baseadas no tempo. Com intuito de lidar com a geometria complexa de um manipulador, serão fixados sistemas de referência às várias partes do mecanismo e depois será descrito as relações entre eles. O estudo da cinemática dos manipuladores envolve, entre outras coisas, como a localização dos sistemas de referência se altera à medida que o mecanismo se articula.

A cinemática é um método para computar a posição e a orientação do efetuador do manipulador em relação à sua base como função das variáveis juntas.

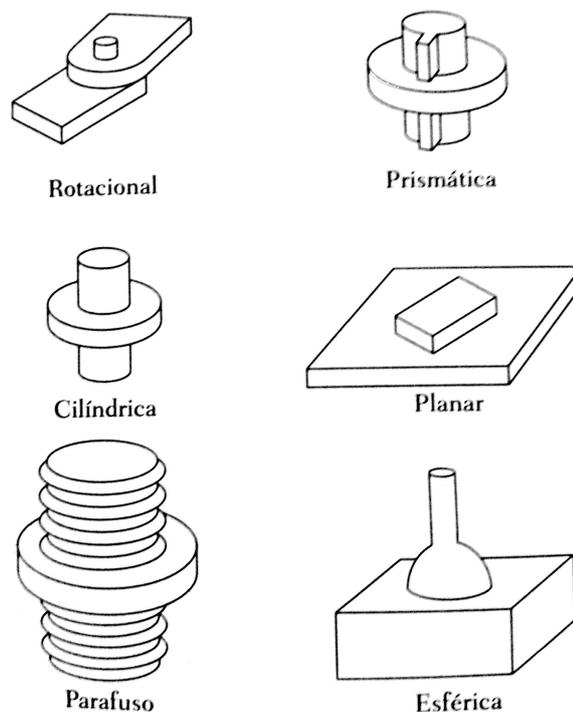
3.1.1 Descrição de elos dos manipuladores

CRAIG (2012) afirma que um manipulador pode ser considerado um conjunto de corpos conectado em cadeia, por juntas, chamados de elos. As juntas formam uma conexão entre um par de elos vizinhos. O termo par inferior é usado para descrever a conexão entre um par de corpos quando o movimento relativo caracteriza-se por duas superfícies que deslizam uma sobre a outra.

Características do projeto mecânico favorecem que os manipuladores sejam em geral construídos com base em juntas que tenham apenas um grau de liberdade. A maioria dos manipuladores tem juntas rotacionais ou possuem juntas deslizantes chamadas juntas prismáticas. (CRAIG, 2012)

A numeração dos elos é feita a partir da base fixa do braço, que pode ser chamada de elo 0. O primeiro corpo móvel é o elo 1, e assim por diante, até a extremidade livre, que é o elo n. Com intuito de posicionar um efetuador, em geral no espaço tridimensional, é necessário um mínimo de seis juntas. Um único elo de um robô típico tem muitos atributos que um projetista mecânico deve considerar durante projeto: o tipo de material usado, as resistências e a dureza do elo, a localização e o tipo dos mancais das juntas, a forma externa, o peso e a inércia e entre outros.

Figura 2 – Os seis tipos possíveis de juntas do par inferior

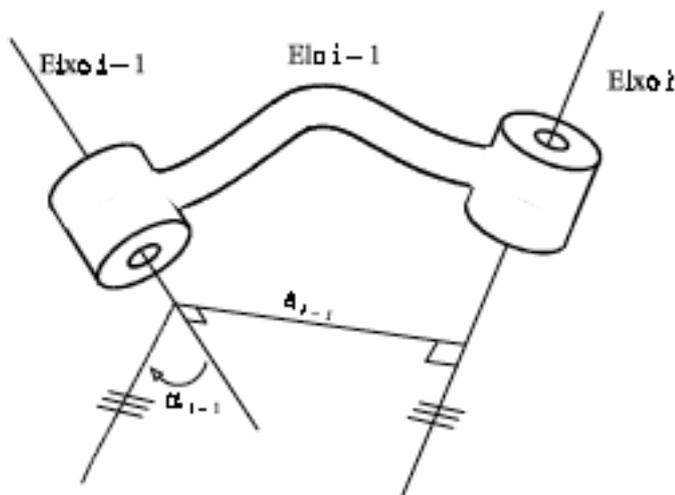


Entretanto, para o objetivo de se obterem as equações cinemáticas do mecanismo, um elo pode ser definido por dois números que definem a localização relativa dos dois eixos no espaço, considerado apenas um corpo rígido de um manipulador. Os eixos de juntas são definidos por linhas no espaço.

O eixo de junta i é definido por uma linha no espaço, ou um vetor direção, em torno do qual o elo i rotaciona em relação ao elo $i-1$. Acontece que, para os propósitos da cinemática, um elo pode ser especificado com dois números que definem a localização relativa dos dois eixos no espaço. (CRAIG, 2012)

De acordo com a apostila de Santos (2004) tem-se a figura 3, que mostra o elo $i-1$ e a linha mutuamente perpendicular ao longo da qual o comprimento de elo, α_{i-1} , é medido. Outra forma de visualizar o parâmetro de elo α_{i-1} é imaginar um cilindro em expansão, cujo eixo é o eixo da junta $i-1$ - quando ele apenas toca o eixo de junta i , o raio do cilindro é igual a α_{i-1} . O segundo parâmetro necessário para definir a localização relativa de dois eixos chama-se torção do elo. Um plano cuja normal é a linha perpendicular mútua, onde é projetado os eixos $i-1$ e i nesse plano e medido o ângulo entre eles. O ângulo é medido do eixo $i-1$ ao eixo i no sentido da mão direita, em torno de α_{i-1} . Ainda na figura 3, α_{i-1} é indicado como ângulo entre o eixo $i-1$ e o eixo i . Quando os eixos se interceptam, a torção é medida no plano que contém os dois eixos, mas o sentido de α_{i-1} é perdido. Neste caso especial o sinal de α_{i-1} é arbitrário.

Figura 3 – Relação fixa entre os dois eixos de juntas.



Fonte:(SANTOS, 2004)

3.1.2 Método de Denavit–Hartenberg

Um robô pode ser descrito, cinematicamente, através de apenas 4 parâmetros, a translação e rotação entre dois elos adjacentes, Denavit e Hartenberg propuseram um método matricial para estabelecimento sistemático de um sistema de coordenadas fixo para

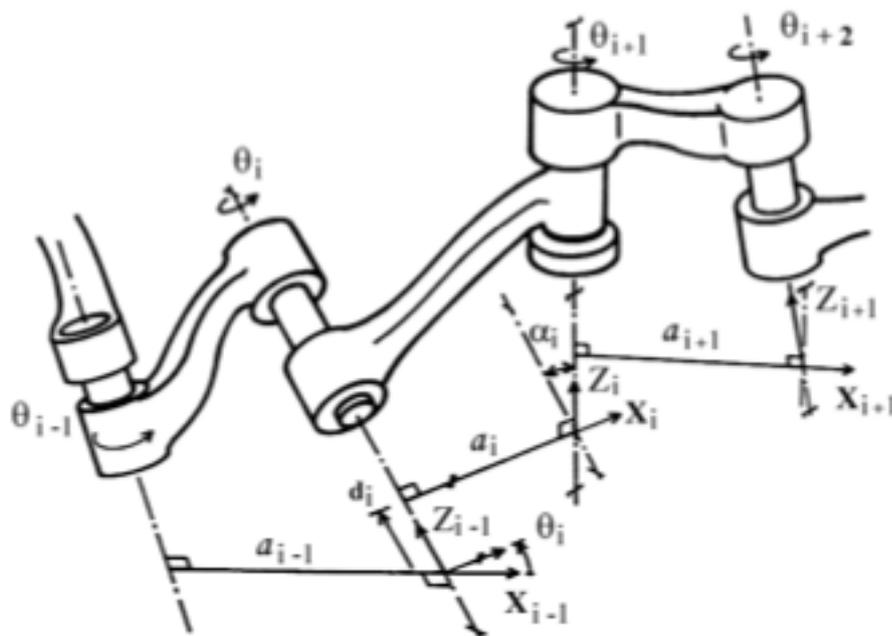
cada elo de uma cadeia cinemática articulada. Rossato (2009) afirma que representação de Denavit-Hartenberg (D-H) resulta na obtenção de uma matriz de transformação homogênea 4×4 , representando cada sistema de coordenadas do elo na junta, em relação ao sistema de coordenadas do elo anterior. Assim, a partir de transformações sucessivas, podem ser obtidas as coordenadas do elemento terminal de um robô (último elo), expressas matematicamente no sistema de coordenadas fixo a base. Assim sendo, um sistema de coordenadas cartesianas ortonormal (X_i, Y_i, Z_i) pode ser estabelecido para cada elo no seu eixo de junta, onde $i = 1, 2, \dots, N$ (N número de graus de liberdade) mais o sistema de coordenadas da base. Assim, uma junta rotacional tem somente 1 grau de liberdade, e cada sistema de coordenadas (X_i, Y_i, Z_i) do braço do robô corresponde a junta $i + 1$, sendo fixo no elo i . Quando o acionador ativa a junta i , o elo i deve mover-se com relação ao elo $i - 1$. Assim, o i -ésimo sistema de coordenadas é solidário ao elo i , se movimentando junto com o mesmo. Assim, o n -ésimo sistema de coordenadas se movimentará com o elemento terminal (elo n). As coordenadas da base são definidas como o sistema de coordenadas 0 (X_0, Y_0, Z_0) , também chamado de sistema de referência inercial. Os sistemas de coordenadas são determinados e estabelecidos obedecendo três regras:

1. O eixo Z_{i-1} é colocado ao longo do eixo de movimento da junta i .
2. O eixo X_i é normal ao eixo Z_{i-1} , e apontando para fora dele.
3. O eixo Y_i completa o sistema utilizando a regra da mão direita.

Através destas regras podemos observar que:

1. A escolha do sistema de coordenadas é livre, podendo ser colocada em qualquer parte da base de suporte, enquanto que a posição do eixo Z_0 , deverá ser a do eixo de movimento da primeira junta.
2. O último sistema de coordenadas (n -ésimo) pode ser colocado em qualquer parte do elemento terminal, enquanto que o eixo X_i é normal ao eixo Z_{i-1} .

Figura 4 – Notação de Denavit-Hartenberg-Organização dos referenciais associados a um sistema articulado.



Fonte:(ROSSATO, 2009)

A representação D-H de um elo rígido dependerá de quatro parâmetros associados ao elo. Estes parâmetros descrevem completamente o comportamento cinemático de uma junta prismática ou revoluta. Estes quatro parâmetros são definidos a seguir:

- θ_i é o ângulo de junta obtido entre os eixos X_{i-1} e X_i no eixo Z_{i-1} (usar a regra da mão direita).
- d_i é a distância entre a origem do $(i-1)$ -ésimo sistema de coordenadas até a interseção do eixo Z_{i-1} com o eixo X_i ao longo do eixo Z_{i-1} .
- a_i é a distância (off-set) entre a interseção do eixo Z_{i-1} com o eixo X_i até a origem do i -ésimo sistema de referência ao longo do eixo X_i (ou a menor distância entre os eixos Z_{i-1} e Z_i).
- α_i é o ângulo offset entre os eixos Z_{i-1} e Z_i medidos no eixo X_i (usando a regra da mão direita).

Para uma junta rotacional, d_i , a_i , e α_i são os parâmetros da junta, variando o seu valor na rotação do elo i em relação ao elo $i-1$. Para uma junta prismática θ_i , a_i e α_i são os parâmetros da junta, enquanto d_i é a variável de junta (deslocamento linear). Obtenção da Matriz de Transformação Homogênea A_i^{i-1}

3.1.3 Algoritmo para obtenção de coordenadas para o elo

Dado um manipulador com N graus de liberdade, o algoritmo descrito a seguir, determina um sistema de coordenadas ortonormais para cada elo do robô, a partir do sistema de coordenada fixo a base de suporte (sistema inercial) até o seu elemento terminal. As relações entre os elos adjacentes podem ser representadas por uma matriz de transformação homogênea 4×4 . O conjunto de matrizes de transformação homogêneo permite a obtenção do modelo cinemático do robô.

ALGORITMO :

D1 - Obtenção do sistema de coordenadas da base: Estabelecer um sistema ortonormal de coordenadas (X_0, Y_0, Z_0) na base de suporte com o eixo Z_0 colocado ao longo do eixo de movimento da junta 1 apontando para o ombro do braço do robô. Os eixos X_0 e Y_0 podem ser convenientemente estabelecidos e são normais ao eixo Z_0 .

D2 - Inicialização e iteração: Para cada $i, i = 1, \dots, N-1$, efetuar passos D3 até D6.

D3 - Estabelecer o eixo das juntas: Alinhar Z_i com o eixo de movimento (rotação ou translação) da junta $i+1$. Para robôs tendo configurações de braço esquerdo-direito, os eixos Z_1 e Z_2 são apontados sempre para o ombro e o tronco do braço do robô.

D4 - Estabelecer a origem do i -ésimo sistema de coordenadas: Situar a origem do i -ésimo sistema de coordenadas na interseção dos eixos Z_i e Z_{i-1} ou na interseção da normal comum entre os eixos Z_{i-1} e Z_i .

D5 - Estabelecimento do eixo X_i : Estabelecer $X_i = \pm(Z_{i-1} \times Z_i)/\|Z_{i-1} \times Z_i\|$ ou ao longo da normal comum entre os eixos Z_i e Z_{i-1} quando eles forem paralelos.

D6 - Estabelecimento do eixo Y_i : Determina-se $Y_i = \pm(Z_i \times X_i)/\|Z_i \times X_i\|$ para completar o sistema de coordenadas. (Estender os eixos Z_i e X_i se necessário para passos D9 a D12).

D7 - Estabelecer a direção do sistema de coordenadas: Normalmente a n -ésima junta é uma junta rotativa. Estabelecer Z_n ao longo da direção do eixo Z_{n-1} apontando para fora do robô. Estabelecer X_n assim que ele é normal tanto aos eixos Z_{n-1} e Z_n . Determine Y_n para completar o sistema de coordenadas.

D8 - Encontrar os parâmetros das juntas e elos: Para cada $i, i = 1, \dots, n$, efetuar passos D9 ao D12.

D9 - Encontrar d_i : d_i é a distância da origem do $(i-1)$ -ésimo sistema de coordenadas até a interseção do eixo Z_{i-1} e o eixo X_i ao longo do eixo Z_{i-1} . Ela é a variável de junta se a junta i é prismática.

D10 - Encontrar a_i : a_i é a distância da interseção do eixo Z_{i-1} e o eixo X_i para a origem do i -ésimo sistema de coordenadas ao longo do eixo X_i .

D11 - Encontrar θ_i : θ_i é o ângulo de rotação entre os eixos X_{i-1} e X_i sobre o eixo Z_{i-1} . Esta é a variável de junta se a junta é rotacional.

D12 - Encontrar α_i : α_i é o ângulo de rotação entre os eixos z_{i-1} e z_i no eixo x_i .

3.1.4 Obtenção da Matriz de Transformação Homogênea A_i^{i-1}

Uma vez os sistemas de coordenadas D-H tenham sido estabelecidos, uma matriz de transformação homogênea pode facilmente ser desenvolvida relacionando o i -ésimo ao $(i-1)$ -ésimo frame de coordenadas. A figura 2 mostra que um ponto r_i expresso no i -ésimo sistema de coordenadas pode ser expresso no $(i-1)$ -ésimo sistema de coordenadas como r_{i-1} aplicando as transformações sucessivamente apresentadas a seguir:

1. Rotação no eixo Z_{i-1} de um ângulo de ϑ_i para alinhar o eixo X_{i-1} com o eixo X_i (o eixo X_{i-1} é paralelo ao eixo X_i e aponta para a mesma direção).
2. Translação uma distância de d_i ao longo do eixo Z_{i-1} para trazer os eixos X_{i-1} e X_i na coincidência.
3. Translação ao longo do eixo X_i uma distância de a_i para trazer as duas origens também como o eixo X na coincidência.
4. Rotação do eixo X_i um ângulo de α_i para trazer os dois sistemas de coordenadas na coincidência.

Cada uma destas quatro operações pode ser expressa através de uma matriz homogênea de rotação-translação, e o produto destas quatro matrizes de transformações elementares produzem uma matriz de transformação homogênea composta ${}^{i-1}A_i$, conhecida como matriz de transformação de D-H, para sistemas de coordenadas adjacentes, i e $i-1$.

$${}^{i-1}A_i = T_{z,d}, T_{z,\theta} T_{x,a} T_{x,\alpha} \quad (3.1)$$

A transformação direta será:

$$[{}^{i-1}A_i]^{-1} = {}^i A_{i-1} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -a_i \\ -\cos\alpha_i \sin\theta_i & \cos\alpha_i \cos\theta_i & \sin\alpha_i & -d_i \sin\alpha_i \\ \sin\alpha_i \sin\theta_i & -\sin\alpha_i \cos\theta_i & \cos\alpha_i & -d_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

onde a_i, α_i, d_i são constantes, e θ_i é a variável de junta para uma junta rotativa.

Para uma junta prismática a variável de junta é d_i , enquanto a_i, α_i, θ_i são constantes. Neste caso, será definido como:

$${}^{i-1}A_i = T_{z,\theta} T_{z,d} T_{x,\alpha} \quad (3.3)$$

e sua inversa será:

$$[{}^{i-1}A_i]^{-1} = A_{i-1}^i \quad (3.4)$$

3.1.5 Matriz Transformada T

A descrição cinemática completa de uma cadeia articulada pode ser obtida a partir do produto matricial entre as diversas matrizes de transformações homogêneas. De acordo com Hermini (2004), usando a matriz ${}^{i-1}A_i$, podemos relacionar um ponto X_i no elo i , e expressar em coordenadas homogêneas, em relação aos sistemas de coordenadas $i-1$, X_{i-1} estabelecido no elo $i-1$ através da relação:

$$X_{i-1} = {}^{i-1}A_i X_i \quad (3.5)$$

onde

$$X_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1}) \text{ e } X_i = (x_i, y_i, z_i)^T$$

Para simplificarmos a notação a matriz ${}^{i-1}A_i$ será designada A_i . Utilizando-se essa relação de modo recorrente podemos escrever:

$$X_{i-2} = A_{i-2} \cdot X_{i-1} = A_{i-2} \cdot A_i \cdot X_i \quad (3.6)$$

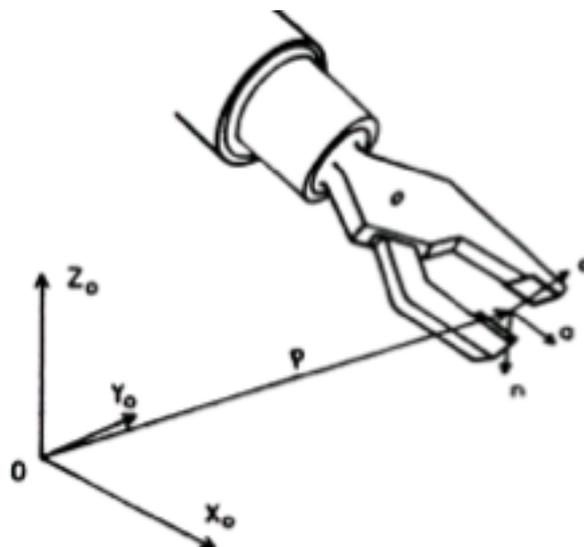
$$X_{i-3} = A_{i-2} \cdot A_{i-1} \cdot A_i X_i$$

...

$$X_0 = A_1 \cdot A_2 \cdot A_3 \dots A_i \cdot X_i$$

Para um robô com seis graus de liberdade, a transformação de coordenadas do referencial situado na base do robô ao referencial situado no seu elemento terminal (ferramenta) é descrito pela matriz de transformação homogênea $T_6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6$. A figura 4 ilustra as coordenadas cartesianas que expressam a posição do elemento terminal de um robô (P_x , P_y , P_z) e sua orientação espacial especificada através das componentes dos vetores de orientação n , s e a .

