

PROJETO E IMPLEMENTAÇÃO DE UMA JUNTA ROBÓTICA EM CONFIGURAÇÃO ANTAGÔNICA COM CONTROLE DE IMPEDÂNCIA

Oscar Henrique Reichow, Rodrigo Trentini, Joel Stryhalski

Instituto Federal de Santa Catarina

Câmpus Jaraguá do Sul – Rau – Curso de Bacharelado em Engenharia Elétrica

e-mail: oscar.r18@aluno.ifsc.edu.br, rodrigo.trentini@ifsc.edu.br, joel@ifsc.edu.br

Trabalho de Conclusão de Curso – 01/07/2022

Resumo – Este trabalho tem como objetivo a construção e projeto de uma junta robótica, que possua a capacidade de variar sua impedância mecânica. O controle será realizado através de um sistema de acionamento antagonístico. Para isso a dinâmica da junta é modelada através da Equação de Lagrange, bem como realiza-se o projeto dos controladores através do método de Lugar Geométrico das Raízes. Atualmente a maioria dos robôs não trabalha no mesmo local que seres humanos, devido ao considerável risco de acidentes graves. Para o acionamento de robôs que operam em ambientes pouco estruturados, exoesqueletos e próteses, a técnica de alterar a impedância de juntas do robô, é utilizada para que estes tenham a capacidade de atuar e minimizar os danos em caso de colisões. Para possibilitar a alteração de impedância, é utilizado um elemento elástico com relação exponencial entre força e deformação, denominado SAT. Os resultados obtidos mostram que é possível controlar a impedância de uma junta rotativa em configuração antagonística atuada por motores.

Palavras-chave – Junta robótica, Controle de impedância, SAT.

DESIGN AND IMPLEMENTATION OF A ROBOTIC JOINT IN ANTAGONICAL CONFIGURATION WITH IMPEDANCE CONTROL

Abstract – This work aims at the construction and design of a robotic joint, which has the ability to vary its mechanical impedance. The control will be carried out through an antagonistic drive system. For this, the joint dynamics is modeled through the Lagrange Equation, as well as the design of the controllers through the method of Geometric Place of Roots. Currently, most robots do not work in the same location as humans, due to the considerable risk of serious accidents. For the activation of robots that operate in poorly structured environments, exoskeletons and prostheses, the technique of changing the impedance of the robot's joints is used so that they have the ability to act and minimize damage in case of collisions. To enable the impedance change, an elastic element with an exponential

relationship between force and deformation, called SAT, is used. The results obtained show that it is possible to control the impedance of a rotating joint in an antagonistic configuration actuated by motors.

Keywords – Robotic joint, Impedance control, SAT.

I. INTRODUÇÃO

Os robôs são partes fundamentais de diversos processos produtivos, e estão cada vez mais fazendo parte do dia-a-dia das pessoas, das indústrias, sobretudo na indústria 4.0. Realizando atividades cada vez menos dedicadas ou específicas, o leque de utilizações da mesma configuração física de um robô é ampliado, o que otimiza e potencializa suas aplicações, sobretudo com o advento da Indústria 4.0 [1].

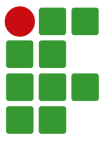
Atualmente, a maioria dos robôs não trabalha no mesmo local que seres humanos, por causa do risco que estas máquinas oferecem, devido a sua baixa sensibilidade e percepção háptica. Desta forma, permitir a coexistência robô-humano requer um projeto cuidadoso e controle de baixo nível [2].

Para o acionamento de robôs que operam em ambientes pouco estruturados, exoesqueletos e próteses, garantindo a segurança de trabalho do robô e de pessoas, a abordagem de controle mais amplamente utilizada é a de controle de impedância mecânica. O controle de impedância é introduzido por Hogan [3] e estendido para robôs articulados e flexíveis por Ott [4].

A impedância mecânica é uma medida de quanto uma estrutura resiste ao movimento quando submetida a uma força harmônica, sendo a razão entre a força aplicada em um ponto e a velocidade resultante naquele ponto [5], [6].