Figura 5 – Configuração do elemento terminal de um robô



Fonte:(HERMINI, 2004)

3.1.6 Cinemática Inversa

A necessidade da obtenção de referências em coordenadas angulares, correspondentes a tarefas definidas no espaço cartesiano é expressa matematicamente pela inversão do modelo geométrico, isto é:

$$\theta = f^{-1}(x) \quad (3.7)$$

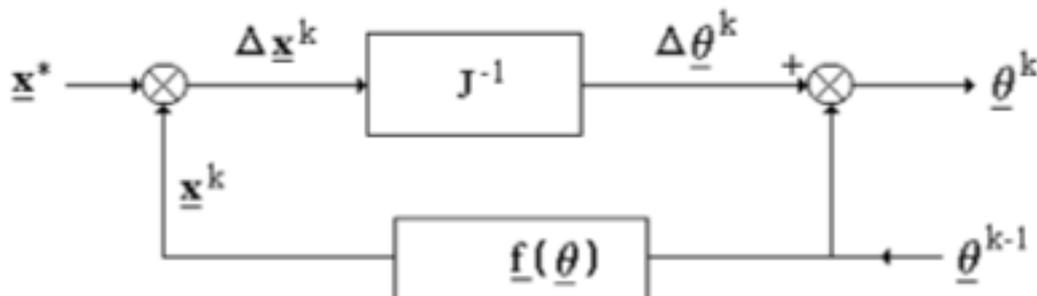
A função f é não linear e composta de soma de produtos de senos e cosenos das coordenadas generalizadas (translações ou rotações elementares). Por isso, a sua inversão é em geral não trivial. Como f é não linear não se pode garantir a existência e/ou a unicidade de uma função inversa f^{-1} . No caso geral, só se pode determinar o número máximo de prováveis soluções.(MALAQUIAS; ASSOLARI; NOGAWA, 2013)

Os métodos de solução do problema da inversão do modelo geométrico são:

- Métodos analíticos: Estes métodos conduzem à obtenção de todas as soluções. Estes métodos não são gerais, isto é, a inversão analítica não é trivial e, além disso, não há garantia de que seja possível fazê-la para um robô qualquer. Os métodos analíticos são adequados para robôs simples, isto é, aqueles que possuem um grande número de parâmetros de Denavit-Hartenberg nulos.
- Métodos numéricos iterativos: Estes métodos convergem para uma solução possível entre todas as existentes, são de caráter geral e, com o atual desenvolvimento dos microcomputadores, a utilização destes métodos em tempo real é viável.

Existem diversos métodos numéricos iterativos, entre eles os métodos recursivos, figura 6, que utiliza ao cálculo do modelo geométrico direto e da matriz Jacobiana inversa, descrito a seguir.

Figura 6 – Utilização do modelo geométrico direto e da matriz Jacobiana inversa para determinar uma configuração ϑ^* correspondente a uma situação desejada \underline{x}^*



Fonte: (MALAQUIAS; ASSOLARI; NOGAWA, 2013)

3.1.7 Descrição Matemática de um robô com N graus de liberdade

A transformação de coordenadas de um robô com n graus de liberdade pode ser formulada da seguinte maneira. A partir de uma configuração inicial do robô, na qual a suas variáveis articulares θ^o são conhecidas, a posição completa de seu elemento terminal (ou ferramenta de trabalho) X^o será conhecida a partir do modelo do sistema.

A mudança de coordenadas consistirá de um funcional que descreverá a correspondência existente entre a cadeia cinemática para um conjunto de variáveis articulares ϑ e sua posição X correspondente.

$$x - x_0 = F(\theta - \theta^o) \tag{3.8}$$

onde o vetor F possui $n \leq 6$ componentes descrevendo a posição e orientação do elemento terminal do robô (no caso de $n = 6$).

No caso da transformação inversa de coordenadas, uma determinada posição X do volume de trabalho do robô será atingida pelo robô a partir de uma posição de repouso x_0 . Esta equação não apresentará uma solução única, e a mesma poderá ser utilizada para o controle cinemático de mecanismos.

$$(\theta - \theta^o) = F^{-1}(x - x_0) \tag{3.9}$$

A transformação direta de coordenadas não apresenta dificuldades na sua resolução, o mesmo não acontecendo com a transformação inversa que é muito complexa, não apresentando uma solução única. Para eliminarmos as indeterminações que aparecem no problema inverso, utiliza-se geralmente a matriz jacobiana, onde a mesma poderá ser utilizada para o controle cinemático de mecanismos.

3.1.8 Matriz Jacobiana

Dado uma configuração inicial θ^o e X^o de um robô, as coordenadas X do elemento terminal são descritas pela equação 3.10. Para pequenos deslocamentos δx associados aos deslocamentos das variáveis articulares $\delta\theta$ podemos escrever:

$$\delta\theta = J^{-1}\delta X \quad (3.10)$$

onde:

n: número de graus de liberdade do robô (coordenadas articuladas)

m: número de graus de liberdade consideradas no espaço de trabalho (coordenadas cartesianas)

A matriz Jacobiana $J(\theta)$ será definida como:

$$J(\theta)_{i,j} = dF_i/d\theta_j \quad (3.11)$$

que poderá ser construída a partir das relações cinemáticas que descrevem a arquitetura de um manipulador:

$$X_1 = F_1(\theta_1, \theta_2, \dots, \theta_n) \quad (3.12)$$

$$X_2 = F_2(\theta_1, \theta_2, \dots, \theta_n)$$

...

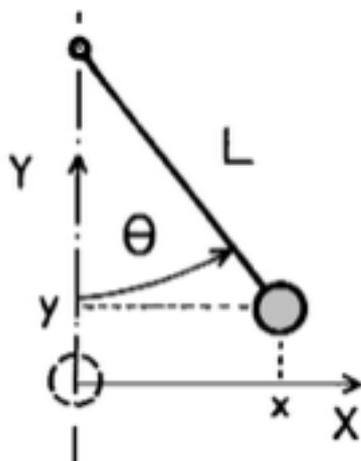
$$X_n = F_n(\theta_1, \theta_2, \dots, \theta_n)$$

Através de derivadas parciais, a matriz Jacobiana $J(\theta)$ será definida como:

$$J(\theta) = \begin{bmatrix} \frac{dF_1}{d\theta_1} & \frac{dF_1}{d\theta_2} & \cdots & \cdots & \frac{dF_1}{d\theta_n} \\ \frac{dF_2}{d\theta_1} & \frac{dF_2}{d\theta_2} & \cdots & \cdots & \frac{dF_2}{d\theta_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{dF_n}{d\theta_1} & \frac{dF_n}{d\theta_2} & \cdots & \cdots & \frac{dF_n}{d\theta_n} \end{bmatrix} \quad (3.13)$$

Referente a manipuladores com 1GL e 2GL:

Figura 7 – Pêndulo simples (1GL)



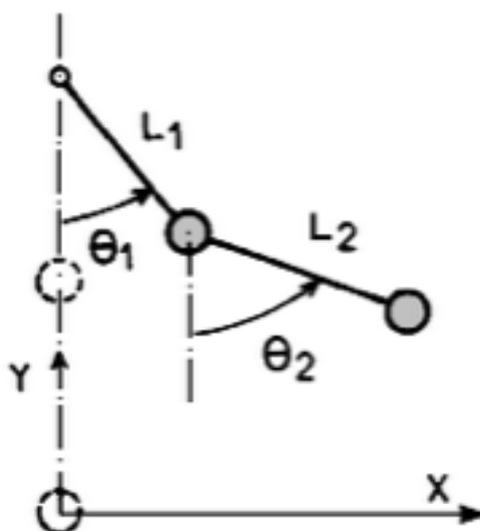
Fonte:(MALAQUIAS; ASSOLARI; NOGAWA, 2013)

$$X_1 = L \sin \theta$$

$$X_2 = L(1 - \cos \theta)$$

$$[J(\theta)] = \begin{bmatrix} L \cos \theta \\ L \sin \theta \end{bmatrix} \quad (3.14)$$

Figura 8 – Pêndulo duplo (2 GL)



Fonte:(MALAQUIAS; ASSOLARI; NOGAWA, 2013)

$$J(\theta) = \begin{bmatrix} L_1 \cos \theta & L_2 \cos \theta \\ L_1 \sin \theta & L_2 \sin \theta \end{bmatrix} \quad (3.15)$$

valores angulares ϑ , podendo ser obtidos a partir das variáveis articulares e métodos de correção de parâmetros.

Finalmente, a matriz Jacobiana, utilizada no método recursivo para o cálculo do modelo cinemático inverso, é uma forma multidimensional da derivada e relaciona a velocidade no espaço de juntas à velocidade no espaço cartesiano. A sua solução deverá ser implementada através da utilização de algoritmos numéricos, onde será aproximada por $\Delta_x = J.\Delta\theta$.(SANTOS, 2004)

3.2 Dinâmica dos Manipuladores

A dinâmica de um sistema mecânico permite relacionar a evolução temporal da sua configuração (nomeadamente, a posição, a velocidade e a aceleração) com as forças e momentos que nele atuam. De acordo com a apostila desenvolvida pelo professor Molina et al. (2005), para controlar a posição do manipulador é preciso conhecer as suas propriedades dinâmicas, de modo, a saber, a quantidade de força (ou torque) que deve ser aplicada às juntas para que ele se mova. Pouca força, por exemplo, fará com que o manipulador reaja vagarosamente, enquanto que força demais pode fazer com que o manipulador esbarre em objetos ou vibre em torno da posição desejada. O modelo dinâmico de um manipulador série operando em espaço livre pode ser representado matematicamente por um sistema de equações diferenciais não lineares que, na forma matricial, pode ser dado por

$$\tau = I(q) \cdot q'' + V(q \cdot q') \cdot q' + G(q) \quad (3.17)$$

em que $I(q)$ e $V(q \cdot q')$ representam, respectivamente, as matrizes de inércia e de Coriolis e centrípetos, $G(q)$ representa o vetor de torque gravitacionais aplicados nas juntas. A necessidade de melhor conhecer o comportamento dinâmico de um manipulador aumenta com a complexidade e com as exigências das tarefas a robotizar.

3.2.1 Formulação Lagrangiana da Dinâmica dos Manipuladores

Dentre as formulações existentes para realizar a dinâmica dos manipuladores, a Lagrangiana é uma abordagem "baseada em energia". Desenvolvendo uma expressão para energia cinética de um manipulador, do i -ésimo elo, k_i , tem-se que:

$$k_i = \frac{1}{2}m_i v_{C_i}^T v_{C_i} + \frac{1}{2} {}^i\omega_i^T C_i I_i {}^i\omega_i \quad (3.18)$$

na qual o primeiro termo é energia cinética devido à velocidade linear no centro de massa do elo e o segundo é energia cinética devido à velocidade angular do elo. A energia cinética total do manipulador é a soma dos elos individuais, ou seja,

$$\kappa = \sum_{i=1}^m \kappa_i \quad (3.19)$$

Os ν_{C_i} e os ${}^i\omega_i$ são funções de Θ e $\dot{\Theta}$, portanto, nota-se que a energia cinética de um manipulador pode ser descrita por uma fórmula escalar como uma função da posição e da velocidade da junta, $\kappa(\Theta, \dot{\Theta})$. De fato a energia cinética de um manipulador é dada por

$$\kappa(\Theta, \dot{\Theta}) = \frac{1}{2} \dot{\Theta}^T M(\Theta) \dot{\Theta} \quad (3.20)$$

em que $M(\Theta)$ é a matriz $n \times n$ de massa do manipulador. Uma expressão da forma descrita acima é conhecida como **forma quadrática**, já que quando é expandida a equação escalar resultante é composta unicamente por termos cuja dependência de $\dot{\theta}_i$ é quadrática. Além disso, como a energia cinética total deve ser sempre positiva, a matriz de massa do manipulador é do tipo **positivo-definida**. Matrizes positivo-definidas são as que tem a propriedade de sua forma quadrática ser sempre um escalar positivo. A equação anterior pode ser vista como análoga à expressão familiar para a energia cinética de uma massa pontual:

$$\kappa = \frac{1}{2} m v^2 \quad (3.21)$$

O fato de a matriz de massa de um manipulador ser positivo-definida é análogo ao de uma massa escalar ser sempre um número positivo.

A energia potencial do i -ésimo elo, u_i , pode ser expressa como

$$u_i = -m_i {}^0g^T P_{C_i} + u_{ref_i} \quad (3.22)$$

onde 0g é o vetor gravidade 3×1 , ${}^0P_{C_i}$ é o vetor que localiza o centro de massa do i -ésimo elo de u_{ref_i} é uma constante escolhida de forma que o valor mínimo de u_i seja zero. A energia total armazenada no manipulador é a soma da energia potencial nos elos individuais, ou seja,

$$u = \sum_{i=1}^n u_i. \quad (3.23)$$

Como os ${}^0P_{C_i}$ são funções de Θ , nota-se que a energia potencial de um manipulador pode ser descrita por uma fórmula escalar como função da posição de junta, $u(\Theta)$.

A formulação Lagrangiana da dinâmica fornece meios para derivar as equações de movimento a partir de uma função escalar chamada **Lagrangiana** que é definida como a diferença entre a energia cinética e a energia potencial de um sistema mecânico. Logo, a Lagrangiana de um manipulador é

$$\mathcal{L}(\Theta, \dot{\Theta}) = \kappa(\Theta, \dot{\Theta}) - u(\Theta) \quad (3.24)$$

As equações de movimento para o manipulador são dadas por

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\Theta}} - \frac{\partial \mathcal{L}}{\partial \Theta} = \tau \quad (3.25)$$

em que τ é o vetor $n \times 1$ dos torques de atuador. No caso de um manipulador, essa equação torna-se

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\Theta}} - \frac{\partial \kappa}{\partial \Theta} + \frac{\partial u}{\partial \Theta} = \tau \quad (3.26)$$

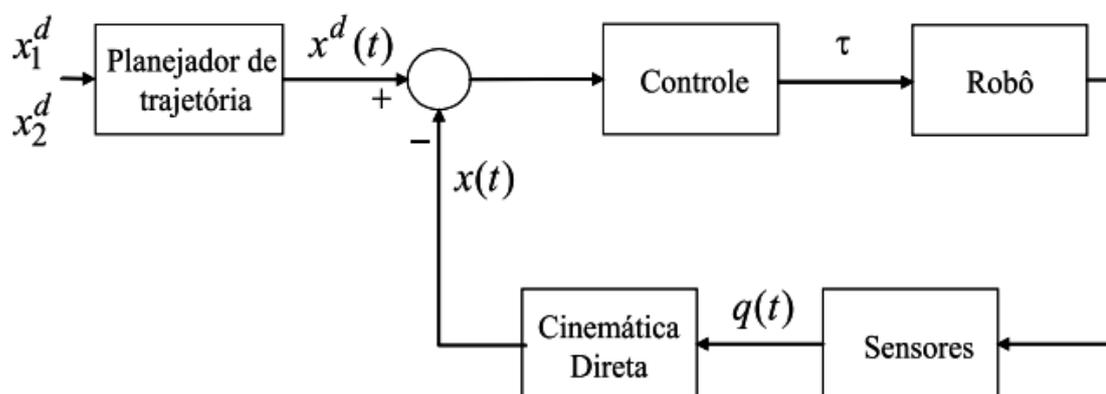
onde os argumentos de $k(\cdot)$ e $u(\cdot)$ foram excluídos por concisão.

3.3 Controle dos Manipuladores

Os robôs podem ser classificados pelo método de controle utilizado.

Molina et al. (2005), relata que pode-se ter robôs com controle em malha aberta, que são os mais antigos, cujos movimentos são limitados por batentes mecânicos. Assim, por exemplo, quando o braço mecânico encontra um batente que limita o seu movimento, esse batente pode acionar um interruptor que desligará o motor da junta e ligará o motor de uma outra junta e assim por diante, até completar o ciclo desejado. Já os robôs modernos são robôs com controle em malha fechada, ou servorobôs, os quais usam um controle computadorizado com realimentação para monitorar o seu movimento. Os servorobôs, por sua vez, são classificados de acordo com o método que o controlador utiliza para guiar o órgão terminal em robôs ponto a ponto (ou robôs PTP, do inglês "*point-to-point*") e robôs de trajetória contínua (ou robôs CP, do inglês "*continuous path*"). Ao robô PTP é ensinado um conjunto de pontos discretos (normalmente através de um TP, o "*Teach Pendant*"), porém não há controle sobre a trajetória que o órgão terminal deve seguir entre dois pontos consecutivos. As coordenadas dos pontos são armazenadas e o órgão terminal passa por eles sem controle sobre a trajetória. Tais robôs são muito limitados em suas aplicações. Já no robô CP toda a trajetória pode ser controlada. Por exemplo, pode ser ensinado ao robô que o seu órgão terminal deve seguir uma linha reta entre dois pontos ou mesmo uma trajetória mais complicada como numa operação de soldagem a arco. Pode-se, também, controlar a velocidade e/ou a aceleração do órgão terminal. Obviamente, os robôs CP requerem controladores e programas mais sofisticados do que os robôs PTP.

Figura 10 – Estrutura de Controle de Robôs Manipuladores



Fonte: Siqueira e Terra (2004)

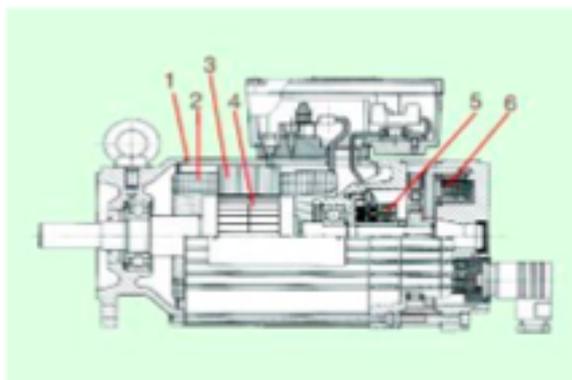
3.4 Servomotor

O estudo do emprego de servomotores vem desde a Segunda Guerra Mundial, quando buscavam evolução em tudo que era feito. Nesse processo de melhoria estava diretamente

relacionado o aumento de demanda Industrial, e conseqüente a melhoria das máquinas até então utilizadas. A melhor forma encontrada foi fazer o *Retrofitting* das mesmas utilizando servomotores, os quais são muito utilizados nesse sentido de melhoria, devido a sua dinâmica, controle e precisão (MATOS et al., 2014).

O Servomotor é um dispositivo eletromecânico possui uma parte fixa (estator) e outra móvel (rotor), como muitas outras máquinas síncronas. O estator possui bastante semelhança ao de uma máquina elétrica convencional, porém com restrições quanto à alimentação, como será verificado posteriormente. O rotor é composto por ímãs permanentes, os quais são posicionados alinhadamente sobre o rotor e com o controlador, ou gerador de sinais, chamado de *resolver*. É possível verificar a estrutura de um servomotor na figura abaixo.

Figura 11 – Principais partes de um Servomotor. Carcaça do Motor (1), Bobinagem do estator (2), Chapas do Estator (3), Rotor com ímãs (4), Resolver (5) e Freio (6)



Fonte: SÁ et al. (2013)

Utilizando a tecnologia de ímãs permanentes, os servomotores podem proporcionar precisão e controle de velocidade e posição, sem contar na grande vantagem de ser possível controlar o torque no eixo, de forma constante e em larga faixa de rotação. Porém para ser possível esse controle de velocidade, posição e torque é necessário o emprego do servoconversor, os quais são desenvolvidos especificamente para otimização dos servomotores. Estes têm a função de controlar o fluxo eletromagnético e fornecê-lo para o Servomotor, pois os fluxos oriundos da rede não apresentam uma conformidade adequada para o acionamento dos servos. Explicando também o bobinamento de estator diferenciado. Outro ponto importante a ser observado é que cada servomotor possui o seu próprio servoconversor.

Outro tipo de servomotor é o assíncrono, que traz a robustez dos motores trifásicos “gaiola de esquilo” e a controlabilidade dos servomotores síncronos, necessitando de um sinal de realimentação. Porém estes obrigam a utilização de ventilação forçada para seu estator, efeito do seu alto desempenho.

Praticamente falando, os servomotores assíncronos são quase idênticos aos motores convencionais na sua forma de construção, diferenciando apenas por uma ponta adicional no eixo do rotor, especificamente para acoplamento em sistemas de controle de posição e velocidade, podendo esse ser um transdutor de posição angular, o encoder, imagem do motor AC Assíncrono na figura a seguir.

Figura 12 – Servomotor AC Assíncrono



Fonte:Tomaschitz (2014)

Outra topologia possível para aplicação de servomotores são os de corrente contínua (DC). Estes outrora foram os mais facilmente encontrados para utilização porém com o tempo os de corrente alternada os substituíram. O funcionamento de servomotor DC é bastante semelhante ao motor DC comum, porém possui um controlador para realimentação. Onde o funcionamento está de acordo com (SCHERPEN; ORTEGA, 1997).

3.5 Microcontrolador

O controle de equipamentos para acionamentos industriais é comumente realizado por intermédio de um microprocessador embarcado no produto, o qual carrega um programa armazenado composto de algoritmos dedicados à aplicação em questão. O tipo de microprocessador que melhor se presta a esta categoria de aplicações costuma ser aquele que integra na mesma pastilha de silício, além da unidade central de processamento (CPU), também circuitos de memória e uma diversidade de circuitos auxiliares (periféricos) dedicados a funções de entrada e saída (E/S) específicas, tais como conversão analógico-digital (A/D) e saídas digitais moduladas por largura de pulso (PWM). Tal tipo de processador é usualmente chamado de microcontrolador (ANDERSON, 2013).

4 MATERIAIS E MÉTODOS

Para realização deste projeto foi desenvolvido um algoritmo, na ferramenta computacional MATLAB®, a fim de validar a modelagem matemática e realizar o controle do manipulador, com intuito de obter a posição do Robô Planar com dois graus de liberdade. Para o desenvolvimento do modelo matemático foram necessários os conhecimentos de cinemática, pelo método de Denavit–Hartenberg, dinâmica, com a formulação de Lagrange e, controle de robô, utilizando os controladores Proporcional, Integral e Derivativo (PID). O controle é realizado por meio de um sistema de *software* e *hardware*, para a comunicação entre ambos foi utilizado a plataforma de Arduino® onde é executado o algoritmo de controle.

Para melhor entendimento do usuário com o sistema, foi desenvolvida uma interface gráfica na ferramenta GUI (*Graphical User Interface*), a fim de obter uma interface humano-máquina para o projeto.

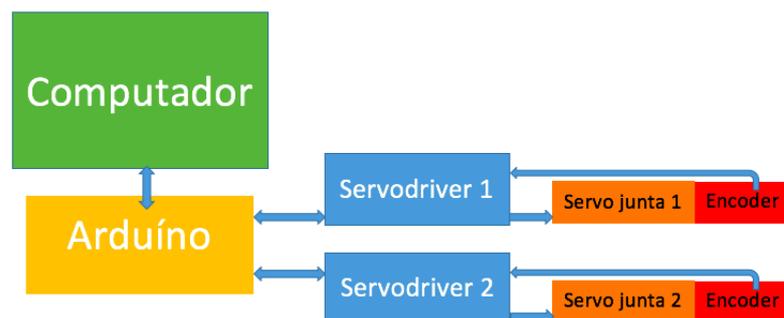
A descrição dos materiais e equipamentos utilizados para a proposta do Hardware, foram pesquisados e discutidos até obter todos os periféricos necessários para proposta de controle e construção do robô. Teste de ligação elétrica dos motores foram efetuadas e obtidos os resultados os quais serão apresentados no decorrer do capítulo.

4.1 Projeto do Robô Planar

A proposta desta monografia é desenvolver um projeto para o controle de posicionamento de um Robô Planar com dois graus de liberdade, contendo a parte mecânica, com a definição de uma proposta de estrutura mecânica dos elos e juntas, e modelagem matemática, tendo como base os parâmetros adotados, como também a parte eletrônica, dimensionando e selecionando os componentes eletro-eletrônicos necessários ao funcionamento do sistema que irá compor o Robô Planar.

Componentes básicos do Robô Planar com dois graus de liberdade:

Figura 13 – Componentes do Robô Planar

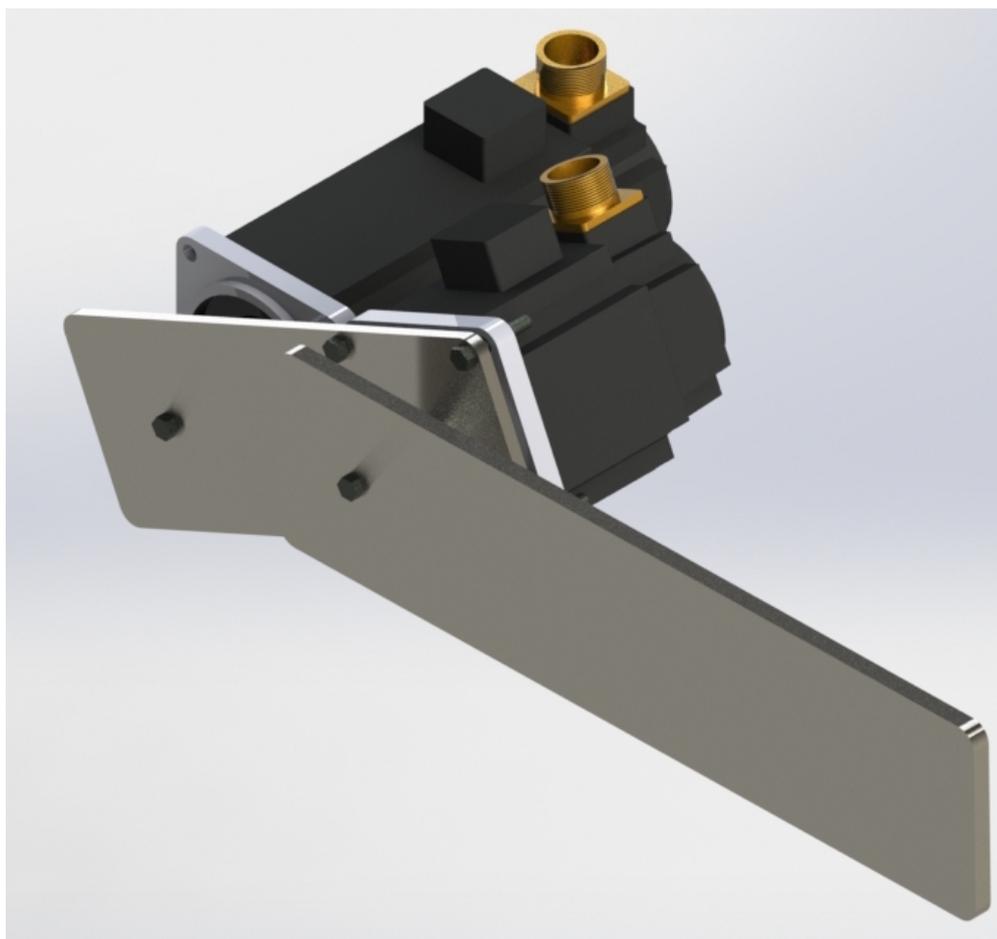


Fonte:Autora

Essa estrutura da figura 13 será descrita nas seções posteriores, assim como os materiais utilizados para execução, os quais estão disponíveis em laboratório.

O modelo proposto do Robô Planar, segue na figura 14, composto por dois servomotores, com dois elos, o que representa os dois graus de liberdade. Neste caso, o primeiro elo deverá ser de um material rígido, pois o mesmo irá controlar o segundo motor que por sua vez, conduzirá o segundo elo, de material mais leve, como o alumínio.

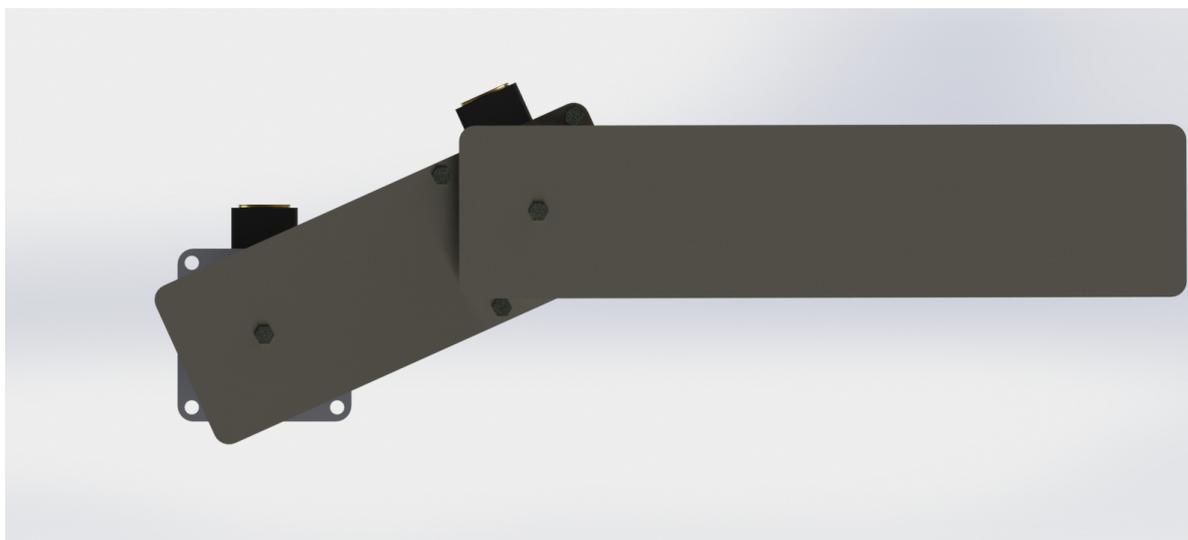
Figura 14 – Projeto Proposto



Fonte: Autora

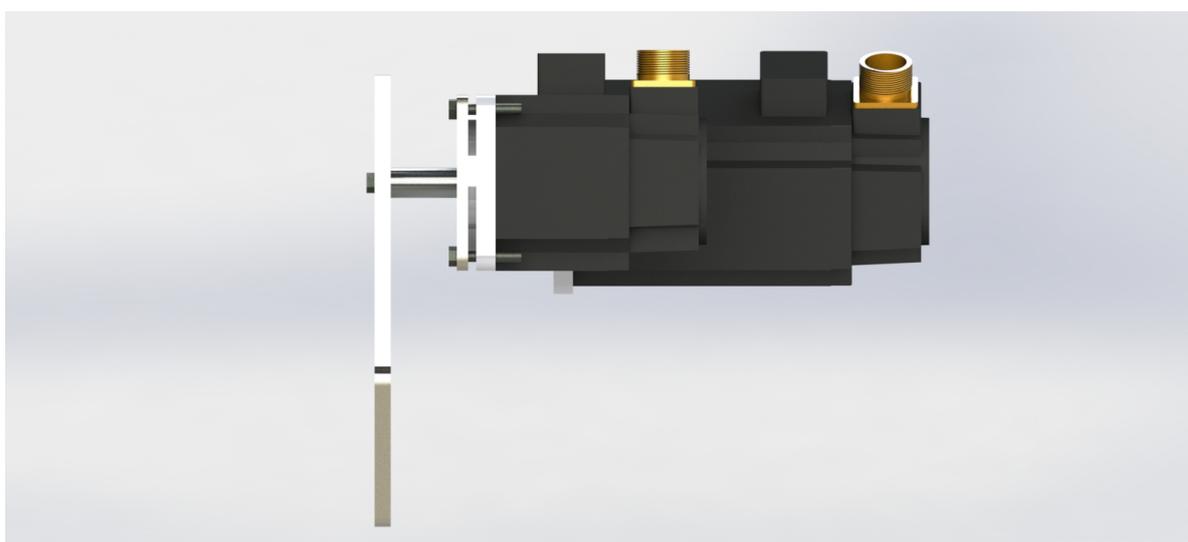
O elo pode ser fixado ao eixo utilizando apenas um parafuso de mesma medida de rosca e na outra extremidade, há quatro furos nos quais é possível fixar o *encoder* com auxílio de quatro porcas. O furo central serve para passar o eixo do *encoder* no qual deve ser fixado o segundo elo. Salienta-se o posicionamento simétrico dos furos transversalmente e longitudinalmente.

Figura 15 – Projeto das juntas



Fonte: Autora

Figura 16 – Conexão das juntas



Fonte: Autora

Tabela 1 – Parâmetros nominais do elo

Parâmetros	Elo
Massa 1 (Kg)	1
Comprimento 1 (m)	0.20
Massa 2 (Kg)	0.3
Comprimento 2 (m)	0.40

Fonte: Autora

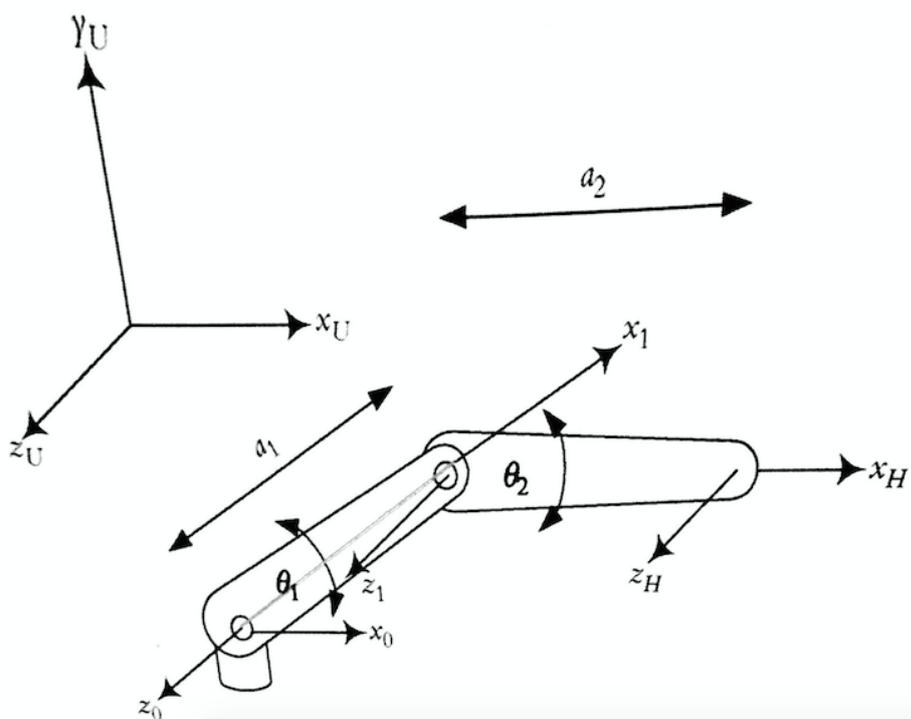
4.2 Modelagem

Para simulação computacional do projeto, é necessário a criação de um modelo matemático que determine o controle de posição do Robô Planar, com dois graus de liberdade, por meio das matrizes de transformação homogênea, para tanto, escolheu-se o algoritmo de Denavit-Hartenberg com ferramenta de apoio. Os detalhes do algoritmo e da modelagem do robô podem ser observados a seguir.

4.3 Modelagem Cinemática

Para se obter os parâmetros de posição e orientação do Robô Planar foi realizada a modelagem cinemática do manipulador, onde será apresentada a seguir:

Figura 17 – Robô Planar de 2 eixos



Fonte: SAEED

Tabela 2 – Tabela de Parâmetros D-H

#	θ	d	a	α
0 - 1	θ_1	0	a_1	0
1 - H	θ_2	0	a_2	0

Fonte: Autora

Como característica do robô, ambas as articulações giram no plano x,y e que um referencial x_H, z_H mostra a extremidade do robô. Primeiro, são atribuídos os eixos z às articulações. O z_0 será atribuído à articulação 1, e z_1 será atribuído à articulação 2. A figura 17 apresenta dois eixos z apontando para fora da página (como são os eixos z_U e z_H). O referencial 0 é fixo e não se move. O robô se move em relação a ele.

Em seguida, são atribuídos os eixos x a cada referencial. Como o primeiro referencial (referencial 0) está na base do robô e, portanto, não existem articulações antes disso, a direção de x_0 é arbitrária. Por conveniência (apenas), pode-se opinar por atribuir a ele a mesma direção que o eixo x do Universo. Não há problema se uma outra direção for escolhida, tudo que isso significa é que, se fosse especificado ${}^U T_H$ em vez de ${}^0 T_H$, teria que incluir uma rotação adicional fixa para indicar que os eixos x_U e x_0 não são paralelos.

Como z_0 e z_1 são paralelos, a normal comum entre eles está na direção entre os dois, e, portanto, o eixo x_1 é como mostrado.

A tabela 2 mostra os parâmetros do robô. Para identificar os valores, foram feitas quatro transformações necessárias para ir de um referencial para o próximo, de acordo com a convenção de D-H:

1. Girar em torno do eixo z_0 de um ângulo θ para fazer x_0 paralelo a x_1 .
2. Como x_0 e x_1 estão no mesmo plano, a translação d ao longo do eixo z_0 é zero.
3. Transladar ao longo do eixo x_0 (já rodado) uma distância a_1 .
4. Como os eixos z_0 e z_1 são paralelos, a rotação em torno do eixo x_1 é zero.