Baseado no controle de força/torque da junta, o controle de impedância impõe um comportamento físico desejado nas articulações do robô. Esse ramo de pesquisa é conhecido como “paradigma da robótica colaborativa” [7].

Mecanismos robóticos que utilizam a técnica de alterar a sua impedância são denominados de “Atuadores de Impedância Variável”. Essa técnica é implementada em diferentes protótipos robóticos, com um amplo espectro de aplicações. Dentre essas aplicações, destacam-se as que envolvem robôs para entretenimento, principalmente para crianças, com o objetivo de terem um toque macio e confortável [8], [9].



Outras aplicações de destaque são próteses robóticas para caminhar, pular e correr [10]–[15], e também as chamadas próteses ativas, as quais tem o propósito de proporcionar uma caminhada mais eficiente e natural. Em sistemas como as mãos e braços robóticos, a impedância da junta pode ser alterada mecanicamente, ou por um controlador [16]–[18], ou pela combinação dos dois [19].

Atuadores de impedância variável são dispositivos mecatrônicos desenvolvidos para construir robôs passivos colaborativos, robustos e hábeis. A escolha de um dos vários tipos de mecanismos, depende de qual aplicação este será empregado.

Neste trabalho utilizou-se uma junta em configuração antagonônica, cujo princípio de atuação do tendão antagonônico provindo da biologia, pode ser aplicado a robôs por meio de tendões que conectam as juntas com os atuadores, permitindo uma realocação dos acionamentos para a base do sistema do robô. Além disso, a arquitetura desse sistema reduz a inércia, levando a um movimento mais rápido, preservando a segurança do ser humano e do robô [20].

A introdução de elementos elásticos no trajeto do tendão melhora a limitação de sistemas rígidos, onde forças no tendão excessivamente grandes levam à sua ruptura [20].

Uma impedância variável permite adaptar a elasticidade da junta a uma dada tarefa de manipulação, seja para agarrar um objeto pesado para evitar danos ao robô ou para uma montagem rápida de peças. Outra característica é que os elementos elásticos conectados em série atuam como armazenadores de energia, que pode ser aplicada a uma tarefa de arremesso por meio da transferência da energia potencial elástica da mola em energia cinética e vice-versa [21], [22].

Embora as molas lineares sejam mais fáceis de trabalhar e estejam prontamente disponíveis, elas não permitem que a impedância da junta varie [23], mesmo na configuração antagonônica. Isso justifica o desenvolvimento de uma mola não linear conhecida como SAT (*Stiffness Adjustable Tendon*) [24], [25] que tem algumas vantagens, como estrutura leve e simples.

Para implementar o algoritmo de rastreamento em robôs atuados antagonicamente, várias abordagens de controle foram propostas nos últimos anos [26]–[28].

Muitos controladores têm sido propostos para mecanismos acionados por tendões, como por exemplo, controladores usando realimentação de tensão [29] e linearização por realimentação [30]. Shimamoto [31] propôs um controle bilateral para robôs multigrados de liberdade com mecanismos leves para colocar o sistema em uso prático.

Vanderborght [32] apresentou uma visão geral dos diferentes atuadores de impedância variável. Em [33] e [34], enfocase a implementação de uma junta de rigidez variável acionada por um par de atuadores de corda torcida em configuração antagonônica. Wimböck [20] estabilizou assintoticamente um sistema controlado por tendão com rigidez variável usando o teorema de imersão e invariância (I&I).

Desta forma, este trabalho tem como objetivo geral controlar a impedância mecânica de uma junta simples em configuração antagonônica. Os objetivos específicos são:

- Modelar uma junta em configuração antagonônica;
- Projetar o controlador;
- Construir um protótipo;
- Analisar e validar os resultados obtidos.

Para que os objetivos sejam alcançados, o trabalho está dividido em cinco seções. Na Seção II, é explicado o funcionamento da mola utilizada, além de ser realizada a modelagem da junta em configuração antagonônica, e explicado como é feito o controle. A Seção III apresenta informações a respeito da montagem do protótipo, bem como o desenvolvimento dos controladores. Na Seção IV os resultados obtidos são analisados e validados. Para finalizar, na Seção V, têm-se as conclusões obtidas neste trabalho.

II. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, primeiro a mola não linear SAT é explicada. Após, é realizada a modelagem matemática da junta em configuração antagonônica, e por fim, é explicado o método de projeto de controladores utilizado, e a discretização do controlador.

A. Mola não linear SAT

Os elementos de transmissão são compostos por elementos elásticos. De acordo com a lei de Hooke, Kobayashi e Ozawa [35] mostram que a força de tração causada por molas é proporcional (linear) com a deformação da mola (ϵ), sendo $F(\epsilon) = K_t \epsilon$ onde K_t é uma matriz de rigidez. Porém, molas lineares não são suficientes para criar uma variação de rigidez, pois a rigidez articular resultante é nula neste caso. De acordo com algumas pesquisas e análises, a geometria e a característica do material fornecem a relação exponencial entre força e deformação [36]:

$$F(\epsilon) = k(e^{\sigma\epsilon} - 1), \quad (1)$$

sendo $F(\epsilon)$ a força exercida pela mola em Newtons, ϵ é a deformação em metros, k é a matriz de rigidez em Nm/rad , e σ é uma constante positiva que depende da forma e material usados nos elementos de transmissão em $1/rad$.

O elemento elástico não linear utilizado neste trabalho, denominado de SAT (*Stiffness Adjustable Tendon*) pode ser visto na Fig. 1. Montado com uma esponja polimérica de silicone como corpo elástico, e inserida dentro de um tubo de malha feita de poliuretano, mais comumente conhecido como por malha náutica, o SAT tem algumas vantagens como por exemplo o seu peso, estrutura simples, baixo custo e muito resistente à ação do tempo [24], [25].

SAT é uma estrutura do tipo cilíndrica na qual uma barra



Fig. 1. Estrutura do SAT.

redonda de esponja polimérica de silicone (denominada corpo elástico) é inserida como um corpo elástico em um tubo de malha que serve como um revestimento. O diâmetro interno do tubo de malha muda de acordo com a mudança do comprimento total do tubo. Quando uma tensão é aplicada em ambas as extremidades do SAT, o tubo de malha se deforma longitudinalmente, e ao mesmo tempo o diâmetro interno se deforma transversalmente, ou seja, o corpo elástico no interior é comprimido. Quando a tensão nas extremidades reduz, o corpo elástico se expande na tentativa de retornar à sua forma original, conseqüentemente, o tubo de malha também retorna ao estado inicial. O SAT se estende até um comprimento em que a força compressiva e a força restauradora são equilibradas.

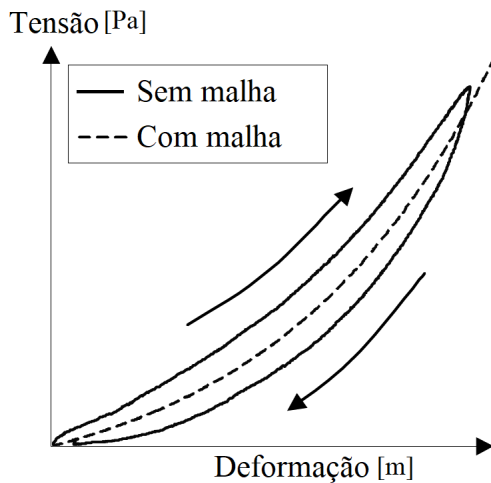


Fig. 2. Característica do SAT.

A Figura 2 mostra o resultado da deformação do SAT para

quando uma tensão é aplicada. O eixo horizontal é a deformação do corpo e o eixo vertical é a tensão aplicada. Pode-se observar que a relação entre tensão e deformação não é linear.

Verifica-se que existem características de histerese diferenciadas entre o processo de aumento da tensão e o processo de devolvê-la. A Figura 2 mostra os resultados da relação entre a tensão e a deformação de um corpo elástico sem cobrir a superfície com um tubo de malha. Pode-se observar que as características não lineares não aparecem apenas com o corpo elástico, e que a geração das características de histerese se deve principalmente às características do corpo elástico.