O mesmo pode ser repetido para a transformação entre os referenciais 1 e H.

A equação cinemática direta do robô pode ser encontrada substituindo esses parâmetros nas matrizes A correspondentes como se segue:

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ e } A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Logo,

$${}^0 T_H = A_1 \times A_2 = \begin{bmatrix} C_1 C_2 - S_1 S_2 & -C_1 S_2 - S_1 C_2 & 0 & a_2(C_1 C_2 - S_1 S_2) + a_1 C_1 \\ S_1 C_2 + C_1 S_2 & -S_1 S_2 + C_1 C_2 & 0 & a_2(S_1 C_2 + C_1 S_2) + a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Usando as funções $C_1C_2 - S_1S_2 = C(\theta_1 + \theta_2) = C_{12}$ e $S_1C_2 + C_1S_2 = S(\theta_1 + \theta_2) = S_{12}$, a transformada é simplificada para:

$${}^0T_H = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_2C_{12} + a_1C_1 \\ S_{12} & C_{12} & 0 & a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

Assim, é possível encontrar a posição e orientação do final do robô com os valores de θ_1, θ_2 , a_1 e a_2 especificados. Para isso, são dados os vetores $\mathbf{n}, \mathbf{o}, \mathbf{a}, \mathbf{p}$:

$${}^0T_H = A_1 \times A_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_2C_{12} + a_1C_1 \\ S_{12} & C_{12} & 0 & a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & d_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

Os ângulos podem ser calculados de forma algébrica, ou pelo desacoplamento das incógnitas, foi escolhida a forma algébrica para desenvolver a cinemática inversa do robô. Os valores dos senos e cossenos de um ângulo, são necessários para identificar corretamente o quadrante em que o ângulo está.

O modelo cinemático inverso, por meio do qual obtém-se o estado das juntas em função da posição desejada para a ferramenta do robô. Uma vez que o usuário entre com os valores de X,Y como valores de posição desejadas, a cinemática inversa tem como função transformar estes valores para uma posição angular, de forma que o Robô Planar compreenda e responda ao usuário a posição solicitada. Para isso tem-se:

Modelo matemático associado:

$$X = L_1 \cdot \cos\theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2)$$

$$Y = L_1 \cdot \sin\theta_1 + L_2 \cdot \sin(\theta_1 + \theta_2)$$

A transformação inversa de coordenadas consistirá na definição de um vetor $\theta = (\theta_1, \theta_2)^T$, a partir do posicionamento do robô num determinado ponto $M(X_0, Y_0)^T$, a partir da obtenção dos valores θ_1 e θ_2 expressos em função de X_0 e Y_0 .

Descrição geométrica:

Para θ_1

$$\theta_1 = \beta - \alpha$$

$$\beta = \operatorname{atan2} \frac{y}{x}$$

$$\alpha = \operatorname{atan2} \frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)}$$

$$\theta_1 = \operatorname{atan2} \frac{y}{x} - \operatorname{atan2} \frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)} \quad (4.3)$$

Para θ_2

$$R^2 = X^2 + Y^2 = (L_1 + L_2 \cos \theta_2)^2 + (L_2 \sin \theta_2)^2$$

$$X^2 + Y^2 = L_1^2 + 2L_1 \cdot L_2 \cos \theta_2 + L_1^2 \cos^2 \theta_2 + L_2 \sin^2 \theta_2 = L_1^2 + L_2^2 + 2L_1 L_2 \cos \theta_2$$

$$\cos \theta_2 = \frac{X^2 + Y^2 - L_1^2 - L_2^2}{2L_1 L_2}$$

$$\sin \theta_2 = \pm \sqrt{1 - \cos^2 \theta_2}$$

$$\theta_2 = \operatorname{atan2} \frac{\sin \theta_2}{\cos \theta_2}$$

$$\theta_2 = \operatorname{atan2} \pm \frac{\sqrt{1 - \frac{(X^2 + Y^2 - L_1^2 - L_2^2)^2}{(2L_1 L_2)^2}}}{\frac{X^2 + Y^2 - L_1^2 - L_2^2}{2L_1 L_2}} \quad (4.4)$$

Com isso,

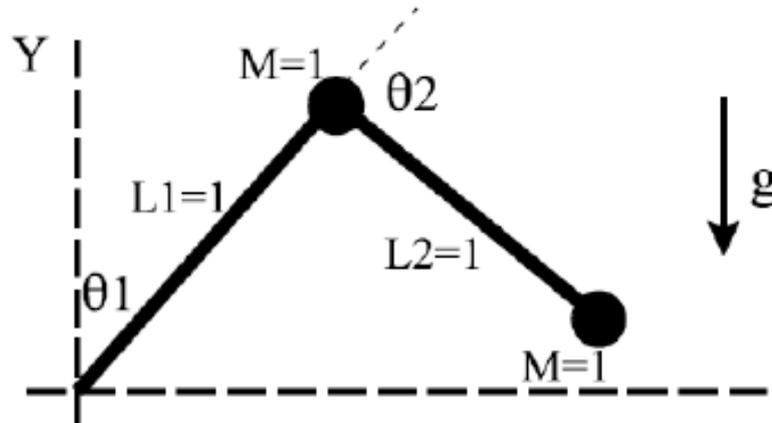
$$\theta = \left(\operatorname{atan2} \frac{y}{x} - \operatorname{atan2} \frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)}, \operatorname{atan2} \pm \frac{\sqrt{1 - \frac{(X^2 + Y^2 - L_1^2 - L_2^2)^2}{(2L_1 L_2)^2}}}{\frac{X^2 + Y^2 - L_1^2 - L_2^2}{2L_1 L_2}} \right)^T \quad (4.5)$$

4.4 Modelagem Dinâmica

Para acelerar os elos de um robô, é necessário ter atuadores capazes de exercer forças e conjugadoe alta demandas suficientemente grandes no elos e juntas para movê-los a uma aceleração e uma velocidade desejadas. Caso contrário, os elos podem não estar se movendo tão rápido quanto necessário e, conseqüentemente, o robô pode não manter a sua precisão posicional desejada. Para calcular o quão forte deve ser cada atuador, é necessário determinar as relações dinâmicas que governam os movimentos do robô. Onde essas relações são as equações de força, massa, aceleração e conjugado, inércia e aceleração angular.

A modelagem dinâmica foi desenvolvida pelo plano cartesiano a seguir:

Figura 18 – Plano Cartesiano de um Robô Planar com dois graus de liberdade



$$x_1 = L_1 \sin \theta_1$$

$$y_1 = L_1 \cos \theta_1$$

$$x_2 = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

$$y_2 = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2)$$

Então, Energia Cinética é formada por:

$$K_E = \frac{1}{2}M_1\dot{x}_1^2 + \frac{1}{2}M_1\dot{y}_1^2 + \frac{1}{2}M_2\dot{x}_2^2 + \frac{1}{2}M_2\dot{y}_2^2$$

Por simplificação,

$$K_E = \frac{1}{2}(M_1 + M_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}M_2L_2^2\dot{\theta}_1^2 + M_2L_2^2\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}M_2L_2^2\dot{\theta}_2^2 + M_2L_1L_2\cos\theta_2(\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1^2) \quad (4.6)$$

E a Energia Potencial é:

$$P_E = M_1gL_1\cos\theta_1 + M_2g(L_1\cos\theta_1 + L_2\cos(\theta_1 + \theta_2)) \quad (4.7)$$

Logo, pela dinâmica de Lagrange calcula-se energia cinética menos energia potencial:

$$\mathcal{L} = K_E - P_E$$

$$\mathcal{L} = \frac{1}{2}(M_1 + M_2)L_1^2\dot{\theta}_1^2 + \frac{1}{2}M_2L_2^2\dot{\theta}_1^2 + M_2L_2^2\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}M_2L_2^2\dot{\theta}_2^2 + M_2L_1L_2\cos\theta_2(\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_1^2) - M_1gL_1\cos\theta_1 + M_2g(L_1\cos\theta_1 + L_2\cos(\theta_1 + \theta_2))$$

Portanto, as equações dinâmicas são:

$$f_{\theta_{1,2}} = \frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{\theta}_{1,2}} \right] - \frac{\partial \mathcal{L}}{\partial \theta_{1,2}} \quad (4.8)$$

Simplificando a equação dinâmica:

$$((M_1 + M_2)L_1^2 + M_2L_2^2 + 2M_2L_1L_2\cos\theta_2)\ddot{\theta}_1 + (M_2L_2^2 + M_2L_1L_2\cos\theta_2)\ddot{\theta}_2 - M_2L_1L_2\sin\theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) - (M_1 + M_2)gL_1\sin\theta_1 - M_2gL_2\sin(\theta_1 + \theta_2) = f_{\theta_1}$$

$$(M_2L_2^2 + M_2L_1L_2\cos\theta_2)\ddot{\theta}_1 + M_2L_2^2\ddot{\theta}_2 - M_2L_1L_2\sin\theta_2\dot{\theta}_1\dot{\theta}_2 - M_2gL_2\sin(\theta_1 + \theta_2) = f_{\theta_2}$$

Assim, o movimento do sistema é descrito da seguinte maneira:

$$F = B(q)\ddot{q} + C(\dot{q}, q) + g(q) \quad (4.9)$$

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

$$B(q) = \begin{bmatrix} ((M_1 + M_2)L_1^2 + M_2L_2^2 + 2M_2L_1L_2\cos\theta_2) & (M_2L_2^2 + M_2L_1L_2\cos\theta_2) \\ M_2L_2^2 + M_2L_1L_2\cos\theta_2 & M_2L_2^2 \end{bmatrix}$$

$$C(\dot{q}, q) = \begin{bmatrix} -M_2L_1L_2\sin\theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ -M_2L_1L_2\sin\theta_2\dot{\theta}_1\dot{\theta}_2 \end{bmatrix}$$

$$g(q) = \begin{bmatrix} -(M_1 + M_2)gL_1\sin\theta_1 - M_2gL_2\sin(\theta_1 + \theta_2) \\ -M_2L_1L_2\sin\theta_2\dot{\theta}_1\dot{\theta}_2 \end{bmatrix}$$

$$F = \begin{bmatrix} f_{\theta_1} \\ f_{\theta_2} \end{bmatrix}$$

Logo,

$$M_1 = M_2 = L_1 = L_2 = 1$$

4.5 Controle do Robô

Segundo Niku (2001), um sistema de controle é usado para controlar o comportamento de um dispositivo, máquina ou processo chamado de planta. A planta deste projeto é Robô planar com dois graus de liberdade, onde sua saída é o movimento do robô. O controlador recebe o sinal de saída e controla a planta e a saída desejada pelo usuário.

O projeto possui um sistema servocontrolado, composto por um controlador em malha fechada, onde há uma operação de soma entre o sinal de realimentação e o sinal de

referência de entrada, onde a subtração do sinal de realimentação e a referência de entrada, resulta no sinal de erro.

Com o sistema de equação:

$$F = B(q)\ddot{q} + C(\dot{q}, q)\dot{q} + g(q)$$

Que também pode ser:

$$\ddot{q} = B(q)^{-1}[-C(\dot{q}, q)\dot{q} - g(q) + \hat{F}]$$

Com,

$$\hat{F} = B(q)^{-1}F \iff F = B(q)\hat{F}$$

Desacoplando o sistema para a nova entrada (não física):

$$\hat{F} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

Contudo, o torque físico das entradas do sistema são:

$$\begin{bmatrix} f_{\theta_1} \\ f_{\theta_2} \end{bmatrix} = B(q) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

O sinal de erro é:

$$e(\theta_1) = \theta_{1f} - \theta_1$$

$$e(\theta_2) = \theta_{2f} - \theta_2$$

Com posição final:

$$\begin{bmatrix} \theta_{1f} \\ \theta_{2f} \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ -\frac{\pi}{2} \end{bmatrix}$$

O sistema tem posições iniciais de:

$$\theta_0 = \begin{bmatrix} -\frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix} \quad (4.10)$$

Estrutura geral do controlador PID para qualquer entrada

$$f = K_p e + K_D \dot{e} + K_I \int e dt \quad (4.11)$$

No caso deste projeto,

$$f_1 = K_{P1}(\theta_{1f} - \theta_1) - K_{D1}\dot{\theta}_1 + K_{I1} \int e(\theta_1) dt$$

$$f_2 = K_{P2}(\theta_{2f} - \theta_2) - K_{D2}\dot{\theta}_2 + K_{I2} \int e(\theta_2) dt$$

Assim, as equações completas de controle são:

$$\ddot{q} = B(q)^{-1}[-C(\dot{q}, q) - g(q)] + \hat{F} \quad (4.12)$$

Com,

$$\hat{F} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} K_{P1}(\theta_{1f} - \theta_1) - K_{D1}\dot{\theta}_1 + K_{I1} \int e(\theta_1) dt \\ K_{P2}(\theta_{2f} - \theta_2) - K_{D2}\dot{\theta}_2 - K_{I2} \int e(\theta_2) dt \end{bmatrix} \quad (4.13)$$

Aplicando todos os controles de ações Proporcional-Integral-Derivativo, um estado "simulado" é adicionado para cada ângulo para assemelhar-se a integração dentro do computador:

$$x_1 = \int e(\theta_1)dt \implies \dot{x}_1 = \theta_{1f} - \theta_1 \quad (4.14)$$

$$x_2 = \int e(\theta_2)dt \implies \dot{x}_2 = \theta_{2f} - \theta_2 \quad (4.15)$$

Assim o sistema de equações são:

$$\dot{x}_1 = \theta_{1f} - \theta_1$$

$$\dot{x}_2 = \theta_{2f} - \theta_2$$

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = B(q)^{-1}[-C(\dot{q}, q) - g(q)] + \begin{bmatrix} K_{P1}(\theta_{1f} - \theta_1) - K_{D1}\dot{\theta}_1 + K_{I1}x_1 \\ K_{P2}(\theta_{2f} - \theta_2) - K_{D2}\dot{\theta}_2 + K_{I2}x_2 \end{bmatrix}$$

No MATLAB, "ode45" comando foi usado para resolver o ODE. (Programa completo no anexo)

Pelo método de sintonia manual, os dois parâmetros controladores foram ajustados para ter o melhor desempenho. Os melhores valores para os parâmetros foi encontrado para ser:

Para θ_1

$$K_{P1} = 15$$

$$K_{D1} = 7$$

$$K_{I1} = 10$$

Para θ_2

$$K_{P2} = 15$$

$$K_{D2} = 10$$

$$K_{I2} = 10$$

Observando a estrutura do ODE acima com controle, pode-se ver:

- K_P é a relação direta com o erro e a velocidade de resposta;
- K_D é a velocidade de intereção com as mudanças de estado;
- K_I é a relação de cancelamento do erro.

No entanto, os argumentos acima são aproximados por causa da alta não linearidade da equação que:

- Produz a interação sensível entre componentes do controlador;
- Pequenas alterações em parâmetros do controlador iria produzir mais superações e oscilações;
- Parâmetros do controlador são altamente sensíveis as posições iniciais e finais;
- Assim, o ajuste *on-line* do PID deve ser considerada para a operação do sistema global (ou seja, posições finais não fixos, controle de trajetória, etc).

Os resultados deste controle será apresentado no capítulo 5.

4.6 Hardware

Nesta seção são descritas as etapas de configuração, ajustes e funcionamento do hardware necessário ao funcionamento do sistema, ligações elétricas, escolha dos componentes da parte eletrônica do projeto, escolha da interface de comunicação do servo com o computador, e testes realizados.

4.6.1 Ligação Elétrica

Após os estudos e cálculos realizados para garantir a proposta deste projeto, foram realizados os primeiros teste com um servomotor modelo HC-KFS73, da Mitsubishi® modelo HC-KFS73, onde servoamplificador é o de modelo MR-J2S-70B, disponíveis no laboratório da instituição. Devido a ligação do servoamplificador ser de alimentação trifásica, foram necessárias alguns ajustes elétricos para efetuar a ligação do servo no laboratório, como adaptação de toda a fiação o equipamento, conforme descrito no manual (MITISUBISHI, 2003). Com isso, o sistema foi energizado.

Com o servo amplificador ligado, foi exibida uma mensagem no *display* do mesmo, indicando um problema, onde de acordo com o manual do fabricante, indicava um problema no *encoder* do equipamento. Tentou-se então efetuar a troca do cabo para resolver o problema, contudo esta ação não foi suficiente, indicando problema no sensor. Assim, outro conjunto de servo estava disponível na universidade, do mesmo fabricante, porem de famílias diferentes, o qual não impedia a realização dos teste do sistema. Por sua vez, o modelo do servomotor HC-RFS503K e o servoamplificador de modelo MR-J2S-500A.

O servoamplificador foi ligado e exibiu a seguinte mensagem: AL. 16. Esta mensagem aparece porque, por segurança, o servomotor possui uma chave de emergência normalmente aberta (NA) e se não há um comando externo desabilitando esta opção, o alarme de emergência, AL. 16, é ativado. Para desativar este alarme, algumas conexões precisam ser

feitas na porta CN1B, uma das portas de controle do servoamplificador. O servoamplificador em questão possui 4 portas para entrada/saída de dados: CN1A, CN1B, CN2 e CN3. Cada porta possui as seguintes funcionalidades:

- CN1A: conector para comandos externos referentes ao modo posição, sensoriamento (pistas do Encoder) e alimentação;
- CN1B: conector para comandos externos referentes aos modos velocidade e torque, alimentação, segurança e habilitação;
- CN2: conector do encoder e;
- CN3: conector para comunicação RS-232C ou RS-422 (computador).

Para conexão também é utilizado um bloco de terminais de parafuso, que vem com um conjunto de *tags* identificadas como MR-J2-A CN1A e CN1B, chamado MR-TB20, o qual possui um rótulo padrão.

Figura 19 – Rótulo padrão do MR-TB20

CN1A										CN1B																													
10	LG	11	PP	12	LZ	13	LB	14	COM	15	OPC	16	PG	17	LZR	18	LBR	19	RD	10	LG	11	VDD	12	SON	13		14	TL	15	P15R	16	COM	17	EMG	18	LSN	19	ZSP
0	NP	1	P15R	2	LA	3	CR	4	SG	5	NG	6	OP	7	LAR	8	INP	9	SD	0	VC	1	DO1	2	TLC	3	PC	4	SG	5	TLA	6	RES	7	LSP	8	ALM	9	SD

Fonte: Mitisubishi

Este periférico é utilizado para facilitar as ligações do servomotor, garantindo um modo de segurança para o usuário.

Figura 20 – MR-TB20 Utilizado no projeto



Fonte: Autora

4.6.2 Funcionamento do sistema servomotor e servoamplificador

Servomotores são compostos por motores DC e um redutor de velocidades, junto com um sensor de posição e um sistema de controle realimentado. Em outras palavras, os servomotores podem ser considerados como sendo motores comandados em posição (angular ou linear), já que, do ponto de vista de quem os utiliza, o controle interno em malha fechada é irrelevante. Os servomotores são pequenos, com ampla variação de torques. O mecanismo de posicionamento ajusta a posição angular por meio de um sinal codificado que lhe é enviado. Enquanto esse código estiver na entrada, o servo irá manter a sua posição angular. Em geral o sinal é do tipo PWM (*Pulse Width Modulation*), ou seja, a posição angular irá depender da largura do pulso enviado.