B. Modelo dinâmico de uma junta em configuração antagonônica

Nesta seção, uma junta simples em configuração antagonônica acionada por tendões, consistindo de uma junta e dois atuadores, conectados por acoplamento elástico é apresentada. Para obter o modelo dinâmico, fazem-se as seguintes considerações:

- A massa da junta é desprezível, logo, as forças relativas à gravidade, centrípeta e de Coriolis são desprezadas;
- A junta é acionada de forma independente por 2 motores CC em configuração antagonônica;
- Para possibilitar a alteração de impedância da junta, os tendões considerados são elementos de transmissão com relação exponencial entre deformação e força (SAT);
- A função da força de tração $F(\epsilon)$ é igual a 0 para $\epsilon < 0$, onde ϵ é a deformação do tendão. Caso $\epsilon > 0$, $F(\epsilon) \geq 0$ e conseqüentemente $\frac{dF(\epsilon)}{d\epsilon} > 0$;
- Os coeficientes de atrito desconhecidos da junta e dos atuadores são desprezados;
- A modelagem dos motores CC não é realizada.

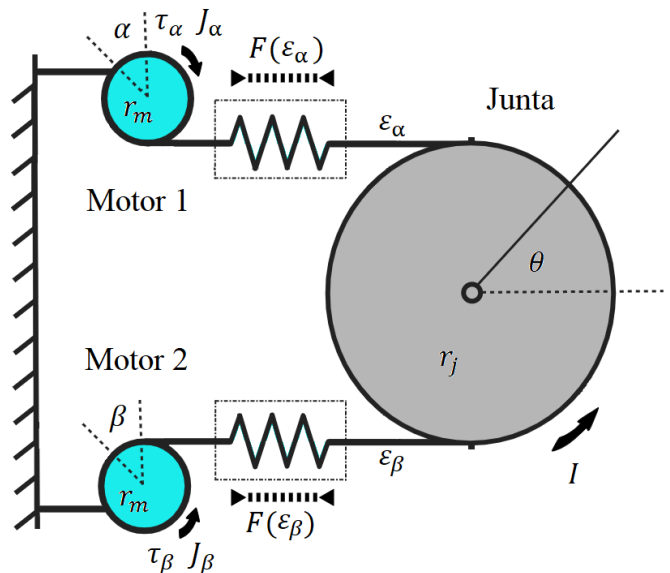


Fig. 3. Junta simples em configuração antagonônica com tendões acionados por motores.



Uma representação esquemática da junta antagônica com tendões controlados é apresentada na Fig. 3. Por meio da Equação de Lagrange, pode-se obter o modelo matemático dinâmico da junta. Para tal, é necessário encontrar a energia cinética e potencial do sistema. Para a energia cinética tem-se [37]:

$$T = \frac{1}{2}I\dot{\theta}^2 + \frac{1}{2}J_{\alpha}\dot{\alpha}^2 + \frac{1}{2}J_{\beta}\dot{\beta}^2, \quad (2)$$

onde θ é a posição angular da junta, α e β são respectivamente as posições angulares dos motores CC agonista e antagonista, também chamados de atuadores, I é o momento de inércia do conjunto conectado pela junta, J_{α} e J_{β} são momentos de inércia dos atuadores.

Os elementos de transmissão são conectados aos atuadores por cabos metálicos. Para evitar folga no cabo, uma região de operação adequada de α , β e θ deve ser definida considerando a geometria do dispositivo.

A deformação resultante entre a junta e seus atuadores é definido como:

$$\varepsilon_{\alpha} = r_m\alpha + r_j\theta, \quad (3)$$

$$\varepsilon_{\beta} = r_m\beta - r_j\theta. \quad (4)$$

onde r_m e r_j são o raio da polia dos atuadores e da junta, respectivamente. Para simplificar os cálculos e diminuir perdas, $r_m = r_j = r$. Assim:

$$\varepsilon_{\alpha} = r(\alpha - \theta), \quad (5)$$

$$\varepsilon_{\beta} = r(\beta + \theta). \quad (6)$$

Sabe-se que energia potencial é igual a [37]:

$$V = \int \vec{F} \cdot d\vec{\varepsilon}. \quad (7)$$

Para a energia potencial do sistema temos:

$$V = k \left(\frac{e^{r\sigma(\beta+\theta)}}{\sigma} - r(\beta + \theta) \right) + k \left(\frac{e^{r\sigma(\alpha-\theta)}}{\sigma} - r(\alpha - \theta) \right). \quad (8)$$

O Lagrangiano do sistema é escrito como [37]:

$$L = T - V, \quad (9)$$

logo:

$$L = \frac{1}{2}I\dot{\theta}^2 + \frac{1}{2}J_{\alpha}\dot{\alpha}^2 + \frac{1}{2}J_{\beta}\dot{\beta}^2 - k \left(\frac{e^{r\sigma(\beta+\theta)}}{\sigma} - r(\beta + \theta) \right) - k \left(\frac{e^{r\sigma(\alpha-\theta)}}{\sigma} - r(\alpha - \theta) \right). \quad (10)$$

Com a Eq.10, chega-se nas equações diferenciais de Lagrange [37]:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} = \frac{\partial L}{\partial \theta} \rightarrow I\ddot{\theta} = -k(r - re^{r\sigma(\alpha-\theta)}) - k(re^{r\sigma(\beta+\theta)} - r), \quad (11)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\alpha}} = \frac{\partial L}{\partial \alpha} \rightarrow J_{\alpha}\ddot{\alpha} = -k(re^{r\sigma(\alpha-\theta)} - r), \quad (12)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\beta}} = \frac{\partial L}{\partial \beta} \rightarrow J_{\beta}\ddot{\beta} = -k(re^{r\sigma(\beta+\theta)} - r). \quad (13)$$

Desta forma, acrescentando os torques atuadores τ_{α} e τ_{β} e o torque externo τ_{ext} como entradas, e os estados do sistema sendo:

$$\theta = x_1, \quad (14)$$

$$\dot{\theta} = x_2, \quad (15)$$

$$\alpha = x_3, \quad (16)$$

$$\dot{\alpha} = x_4, \quad (17)$$

$$\beta = x_5, \quad (18)$$

$$\dot{\beta} = x_6, \quad (19)$$

o sistema por representação em espaço de estados é:

$$\dot{x}_1 = x_2, \quad (20)$$

$$\dot{x}_2 = -\frac{kr}{I}(e^{r\sigma(x_5+x_1)} - e^{r\sigma(x_3-x_1)}) - \frac{\tau_{ext}}{I}, \quad (21)$$

$$\dot{x}_3 = x_4, \quad (22)$$

$$\dot{x}_4 = -\frac{kr}{J_{\alpha}}(e^{r\sigma(x_3-x_1)} - 1) + \frac{\tau_{M1}}{J_{\alpha}}, \quad (23)$$

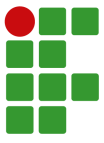
$$\dot{x}_5 = x_6, \quad (24)$$

$$\dot{x}_6 = -\frac{kr}{J_{\beta}}(e^{r\sigma(x_5+x_1)} - 1) + \frac{\tau_{M1}}{J_{\beta}}. \quad (25)$$

A impedância resultante da junta pode ser expressa como:

$$S = \frac{\partial \tau}{\partial \theta}. \quad (26)$$

Sabendo que $I\ddot{\theta} = \tau$, tem-se que a impedância da junta é:



$$S = -kr^2\sigma(e^{r\sigma(\alpha-\theta)} + e^{r\sigma(\beta+\theta)}). \quad (27)$$

Devido a relação exponencial entre compressões e as forças dos elementos de transmissão, os valores de impedância da junta sempre são negativos. Particularmente, os valores dependem dos parâmetros k e σ . O valor de impedância inicial da junta S_0 , que corresponde a condição $\varepsilon_\alpha = \varepsilon_\beta = 0$ é obtido por $S_0 = -2k\sigma r^2$.