O conjunto de servos utilizado neste projeto são, respectivamente Servomotor HC-RFS503S e Servoamplificador MR-J2S500A.

Figura 21 – Servomotor HC-RFS503S



Fonte:<http://www.policompcomponentes.com.br>

O conjunto Servomotor e Servoamplificador da Mitsubishi Electric® fornecem o torque ao Robô Planar, sendo atuador e sensor ao mesmo tempo pois o Servomotor possui um transdutor de posicionamento angular de alta resolução acoplado ao seu eixo fornecendo até 131.072 pulsos por revolução.

Figura 22 – Servoamplificador MR-J2S-500A



Fonte: <http://www.nexinstrument.com>

Tabela 3 – Principais especificações do Servoamplificador

Alimentação - Tensão/Frequência	3 fases de 200 a 230VAC, 50/60Hz
Freio Dinâmico	Acoplado-Interno
Max. Freq. pulsos entrada (kpps)	500 (diferencial), 200 (pulsos coletor aberto)
Controle de velocidade (rpm)	Com. analógico: 2000; Com. interno: 5000
Potência de entrada	5kW
Modos de controle	Posição (P), Velocidade (S) e Torque (T)
Modos de posição híbridos	P/S, P/T, S/T
Software de Configuração	MR Configurator - MRZ-JW3 SETUP151E
Saída do Servo Amplificador	170V 0-360Hz, 28.0A
Qtd. Parâmetros de Controle	84

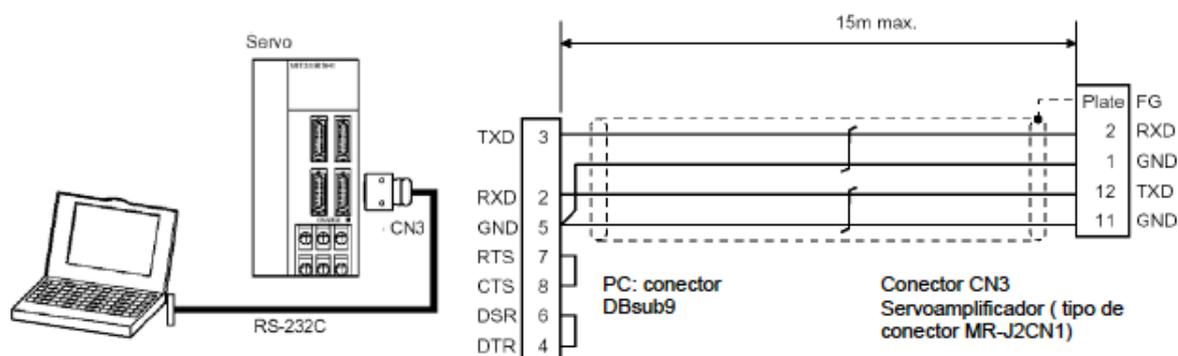
Fonte: Mitisubishi

O servoamplificador em questão possui três modos de funcionamento: posição, velocidade e torque. No modo posição, que é o modo utilizado neste projeto, os pinos PP e NP da porta CN1A trabalham recebendo pulsos de um equipamento externo. Para utilizar este modo, o parâmetro 0 precisar ser configurado como xxx0. A chave referente a SON (servo ON) precisa estar fechada e então o equipamento pode enviar pulsos ao servo. Na seção 4.6.3, será apresentado os testes de funcionamento realizados, onde estará descrito com mais detalhes sobre este modo posição do servoamplificado.

4.6.3 Testes de funcionamento do sistema de servo

O primeiro teste para comunicar o computador com o sistema de servo, foi por meio de um *software* recomendado pelo manual do fabricante, *MR-Configurator* onde é possível realizar testes dos modos de funcionamento, configuração dos parâmetros do servoamplificador e criação de algoritmos por meio da linguagem de programação disponível. A comunicação com este *software* é possível pela porta serial, com o cabo RS-232, conectado a porta CN3, do servoamplificador. A primeira dificuldade encontrada foi com este cabo, pois não estava disponível no laboratório, com isso foi providenciado a confecção deste cabo de comunicação serial, seguindo as instruções do manual, conforme o esquema da figura 23.

Figura 23 – Esquemático do cabo de comunicação RS-232



Fonte: Mitisubishi

Assim, foi possível realizar o teste de comunicação com o *software* indicado. Com a conexão estabelecida, pode-se efetuar alguns testes com o modo velocidade e com o modo posição, onde ao testar o conjunto servo o usuário pode definir a velocidade do eixo do servo em rpm (rotações por minuto), o tempo de aceleração em ms e também um valor de deslocamento angular em pulsos. O valor de deslocamento angular é dado em pulsos por conta da forma como funciona o modo posição, com pulsos. Como a resolução do encoder é de 131.072 pulsos por revolução, se este valor for colocado no *dropdown Move distance* (mover a distância), o eixo do servomotor dará uma volta e assim sucessivamente.

Com citado anteriormente, há uma linguagem de programação disponível para criação de algoritmos, com comandos limitados, impossibilitando por exemplo, a implementação de um controlador dentro da linguagem. Segue a tabela 4 com os comandos da linguagem do *software*.

Tabela 4 – Comandos de programação do MR-Configurator

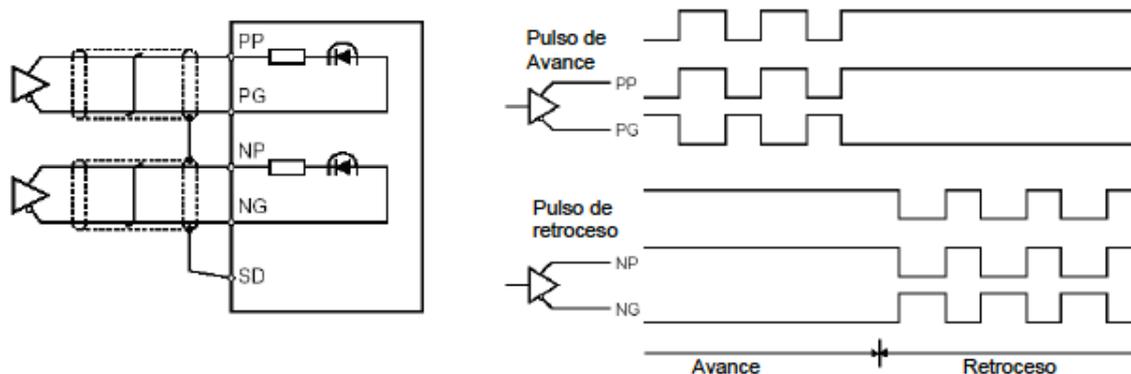
Comando	Descrição
SPN	Configura velocidade do servo
STC	Configura aceleração do servo
MOV	Configura quantidade de pulsos para posicionamento
SYNC	Habilita algumas entradas digitais do servo
TIM	Tem a função de um delay (retardo)
TIMES	Tem a função de uma estrutura de repetição
STOP	Para a execução do programa

Fonte: Mitisubishi

Porém, o *software* possui um protocolo fechado, limitando o acesso ao funcionamento da programação e execução da comunicação computador-sistema servo. Apesar do bom desempenho, o *software* não atende as necessidades do projeto.

O próximo teste realizado com o conjunto de servos, foi com o modo de posição do servoamplificador, utilizando um gerador de funções para enviar pulsos, configurado com pulsos de onda quadrada e frequência variável. O gerador utilizado foi o de modelo MFG-4221 20MHz DDS do fabricante Minip® e para visualizar as formas de onda, foi utilizado um osciloscópio digital, do mesmo fabricante, modelo *Multi Variable bandwidth* MVBDSO. De acordo com manual do fabricante, há dois modos de envio do trem de pulsos, em forma coletor, com frequência máxima de 200kHz e, em modo diferencial, o qual foi utilizado pra verificar o funcionamento, pois este recebe frequência de até 500kHz.

Figura 24 – Envio de trem de pulsos no modo posição



Fonte: Mitisubishi

A entrada PG deve receber sinal simétrico ao que recebe a entrada PP (o mesmo vale para NP e NG). Isto foi feito por meio do gerador de funções, defasando os sinais dos canais do mesmo em 180°. Depois, aplicando uma frequência de 1kHz, o servomotor atuou. O mesmo teste foi realizado com NP e NG e o servo atuou girando em sentido oposto. A frequência do gerador de funções foi variada até seu valor máximo (500kHz), frequência

em que o eixo do servomotor girava com muita velocidade. Um tacômetro não estava disponível para aferir o valor precisamente, mas pode-se dizer que a velocidade máxima estava entre 2000 e 2500 rpm.

No entanto, não há como controlar a frequência do gerador de funções de outra forma que não seja mecanicamente, com isso se viu necessário outro método de atuação para juntas do Robô planar. Para isso, são necessários alguns componentes auxiliares para atuação como um conversor tensão-frequência, ilustrado na figura 25. Para atuar o Robô Planar, foi utilizado este componente para receber sinais elétricos, como tensão ou corrente, que é a forma de comunicação do computador.

Figura 25 – Conversor tensão-frequência UD-C450



Fonte: Autora

O princípio de funcionamento deste equipamento é que, dado um valor de tensão na entrada do circuito, haja uma conversão para um valor de frequência na saída e para variações lineares da tensão de entrada, haja uma variação linear também na saída. Este componente gera frequência de até 50KHz para uma entrada de 10V com um erro associado muito pequeno.

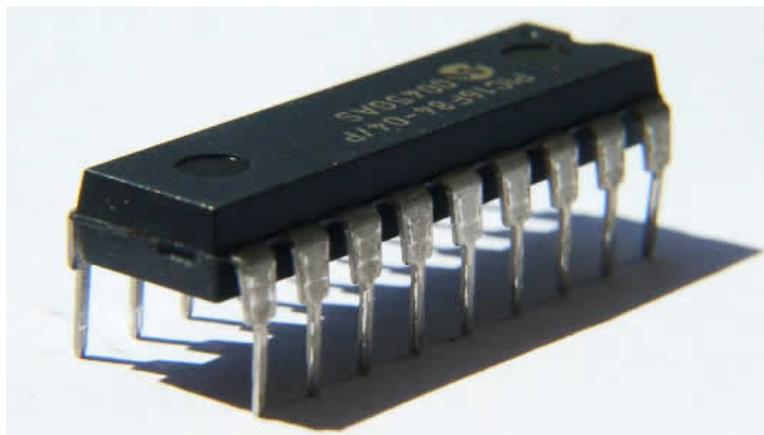
Outro componente auxiliar para completar a atuação, o inversor de sinal. Alguns circuitos integrados podem ser utilizados para esta função, como é o caso de amplificadores operacionais e inversores de sinal.

Para utilizar o amplificador operacional, precisa-se de uma fonte simétrica, que normalmente é obtida juntando duas fontes normais, conectando-se o positivo de uma fonte ao negativo da outra fonte (BOYLESTAD; NASHELSKY, 2004). Entretanto, esta seria uma forma de subutilização das fontes, em relação à utilização de um circuito inversor de sinal.

Optou-se então pelo circuito integrado CD4049UBC buffer hexadecimal, que pela teoria de Fairchild (2002) é monolítica MOS complementar (CMOS) circuitos integrados construídos com N- e modo de melhoramento de canal P transistores. Este dispositivo

apresenta conversão nível lógico utilizando apenas uma tensão de alimentação (VDD). O sinal de entrada alta nível (VIH) pode exceder a tensão de alimentação VDD quando este dispositivo é utilizado para a conversão de nível lógico. Este dispositivo é destinado para uso como buffers hex, CMOS para DTL / Conversores TTL, ou como CMOS drivers atuais, e em VDD = 5.0V, ele pode acionar diretamente duas cargas DTL / TTL sobre a faixa de temperatura operacional completo.

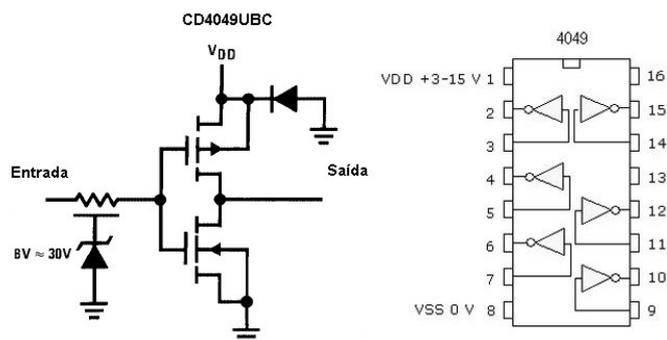
Figura 26 – Circuito Integrado



Fonte: <http://www.bpiropo.com.br/>

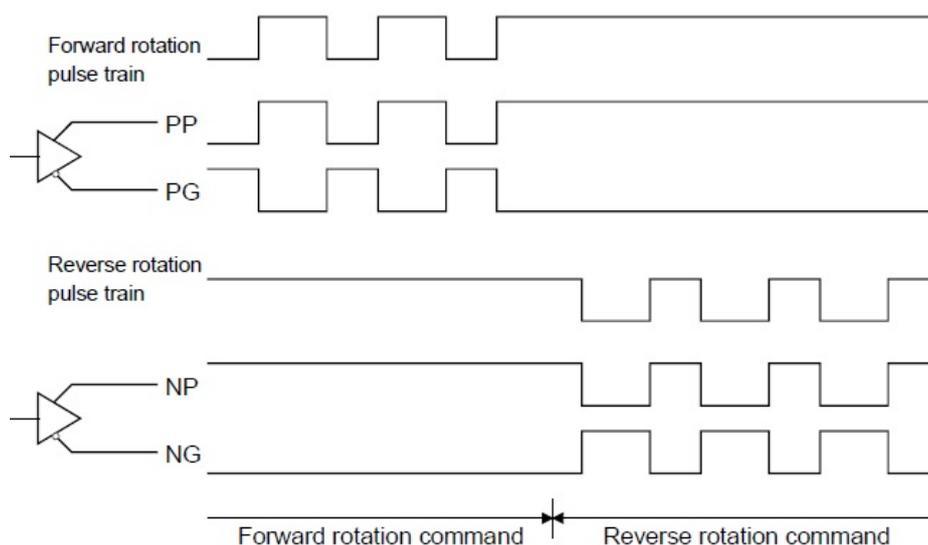
O conversor tensão-frequência fornece apenas um trem de pulsos por dispositivo, sem ajuste de defasagem. Contudo, para utilizar o modo posição do Servoamplificador, é necessário uma saída diferencial, isto é, um sinal sem inversão e um sinal com inversão de fase (MITISUBISHI, 2003), como na figura 27. Por questão de custos, escolheu-se utilizar um CI que realizasse a inversão necessária. A figura 28 mostra o esquema interno deste CI e sua pinagem.

Figura 28 – CI CD4049UBC Hex Inverting Buffer



Fonte: SEMICONDUCTOR, 2002

Figura 27 – Entrada diferencial - Modo Posição do MR-J2S-500A



Fonte: MITSUBISHI, 2007

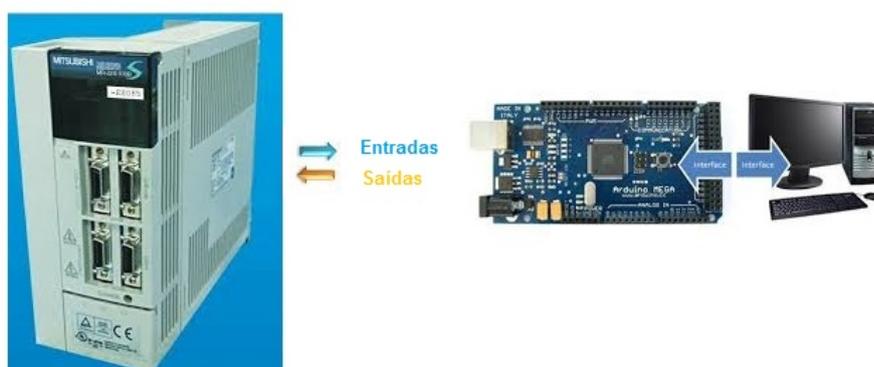
Foram feitas as ligações elétricas pertinentes e depois realizado o teste utilizando a fonte de alimentação de tensão contínua ligada a entrada do conversor V/F e a saída deste dividida em duas, uma indo para o CI e sendo invertida e posteriormente indo para o pino PP e a outra indo para o pino PG, sem sofrer inversão.

O servoamplificador possui um pino de alimentação chamado P15R, que fornece uma tensão de +15V, eliminando a necessidade de fonte de alimentação externa e também 3 tipos de terra (ou massa) a saber, SG (comum das entradas digitais), LG (comum das entradas de controle) e SD (Massa). Portanto pode-se alimentar este CI com as tensões do próprio servoamplificador, sem necessidade de fonte, como é necessário no amplificador operacional. Foram feitas todas as ligações, ligada a fonte de tensão que alimenta o conversor V/F em 5V. E como resultado obteve-se o eixo girando, recebendo pulsos de 25KHz. Logo a atuação foi realizada, alcançando o esperado.

4.7 Interface de comunicação

Para enviar e receber as informações necessárias à execução do processo, é necessária uma interface de comunicação entre o computador, onde executa o programa, e servoamplificador, com ilustrado na figura 29. Um dispositivo que atende as necessidades estabelecidas é o Microcontrolador Arduino®, onde segundo Kurniawan (2013), este atua como uma plataforma eletrônica flexível baseado em código aberto de fácil uso de Hardware e Software.

Figura 29 – Interface de Comunicação

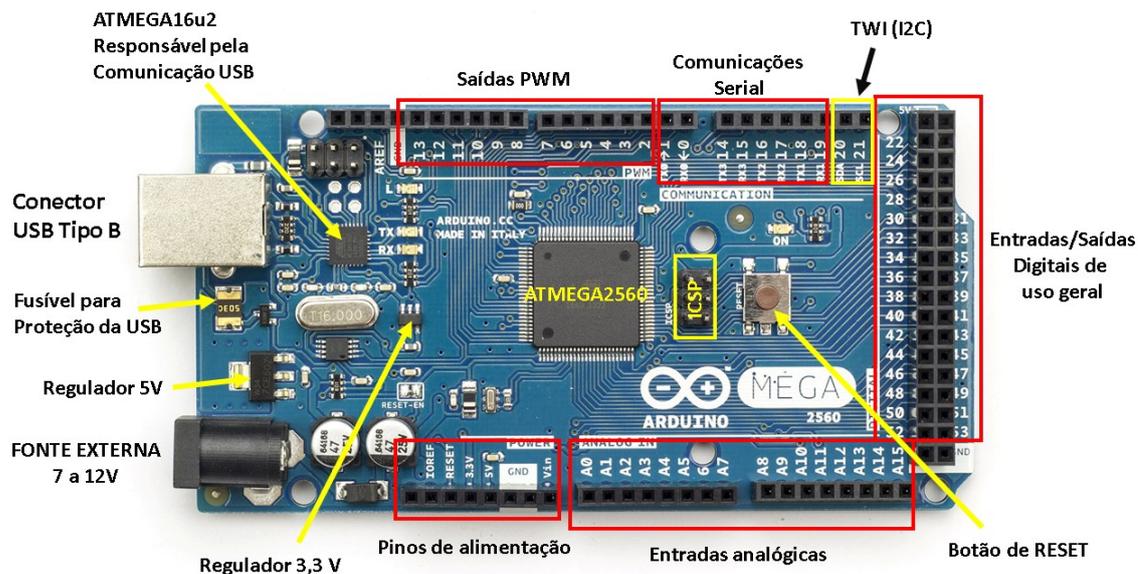


Fonte: <http://brzu.net/06cv8>

4.7.1 Microcontrolador Arduino Mega 2560

Utilizado para efetuar a interface de comunicação entre o computador que executa o controle do sistema e o Servo Amplificador, Arduino é uma boa alternativa no quesito custo-benefício pois possui 54 pinos de entrada-saída, opera com 5V e fornece interface de programação USB, além de 4kB de memória EEPROM e 256kB de Memória Flash (CC, 2015), sendo uma plataforma flexível, de código aberto e de fácil utilização em termos de software e hardware (SUDARMA; SWAMARDIKA; PRANATA, 2015). Arduino possui ainda compatibilidade com Simulink® a partir da versão 2013a, o qual disponibiliza uma biblioteca para utilização deste. Na figura 30, tem-se a pinagem da plataforma Arduino Mega 2560.