C. Controle de uma junta em configuração antagônica

O passo inicial para projetar o controlador, é linearizar o sistema. Através do método de linearização por Série de Taylor, o sistema por espaço de estados fica da seguinte forma:

$$\dot{x}_1 = x_2, \quad (28)$$

$$\begin{aligned} \dot{x}_2 = & -\frac{kr}{I}[r\sigma(e^{r\sigma(\bar{x}_5+\bar{x}_1)} + e^{r\sigma(\bar{x}_3-\bar{x}_1)})]\Delta x_1 + \frac{kr^2\sigma}{I}e^{r\sigma(\bar{x}_3-\bar{x}_1)}\Delta x_3 \\ & - \frac{kr^2\sigma}{I}e^{r\sigma(\bar{x}_5+\bar{x}_1)}\Delta x_5 - \frac{1}{I}\Delta\tau_{ext}, \end{aligned} \quad (29)$$

$$\dot{x}_3 = x_4, \quad (30)$$

$$\dot{x}_4 = \frac{kr^2\sigma}{J_\alpha}e^{r\sigma(\bar{x}_3-\bar{x}_1)}\Delta x_1 - \frac{kr^2\sigma}{J_\alpha}e^{r\sigma(\bar{x}_3-\bar{x}_1)}\Delta x_3 + \frac{1}{J_\alpha}\Delta\tau_{M1}, \quad (31)$$

$$\dot{x}_5 = x_6, \quad (32)$$

$$\dot{x}_6 = -\frac{kr^2\sigma}{J_\beta}e^{r\sigma(\bar{x}_5+\bar{x}_1)}\Delta x_1 - \frac{kr^2\sigma}{J_\beta}e^{r\sigma(\bar{x}_5+\bar{x}_1)}\Delta x_5 + \frac{1}{J_\beta}\Delta\tau_{M2}. \quad (33)$$

sendo $\bar{x}_1 = 0$, $\bar{x}_3 = 0$ e $\bar{x}_5 = 0$.

Com o sistema linearizado, as matrizes de estados e saída são:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{(a+b)}{I} & 0 & \frac{a}{I} & 0 & -\frac{b}{I} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{a}{J_\alpha} & 0 & -\frac{a}{J_\alpha} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{b}{J_\beta} & 0 & 0 & 0 & -\frac{b}{J_\beta} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{I} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{J_\alpha} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J_\beta} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D = 0$$

sendo $a = kr^2\sigma e^{r\sigma(\bar{x}_3-\bar{x}_1)}$ e $b = kr^2\sigma e^{r\sigma(\bar{x}_5+\bar{x}_1)}$.

Em um sistema na forma do espaço dos estados, podem existir dinâmicas que não são vistas pelas saídas do sistema ou não são influenciadas pelas entradas. Olhando para uma função de transferência, entende-se que um cancelamento de um pólo com um zero, implica que alguma dinâmica no sistema deixa de ser vista pela saída e nem pode ser alterada pela entrada, o que gera perda de controlabilidade.

Desta forma, com o sistema linear, é possível obter-se as equações de transferência no domínio da frequência, através de:

$$\frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D. \quad (34)$$

Com as equações de transferência do sistema, optou-se por projetar o controle através do método de Lugar Geométrico das Raízes (LGR).

O método LGR permite que seja analisada a posição dos polos da malha direta da planta, e possibilita que seja escolhida a posição dos polos de malha fechada de acordo com a resposta desejada.

D. Controle Discreto

Para a implementação do controle em um sistema com componentes digitais, como por exemplo um microcontrolador, se faz necessário o uso de técnicas de controle digital. Imagine que, apesar de existir de maneira contínua - ou seja, existe um valor a qualquer instante de tempo -, a variável de saída só está disponível em instantes regularmente espaçados. Por exemplo, ao se monitorar a posição de um avião por um radar, a informação é atualizada a cada varredura e não de maneira contínua [38].

Para gerar um sinal discreto no tempo, a técnica utilizada é a de amostragem de um variável que existe continuamente ao longo do tempo. O intervalo de tempo entre uma medida e sua subsequente, é conhecido como período de amostragem T_s [38].