Figura 30 – Microcontrolador Arduino Mega 2560



Fonte: <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Tabela 5 – Características principais do Arduino Mega 2560

Descrição	Característica
Microcontrolador	ATmega2560
Tensão de Operação	5V
Tensão de Entrada	7-12V
Pinos Digitais de E/S	54 (15 com função PWM disponível)
Portas Analógicas	16

Fonte: <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Segundo o manual do fabricante Mitisubishi (2003), a resolução do *encoder* do servomotor é configurada como 4000 pulsos/revolução por meio do parâmetro 27 (ENR - *Encoder Output Pulses* ou Pulsos de Saída do *Encoder*), sendo deste valor na verdade $\frac{1}{4}$ é utilizado por questão de construção do fabricante. Contudo, quando a resolução do *encoder* é alterada, tem-se que ter em mente um compromisso precisão x volume de dados, pois uma vez que se está trabalhando com *encoder* do tipo incremental, quanto mais pulsos por revolução se recebe, maior é a precisão do sistema, entretanto maior também é quantidade de pulsos para serem contados e se o hardware (ou mesmo o software) utilizado para efetuar estas contagens possuir limitação na frequência, é certo que muitos pulsos serão perdidos, podendo o sistema não responder da forma como foi projetado ou mesmo tornar-se instável, pois não se pode controlar o que não se pode mensurar (NOY et al.,

2009). Para este projeto, foi configurado a resolução do *encoder* do servomotor para um valor de 3 pulsos por grau, o que corresponde a 1080 pulsos por revolução, ou seja, 4320 pulsos por revolução no servoamplificador, por se acreditar ser um valor equilibrado no compromisso precisão x volume de dados pela deficiência do Arduino Mega 2560.

Para aplicação deste projeto, serão utilizadas as saídas PWM, pois alterando a largura do pulso, podemos alterar o valor médio (V_{RMS}) da amplitude da tensão e variar a frequência da tensão gerada na saída do conversor tensão-frequência, variando assim a velocidade do eixo do servomotor, o que pode ser constatado por meio da equação 4.16.

$$V_{RMS} = \sqrt{V_p^2 \frac{t_p}{T} + V_n^2 \left(1 - \frac{t_p}{T}\right)} \quad (4.16)$$

Sendo, V_p valor de pico superior de tensão, t_p tempo de nível alto de onda, T período da onda e , e V_n o valor de pico inferior de tensão.

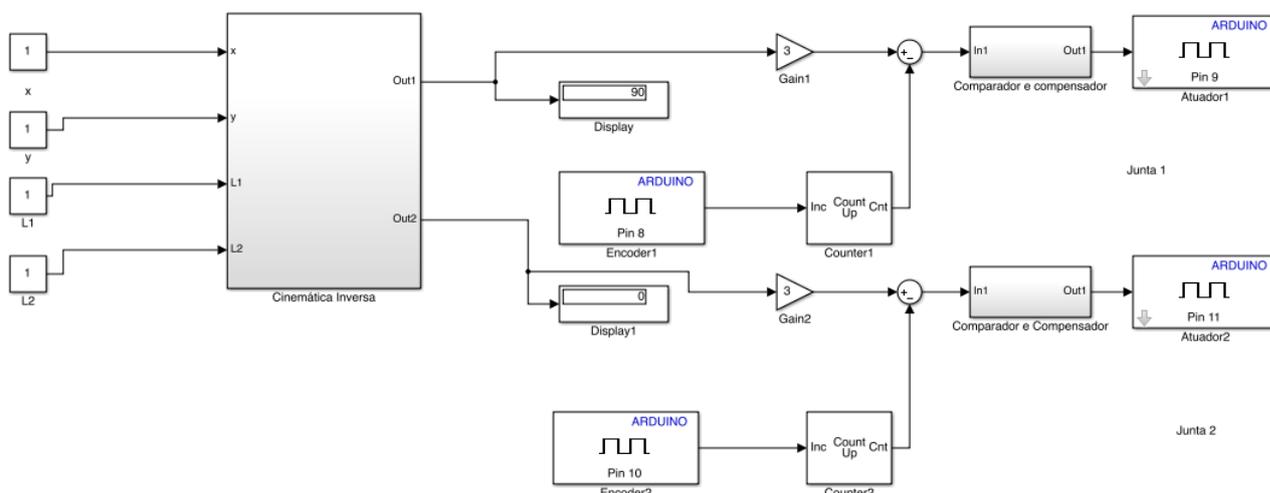
Após realizar testes da funcionalidade do Arduino Mega 2560, ao ter conhecimento da geração/recepção de frequências para o sistema servo, se iniciou a construção de um algoritmo que pode atuar e sensoriar o Robô Planar a fim de validar a proposta de utilizar este sistema do robô de dois graus de liberdade.

O ambiente para criação do algoritmo teste de comunicação do computador com servomotor, foi o ambiente de simulação Simulink®, escolhido por ser um software bastante utilizado no meio acadêmico e por sua capacidade de mostrar o sistema em forma de blocos. E como requisitos para criação do algoritmo tem-se:

- Permitir que o usuário ajuste um setpoint de valor de posição angular desejada, onde o eixo do motor deverá executar;
- Receber os pulsos que vem do *encoder*;
- Armazenar os pulsos do *encoder* para comparação com o valor de referência e ;
- Atua no conversor tensão-frequência até que alcance à posição setada.

Com este algoritmo representado na figura 31, tem-se a validação da proposta de aplicação dos componentes utilizados neste projeto, com a atuação nos elos do robô.

Figura 31 – Algoritmo teste de controle do elo do Robô



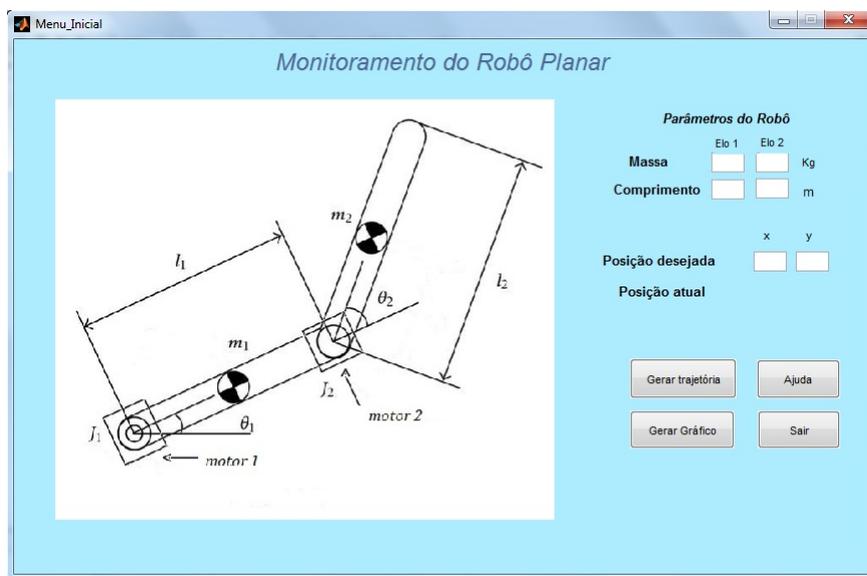
Fonte: Autora

4.7.2 Interface Gráfica

O MATLAB® possui uma ferramenta que permite construir interfaces gráficas de interação com o utilizador, GUI (*Graphical User Interface*), é um aplicativo previsível e compreensível, para que o usuário saiba o que esperar ao executar uma ação. Para iniciar a criação de uma nova interface, basta digitar GUIDE (*GUI Design Environment*) no ambiente de trabalho e de imediato se abre uma janela para a edição do aspecto que irá ter a interface a ser construída. Para facilitar o entedimento do usuário com o sistema, foi criada uma proposta de interface gráfica, para a manipulação do sistema de controle do Robô planar com dois graus de liberdade.

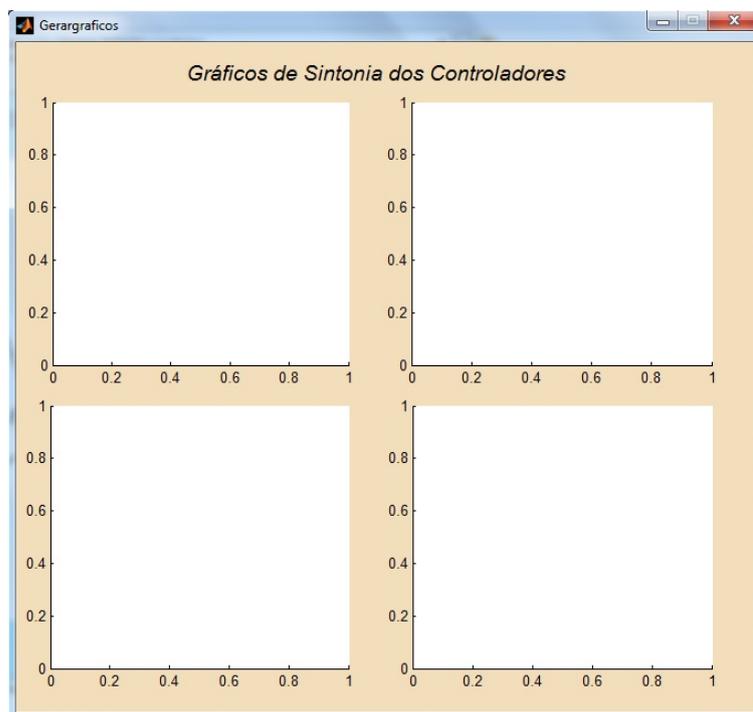
A primeira janela na figura 32, é a tela de menu inicial, onde o usuário entra com os parâmetros do robô, como massas e comprimentos dos elos, e digita a posição desejada. O bloco de posição atual indica onde a posição em que o robô se encontra. Para gerar animação da trajetória, há um botão gerar trajetória, onde será apresentado na janela de trajetória da figura 34. Os gráficos de comportamento do sistema, (onde será abordado no capítulo 5), também serão gerados quando solicitado pelo usuário, a janela está ilustrada na figura 33. Para esclarecer dúvidas básicas de como utilizar o menu inicial, há um botão de ajuda para facilitar o entendimento, como na figura 35.

Figura 32 – Ambiente de controle pelo usuário



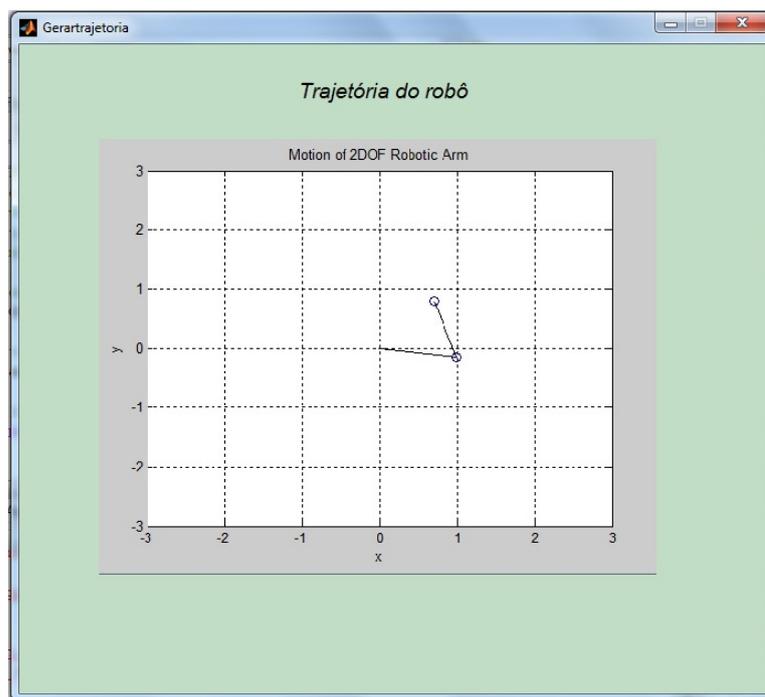
Fonte: Autora

Figura 33 – Janela de resultados



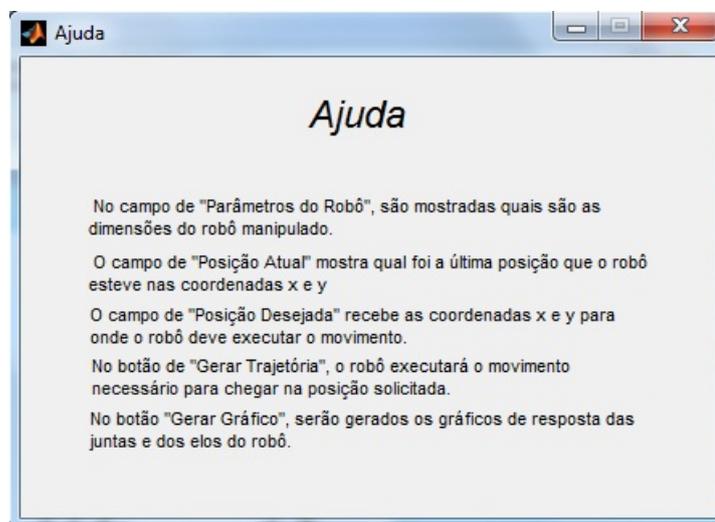
Fonte: Autora

Figura 34 – Janela Trajetória



Fonte: Autora

Figura 35 – Janela Ajuda



Fonte: Autora

5 RESULTADOS OBTIDOS

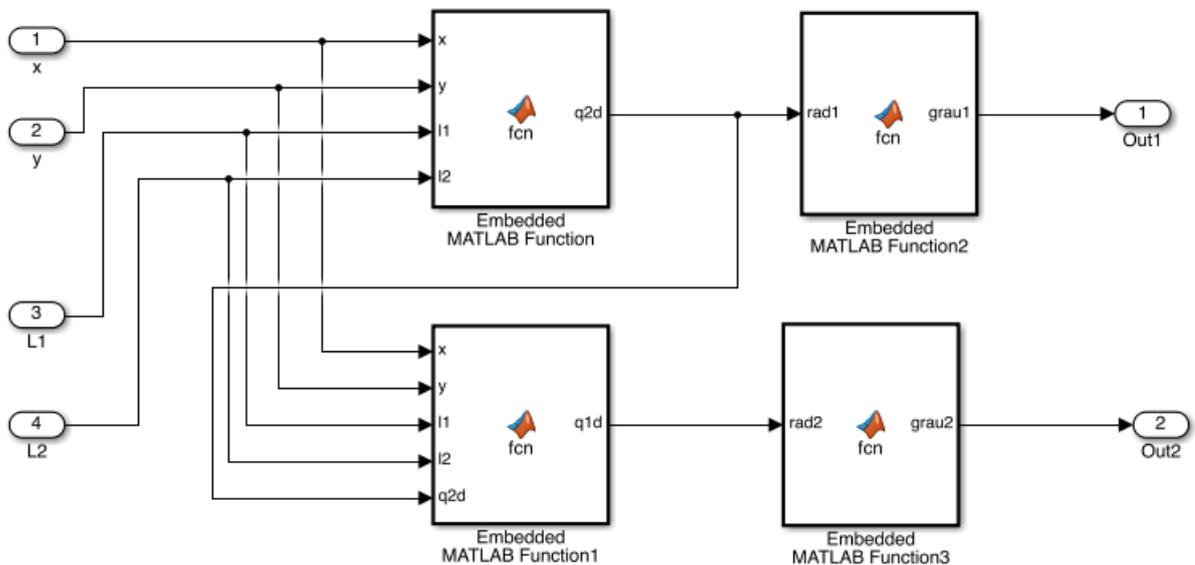
Neste capítulo serão abordados os principais resultados obtidos no projeto, de acordo com a metodologia abordada no capítulo 4.

5.1 Algoritmo teste de controle do elo do Robô

Foi criado um algoritmo em linguagem de diagrama de blocos no Simulink® para validar o funcionamento do Robô planar, como ilustrado na figura 31.

Como aplicação, o usuário pode definir um valor de *setpoint*, como referência de X e Y, onde o próprio algoritmo faz a conversão desses valores para graus, de modo que o sistema responda com a posição desejada. O bloco cinemática inversa, recebe as referências X, Y, L1 e L2 e realiza a cinemática inversa pelo algoritmo representado na figura 36. O cálculo de cinemática inversa está descrito no capítulo 4 desta monografia.

Figura 36 – Algoritmo Cinemática Inversa



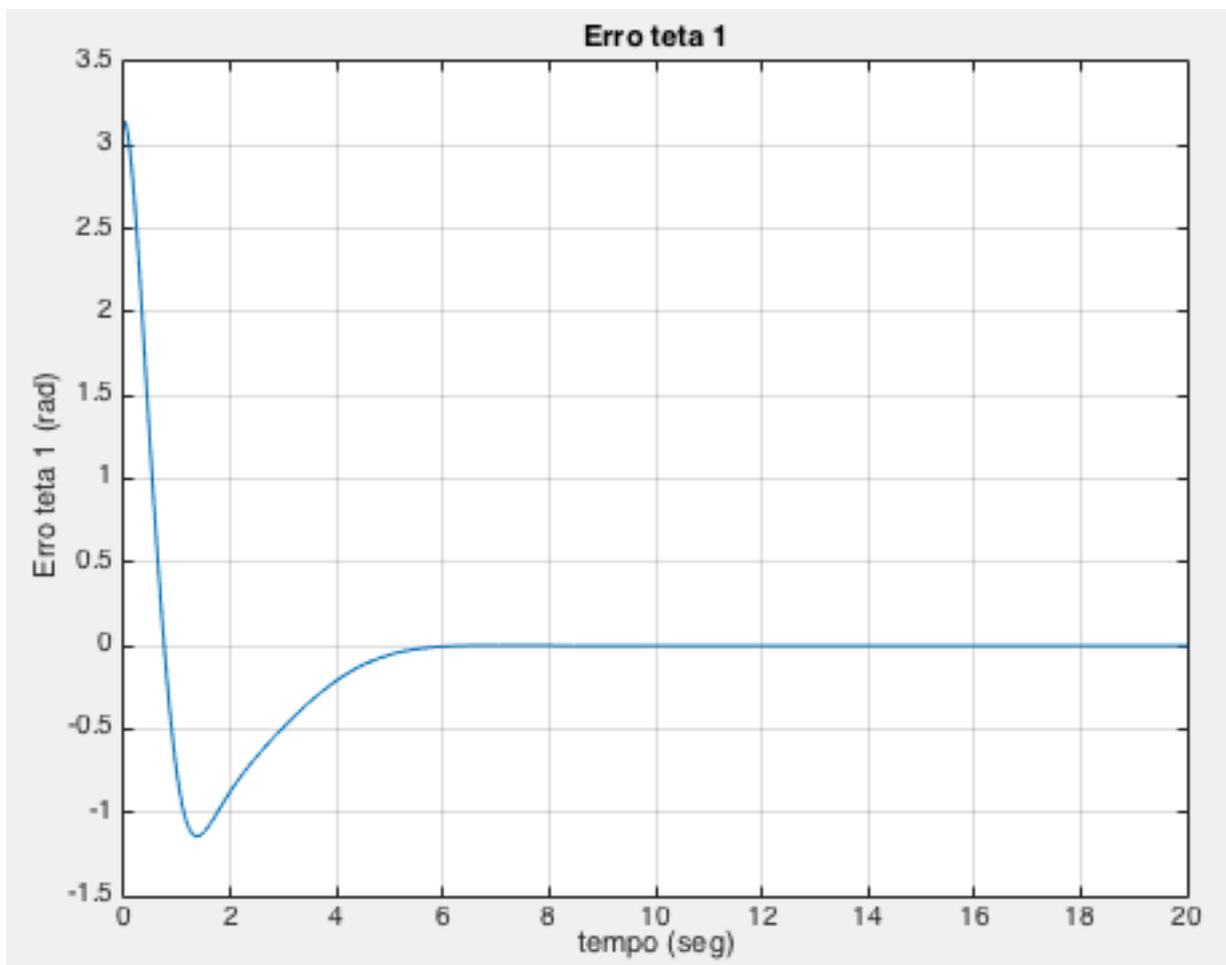
Fonte: Autora

O bloco comparador e compensador, recebe o sinal vindo do somador que efetua a aritmética entre o valor de pulsos acumulado no contador e atua conforme a necessidade no conversor tensão-frequência. Conforme o valor de posição do *encoder* incrementa o contador, tem-se uma diminuição no sinal recebido pelo bloco compensador e comparador, o que ocorre até a chegada no valor de referência. Assim, é validada a proposta de aplicação dos componentes utilizados neste teste como atuação nas juntas do robô planar de dois graus de liberdade. Quanto ao servomotor possui torque suficiente para ser utilizado em aplicações de torque elevado.

5.2 Simulação de controle de posição do robô planar

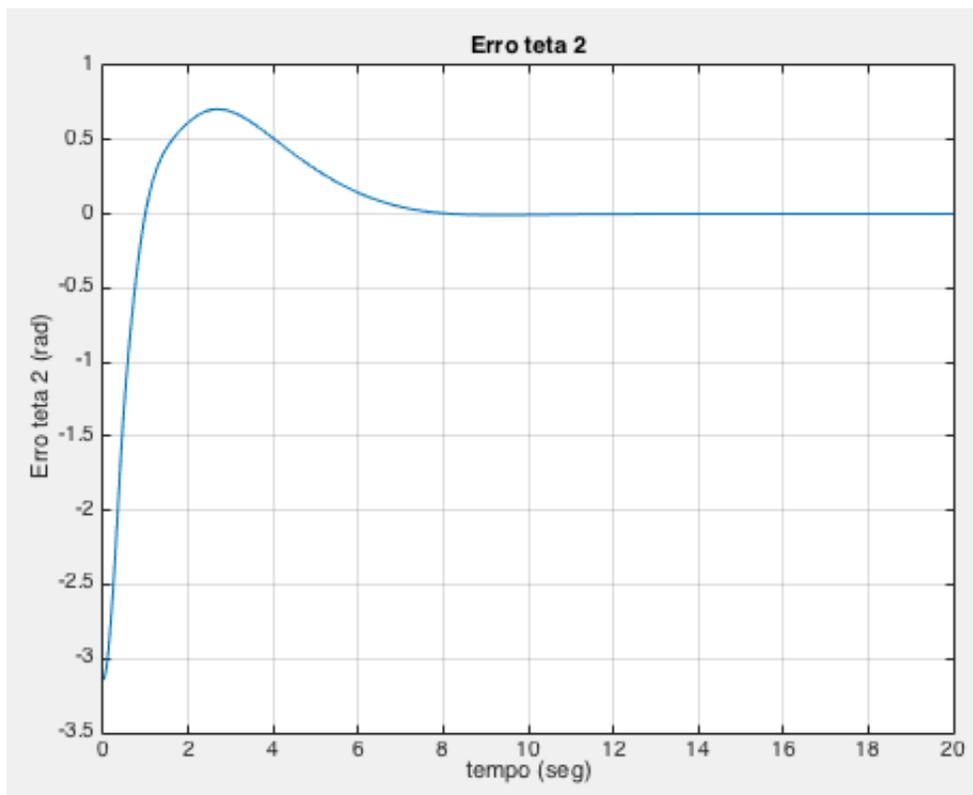
Para obter a posição desejada do manipulador, foram realizadas modelagem e controle do robô, utilizando o controlador PID para este fim. Realizando os testes com os parâmetros do controlador, pode-se observar que K_P é a relação direta com o erro e a velocidade de resposta, ou seja, aumentando este parâmetro, aumenta a velocidade de resposta, quando ao K_D é a velocidade de interação com as mudanças de estado, quanto menor, melhor a resposta em relação as mudanças de estado e K_I é a relação de cancelamento do erro, sem este parâmetro não é possível obter a posição final desejada, como já descrito, é a redução de erros do sistema controlado. Por meio desse controle, obtém-se as formas de onda dos erros das juntas θ_1 e θ_2 , indicando que a posição desejada foi alcançada.

Figura 37 – Erro da junta θ_1



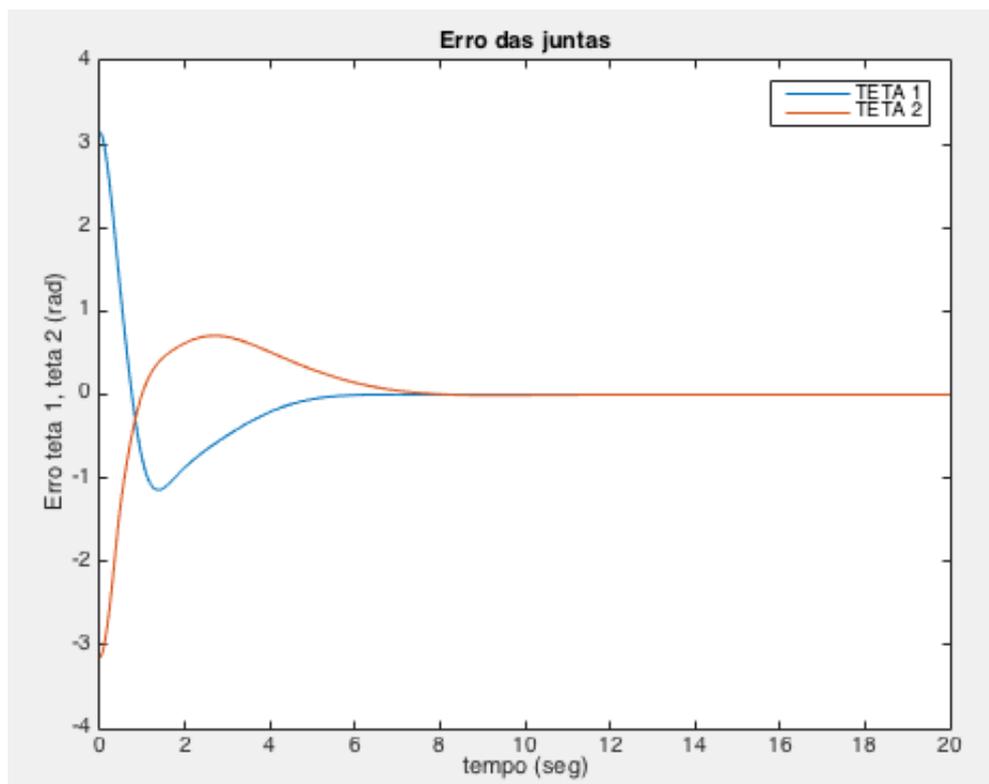
Fonte: Autora

Figura 38 – Erro da junta θ_2



Fonte: Autora

Figura 39 – Estabilidade dos elos



Fonte: Autora

Pelas formas de ondas das figuras 37 e 38 pode-se notar que:

- *Overshoot* aceitável;
- Tempo de acomodação aceitável;
- Comportamento "Linear".

E para comparação da estabilidade das juntas tem-se a figura 39.

E como resultado dos torques das juntas, onde representa a trajetória que o elo realizou até alcançar a posição desejada.

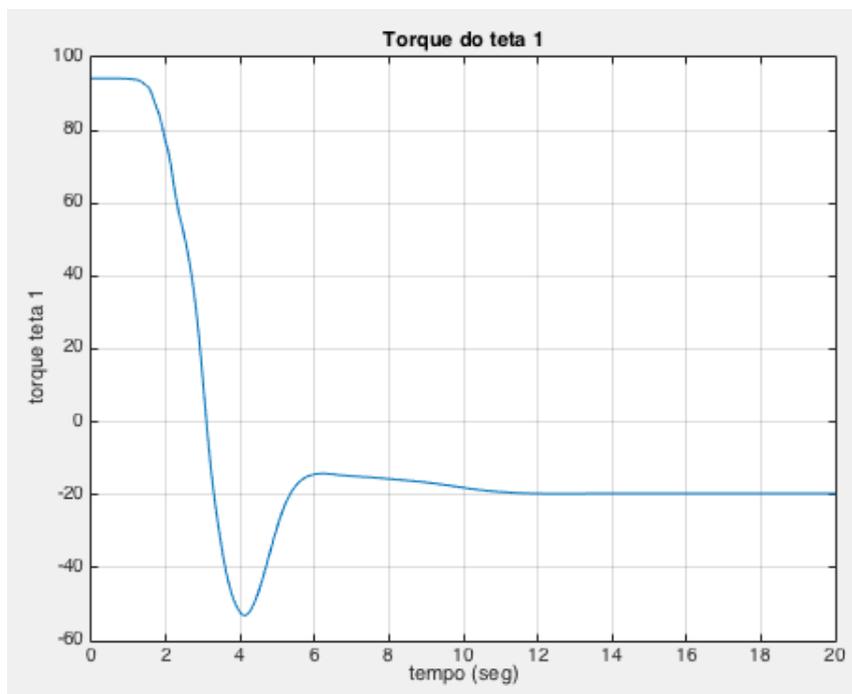
Aqui pode se analisar que os torques resultaram. As entradas de controle aqui são os binários. No entanto, colocar em mente que as articulações torques são reais

$$\begin{bmatrix} f_{\theta_1} \\ f_{\theta_2} \end{bmatrix} = B(q) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (5.1)$$

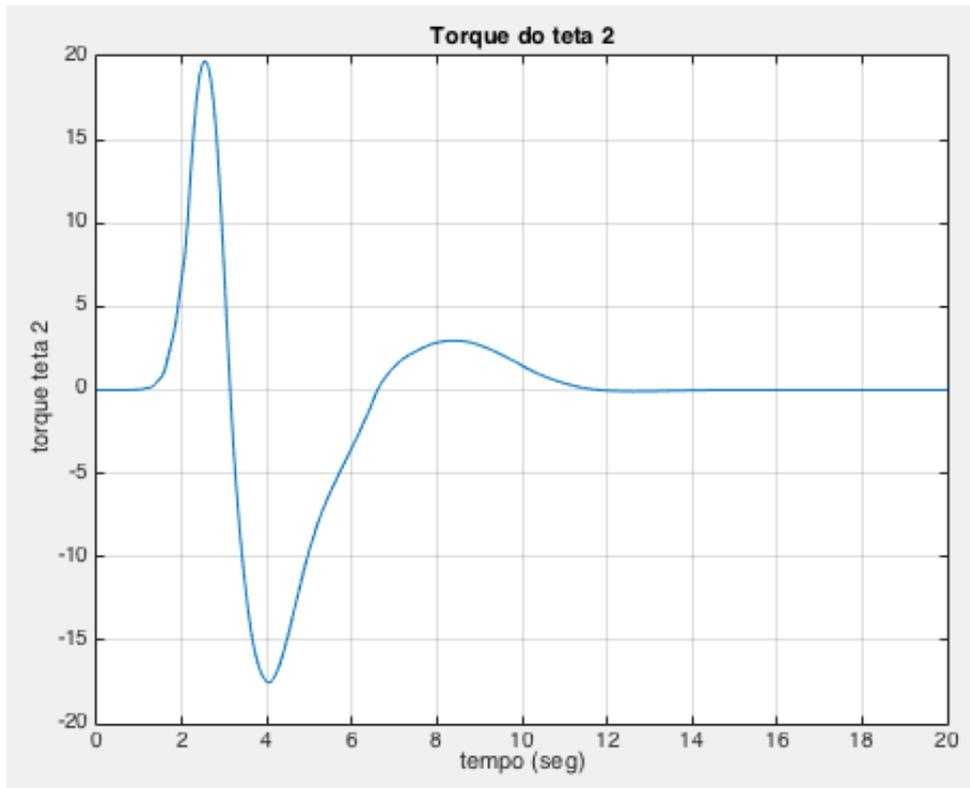
O ângulo θ_1 de alguma forma encontrar alto torque em tempo relativamente pequeno e o desempenho geral é aceitável como relativamente energia gasta é válida.

As formas de onda das articulações são binários:

Figura 40 – Torque θ_1



Fonte: Autora

Figura 41 – Torque θ_2 

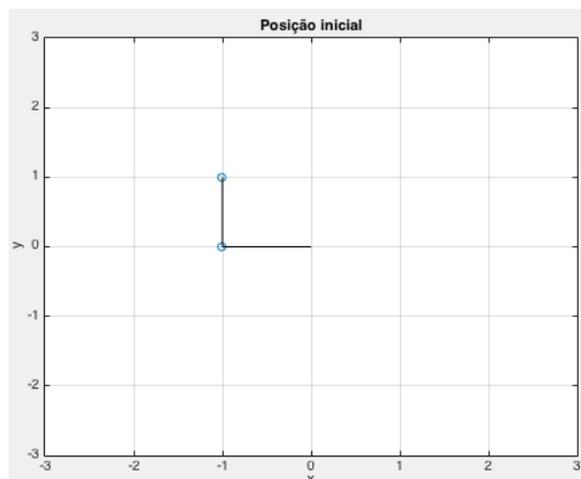
Fonte: Autora

5.2.1 Animação do movimento

Usando MATLAB®, o movimento resultante é animada no plano XY para mostrar o movimento real do robô. Um arquivo "mpg" é gerada. (Programa MATLAB consta no Anexo).

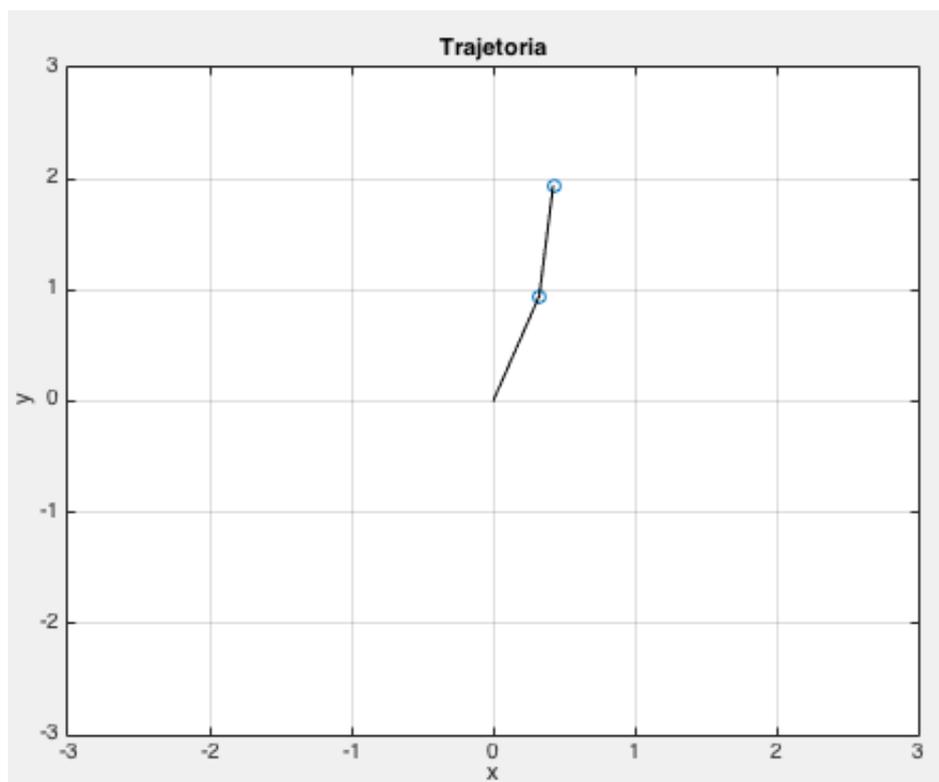
A figura 42, ilustra a posição inicial em X e Y (0,1). A trajetória do robô está exemplificada na figura 43. E por fim a posição desejada do robô na figura 44.

Figura 42 – Posição inicial



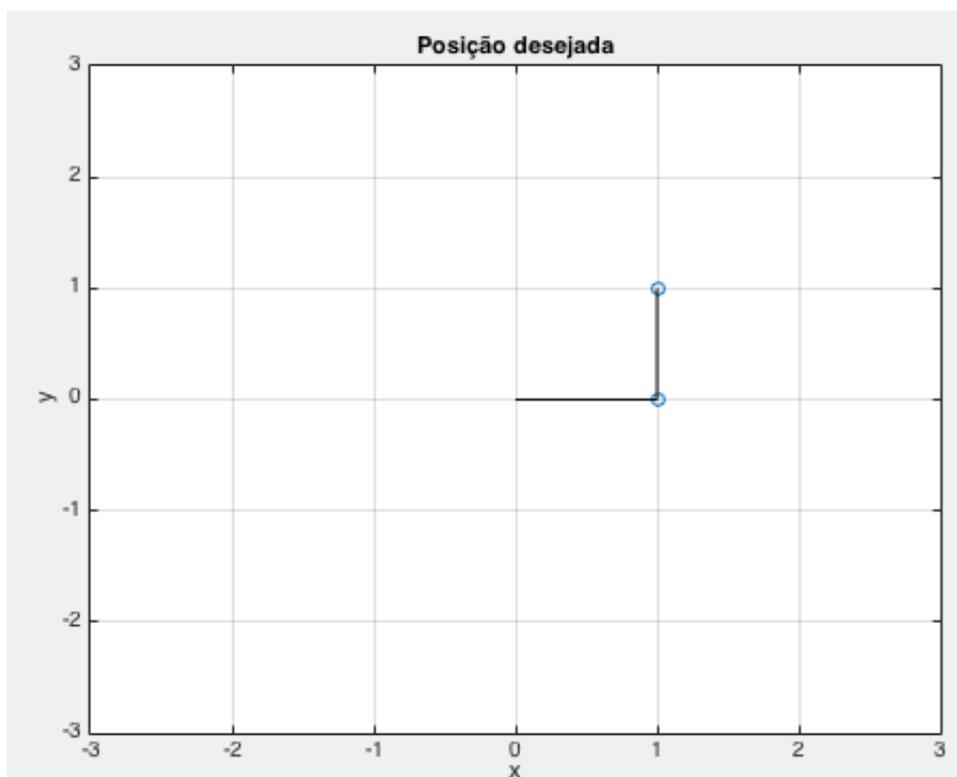
Fonte: Autora

Figura 43 – Trajetória



Fonte: Autora

Figura 44 – Posição desejada



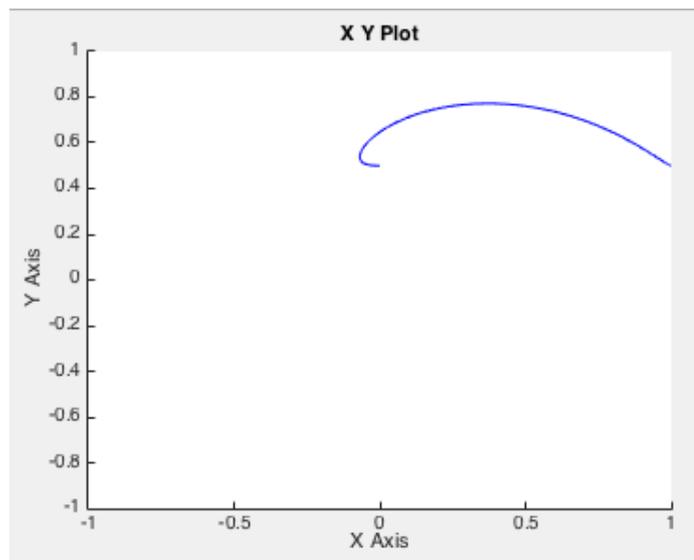
Fonte: Autora

5.3 Algoritmo completo simulação de um manipulador com dois graus de liberdade

O diagrama de blocos ilustrado na figura 46, representa um manipulador de dois graus de liberdade. O qual o usuário entra com as referências de X e Y, massas e comprimentos dos elos. É um algoritmo de simulação, utilizado para comparação do algoritmo (figura 31), que realiza a interface entre o computador e o robô. Os cálculos de cinemática, dinâmica e controle estão descrito no capítulo 4 desta monografia.

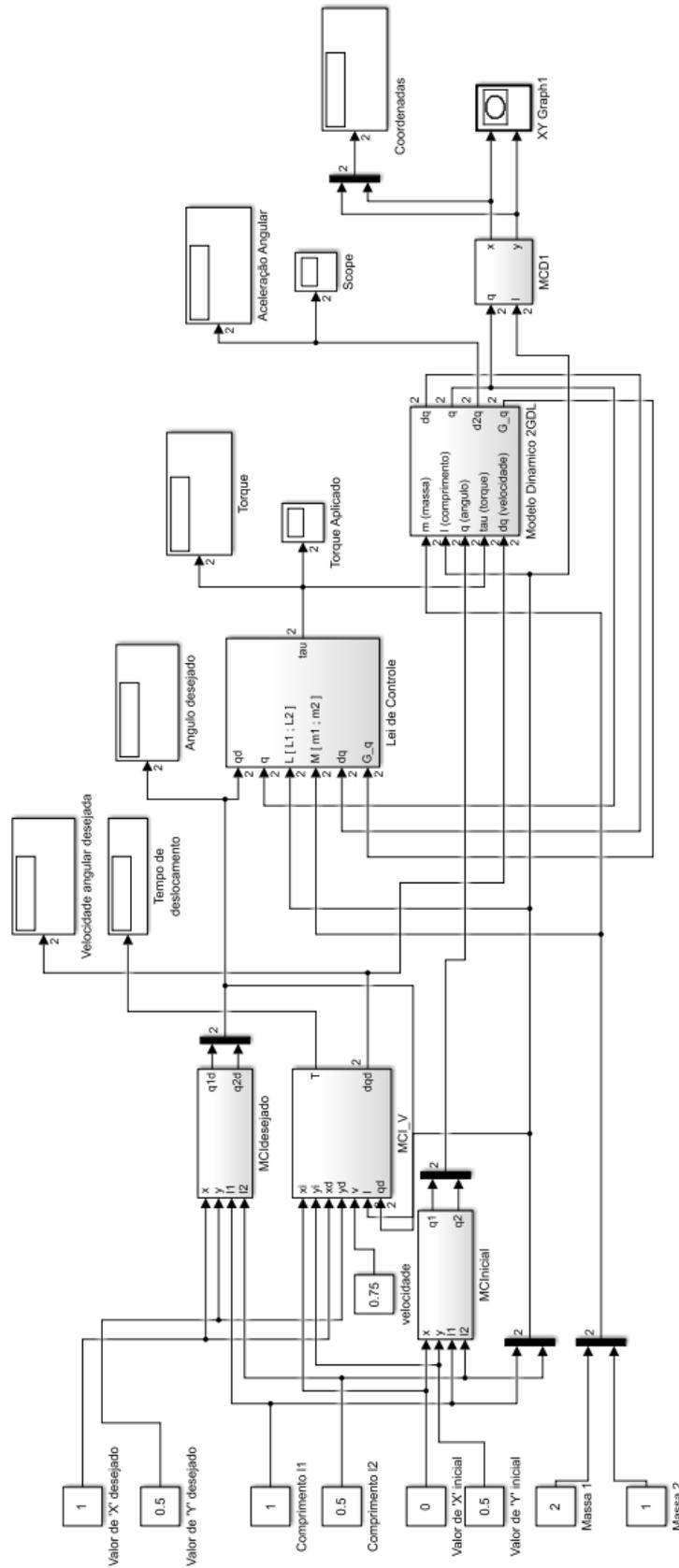
O bloco MCinicial, efetua o cálculo dos valores dos ângulos das juntas do robô, por meio da cinemática inversa. No bloco MCI_V, tem-se o cálculo jacobinano, que é a relação entre a velocidade das juntas e a velocidade do extremo. O bloco MCIdejado, define os valores de *setpoint*, massas e comprimentos dos elos, também por meio da cinemática inversa, que faz a conversão para valores angulares. Para o controle, utilizou-se o bloco Lei de controle, que contém os parâmetros do controlador, que para este algoritmo foi usado o controlador PD com compensação de gravidade, cuja a saída é o torque do manipulador. A dinâmica do sistema está representada no bloco Modelo dinâmico 2GDL, no qual constam as matrizes de inércia, coriolis e centrípedas, e torques gravitacionais. E por fim, no bloco MCD é feita a conversão dos valores das juntas, isto é a posição angular das juntas para as coordenadas cartesianas X e Y. A figura 45, representa a trajetória do movimento desejado, por meio de simulação do programa Simulink®.

Figura 45 – Trajetória do movimento



Fonte: Autora

Figura 46 – Algoritmo completo simulação de um manipulador com dois graus de liberdade



Fonte: Autora

6 CONCLUSÕES

Por meio dos resultados obtidos pode-se afirmar que é possível efetuar o controle do robô planar utilizando ferramentas computacionais, como o programa MATLAB® e o ambiente de simulação Simulink®.

Foram realizados cálculos de modelagem cinemática, que é definido a posição e orientação do robô, e modelagem dinâmica que é a relação entre posição, velocidade e aceleração com força e movimento, juntamente com o controle do robô, com a obtenção de parâmetros de posição do manipulador para definição do programa de simulação, que por sua vez, representou o controle de posição de um robô planar com dois graus de liberdade, validando a proposta da pesquisa.

Com a ligação elétrica do servomotor acoplado ao servomplificador, foi possível realizar análise das características do sistema, como velocidade, torque e controle de posição. Após o funcionamento do conjunto de servos, foi determinado os componentes auxiliares, como Circuitos Integrados (CI's) específicos, conversores de tensão-frequência, para atuação dos eixos do motor, representando os elos do robô. Definido os equipamentos para criação do *Hardware*.

Para comunicação da interface entre o computador e o robô, foi aplicado um algoritmo de blocos no *software* Simulink®, em que se obteve sucesso por meio da plataforma de comunicação Arduino®. Este algoritmo representa a interação do usuário com o sistema, onde pode ser definido os valores de posição desejada, no qual o sistema responde de forma satisfatória ao que se é solicitado.

6.1 Trabalhos Futuros

Este projeto possibilita ao corpo discente obter o controle de posição de um robô planar, por meio do programa desenvolvido na ferramenta computacional MATLAB®. Porém, o projeto é uma proposta para implementação de aplicações futuras. Portanto, para continuação do projeto, e tendo-o como base é possível realizar os seguintes trabalhos para complemento do projeto:

- Realizar a confecção de um *Hardware* melhorado:

Com a elaboração dos elos do robô, sendo a primeira junta de material mais resistente como uma liga metálica, pois este terá a função de mover o segundo elo, que por sua vez poderá ser construído com um material mais leve.

- Controle de coordenação do movimento, geração de trajetória:

Foi realizado nesta monografia o controle da posição do robô planar, o próximo passo a realizar seria o controle de movimento do manipulador, com geração da trajetória do movimento.

- Aplicação com outros métodos de controle:

E por fim, utilizar outros métodos de controle para obtenção de novos parâmetros, objetivando estudos comparativos para o método de controle mais eficaz.

REFERÊNCIAS

- ANDERSON, L. A. L. *RECONHECIMENTO E LOCALIZAÇÃO DE PESSOAS*. Tese (Doutorado) — UNIVERSIDADE CATÓLICA DE PELOTAS, 2013. 33
- BOYLESTAD, R. L.; NASHELSKY, L. *Dispositivos eletrônicos e teoria de circuitos*. [S.l.]: Pearson Prentice Hall, 2004. 51
- BRYAN, N. A. P. Desenvolvimento e desenvolvimento sustentável: metamorfoses do conceito e as suas relações com a política educacional. *Revista de Educação PUC-Campinas*, n. 20, 2012. 11
- CAMARGO, J. T. F. de; VERASZTO, E. V.; BARRETO, G. Simulação do modelo cinemático inverso de um robô através do uso de redes neurais artificiais: Um complemento ao ensino de robótica. 2003. 15
- CAMPBELL, C. et al. Desenvolvimento de um robô manipulador industrial. 2007. 14
- CRAIG, J. J. *Robotica*. [S.l.: s.n.], 2012. 16, 17, 18
- FAIRCHILD. *CD4049UBC Hex Inverting Buffer Hex Non-Inverting Buffer*. [S.l.], 2002. 51
- HERMINI, H. A. Modelagem cinemática direta e inversa de robôs. 2004. Disponível em: <<http://www.fem.unicamp.br/~hermini/Robotica/aulas/Notas%20da%20Aula%203.doc>> 23, 24
- KURNIAWAN, A. Getting started with matlab simulink and arduino. 2013. 54
- MALAQUIAS, F. D.; ASSOLARI, K.; NOGAWA, M. L. Desenvolvimento de um sistema de controle de um braço robótico de cinco eixos. Curitiba, 2013. 24, 25, 27
- MATOS, N. M. R. d. et al. Análise do funcionamento de um servomotor de corrente alternada com ímãs permanentes. 2014. 32
- MITISUBISHI. *SERIE SUPER SERVO DRIVE MR-J2-A*. [S.l.], 2003. 13, 45, 52, 55
- MOLINA, R. et al. Introdução à dinâmica e ao controle de manipuladores robóticos. *curso de engenharia de controle e automação da PUC-RS*, 2005. 29, 31
- NIKU, S. B. *Introduction to robotics: analysis, systems, applications*. [S.l.]: Prentice Hall New Jersey, 2001. 42
- NOY, N. F. et al. Biportal: ontologies and integrated data resources at the click of a mouse. *Nucleic acids research*, Oxford Univ Press, p. gkp440, 2009. 56
- ROSSATO, D. B. *Desenvolvimento de um sistema aberto para ensino de robôs manipuladores*. Tese (Doutorado) — Universidade de São Paulo, 2009. 19, 20
- SÁ, F. L. D. et al. Universidade do estado de santa catarina–udesc centro de ciências tecnologicas–cct departamento de engenharia eletrica–dee programa de pós-graduação em engenharia elétrica–ppgeel. 2013. 32

SANTOS, V. M. Robótica industrial. *Departamento de Engenharia Mecânica*, 2004. 18, 28, 29

SCHERPEN, J. M.; ORTEGA, R. On nonlinear control of euler-lagrange systems: Disturbance attenuation properties. *Systems & control letters*, Elsevier, v. 30, n. 1, p. 49–56, 1997. 33

SIQUEIRA, A. A.; TERRA, M. H. Nonlinear and markovian h^∞ controls of underactuated manipulators. *Control Systems Technology, IEEE Transactions on*, IEEE, v. 12, n. 6, p. 811–826, 2004. 31

SUDARMA, M.; SWAMARDIKA, I.; PRANATA, I. D. Design and implementations of control system quadraped robot driver application based on windows platform. *International Journal of Electrical and Computer Engineering (IJECE)*, v. 5, n. 2, p. 251–258, 2015. 54

TOMASCHITZ, F. R. S. Controle e análise da velocidade de servos-motores de manipuladores cartesianos em uma linha automatizada de estamperia. Curitiba, 2014. 33

ANEXO A – ALGORITMO DE CONTROLE IMPLEMENTADO NO MATLAB®

1) Função ODE do sistema

```
function xdot=r2dof(t,x,ths,spec,Kpid)

xdot=zeros(8,1);
%% set-points
th1s=ths(1);
th2s=ths(2);

%% Especificações do robô
M1=spec(3);
M2=spec(4);
L1=spec(1);
L2=spec(2);
g=9.8;

%% Matriz de inércia
b11=(M1+M2)*L1^2+M2*L2^2+2*M2*L1*L2*cos(x(4));
b12=M2*L2^2+M2*L1*L2*cos(x(4));
b21=M2*L2^2+M2*L1*L2*cos(x(4));
b22=M2*L2^2; Bq=[b11 b12;b21 b22];

%% Matriz C
c1=-M2*L1*L2*sin(x(4))*(2*x(5)*x(6)+x(6)^2);
c2=-M2*L1*L2*sin(x(4))*x(5)*x(6);
Cq=[c1;c2];

%% Matriz Gravitacional
g1=-(M1+M2)*g*L1*sin(x(3))-M2*g*L2*sin(x(3)+x(4));
g2=-M2*g*L2*sin(x(3)+x(4));
Gq=[g1;g2];

%% Controle PID
% Parâmetros PID para primeira junta (teta 1)
```

```

Kp1=Kpid(1);
Kd1=Kpid(2);
Ki1=Kpid(3);

% Parâmetros PID para segunda junta (teta 2)
Kp2=Kpid(4);
Kd2=Kpid(5);
Ki2=Kpid(6);

% Entrada de controle desacoplado
f1=Kp1*(th1s-x(3))-Kd1*x(5)+Ki1*(x(1));
f2=Kp2*(th2s-x(4))-Kd2*x(6)+Ki2*(x(2));
Fhat=[f1;f2];

F=Bq*Fhat; % Entrada real do sistema

%% Estados do sistema
xdot(1)=(th1s-x(3)); %Estado simulado de integração de teta 1
xdot(2)=(th2s-x(4)); %Estado simulado de integração de teta 2

xdot(3)=x(5); %teta1-dot
xdot(4)=x(6); %teta2-dot

q2dot=inv(Bq)*(-Cq-Gq+F);

xdot(5)=q2dot(1); %teta1-2dot
xdot(6)=q2dot(2); %teta1-2dot

%Controle de entrada para função de saída do computador externo
xdot(7)=F(1);
xdot(8)=F(2);

end

2) Solução e Simulação do controle do sistema

close all
clear all
clc

```

```
%% Inicialização
th_int=[-pi/2 pi/2];

%Posição inicial
ths=[pi/2 -pi/2];

%set-points
x0=[0 0 th_int 0 0 0 0]; %Valores do estado inicial
Ts=[0 20]; %time span

%% Especificações do robô
L1=1; %elo 1
L2=1; %elo 2
M1=1; %massa 1
M2=1; %massa 2
spec=[L1 L2 M1 M2];

%% PID Parameters
% Parâmetros PID para primeira junta (teta 1)
Kp1=15;
Kd1=7;
Ki1=10;
% Parâmetros PID para primeira junta (teta 2)
Kp2=15;
Kd2=10;
Ki2=10;
Kpid=[Kp1 Kd1 Ki1 Kp2 Kd2 Ki2];

%% Solução ODE
% opt1=odeset('RelTol',1e-10,'AbsTol',1e-20,'NormControl','off');
[T,X] = ode45(@(t,x) r2dof(t,x,ths,spec,Kpid),Ts,x0);

%% Saída
th1=X(:,3);%Forma de onda teta 1
th2=X(:,4); %Forma de onda teta 2

%Cálculo dos torques dos 7º, 8º estados da ODE
F1=diff(X(:,7))./diff(T);
F2=diff(X(:,8))./diff(T);
```

```

tt=0:(T(end)/(length(F1)-1)):T(end);

%xy
x1=L1.*sin(th1); % X1
y1=L1.*cos(th1); % Y1
x2=L1.*sin(th1)+L2.*sin(th1+th2); % X2
y2=L1.*cos(th1)+L2.*cos(th1+th2); % Y2

%Erro teta 1 gráfico
(T,ths(1)-th1)
grid
title('Teta-1 erro')
ylabel('teta1 erro (rad)')
xlabel('tempo (seg)')
%teta2 erro gráfico
figure
plot(T,ths(2)-th2)
grid
title('Teta-2 erro')
ylabel('teta2 erro (rad)')
xlabel('tempo (seg)')
%torque 1 gráfico
figure
plot(tt,F1)
grid
title('Torque teta 1')
ylabel('teta 1 torque')
xlabel('tempo (seg)')
%torque2 plot figure plot(tt,F2)
grid
title('Torque of theta 2')
ylabel('teta 2 torque')
xlabel('tempo (seg)')

```

3) Animação do movimento do Robô

```

%% Configuração de velocidade
d=2;
j=1:d:length(T);

```

```
%% gerando as imagens 2D
figure
for i=1:length(j)-1
hold off
plot([x1(j(i)) x2(j(i))],[y1(j(i)) y2(j(i))],'o',[0 x1(j(i))],[0
y1(j(i))],'k',[x1(j(i)) x2(j(i))],[y1(j(i)) y2(j(i))],'k')
title('Movimento do robô de 2GDL')
xlabel('x')
ylabel('y')
axis([-3 3 -3 3]);
grid
hold on
MM(i)=getframe(gcf);
end
drawnow;

%% exportar para filme 'mpg'
mpgwrite(MM,'RGB','2DOF_rob.mpg')
```