Desta forma, uma restrição é implicada, fazendo com que a saída só possa ser comparada com a referência nos instantes em que seus valores são conhecidos. Logo, tem-se uma sequência que pode ser infinita, mas contável, de valores de erro $e(T_s)$, $e(2T_s)$, $e(3T_s)$, ..., ao invés de um conjunto infinito e não contável que corresponde à uma variável no tempo contínuo. Assim, não apenas a lei de controle deverá ser capaz de determinar uma ação a partir da sequência de número de erros, como essa própria ação será uma sequência de números, o que é ilustrado na Figura 4.

Nota-se que na Figura 4 que a parte do sistema que está dentro da linha tracejada, opera com sequências discretas, enquanto o que está fora, é de tempo contínuo. Ou seja, a informação adquirida em si, é analógica, mas após ser amostrada e passar por um conversor analógico-digital (AD), ela se torna digital. O conversor AD é representado pela chave na Figura 4 denominada interface.

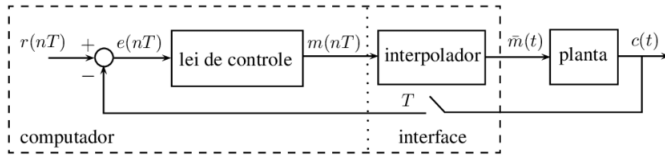
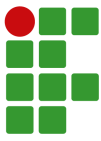


Fig. 4. Diagrama de blocos de um sistema amostrado de controle [38].

Devido ao sistema amostrado de controle ter uma parte discreta e outra contínua, deve-se ser capaz de aplicar a ação de controle discreta ao sistema contínuo, procedimento chamado de interpolação, ou mais comumente, conversão digital-analógico (DA).

Segundo Monzani [39], a frequência de amostragem (T_s^{-1}) deve ser no mínimo dez vezes o valor da frequência do sistema, para assim garantir a estabilidade e manter-se dentro das limitações físicas do sistema.

Com o controlador projetado no domínio da frequência de Laplace, é possível convertê-lo para aplicá-lo em sistemas com componentes digitais, através da Transformada Z [39], convertendo assim, as equações diferenciais do domínio da frequência, em equações a diferenças, do domínio discreto.

No domínio Z, pode-se aplicar alguma técnica de aproximação de diferenciação para a implementação do controle em um microcontrolador, como por exemplo o Método de Euler (*forward difference*), o método diferencial (*backward difference*), ou o método de Tustin [40].

III. METODOLOGIA

Nesta seção são apresentados os métodos e componentes empregados para a obtenção do protótipo final, e também, é realizada a simulação do sistema em malha fechada, utilizando os parâmetros definidos para protótipo.

A. Montagem do Protótipo

1) *Especificações do Projeto*: Para o desenvolvimento do protótipo de uma junta simples em configuração antagonista com tendões acionados por motores, é necessário primeiramente a definição das especificações do projeto. Na Tabela I são apresentados todos os parâmetros provindos do modelo do sistema. Dentre estes parâmetros, os momentos de inércia J_α e J_β respectivamente do Motor 1 e do Motor 2, são definidos como sendo unitários, e o momento de inércia da junta I é definido como sendo 10% do valor dos momentos dos motores. O motivo destes parâmetros serem escolhidos ao invés de calculados, é devido a dificuldade de um cálculo preciso, pois como o sistema real tem muitas variáveis desconhecidas, como por exemplo toda a montagem da redução dos motores, seriam necessários experimentos para identificar estes parâmetros, o que dificulta encontrar um valor preciso. Sabe-se, verificando empiricamente, que o momento de inércia da junta é menor do que o dos motores.

O raio da polia da junta e das polias dos motores r , tem seu diâmetro no valor de 30 mm e são usinadas em alumínio. Já

as constantes que dependem do formato e material utilizados nos elementos de transmissão k e σ , tem seus valores provindos de resultados de ensaios realizados em laboratório, apresentados na Seção IV.

TABELA I

Parâmetros da Planta

Parâmetro	Valor	Unidade
I	0,1	kgm^2
J_α	1	kgm^2
J_β	1	kgm^2
r	0,03	m
k	7,598	Nm/rad
σ	0,1897	$1/rad$

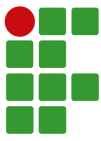
2) *Atuadores*: Para os atuadores da junta, foram utilizados dois motores CC com encoder acoplado, como mostra a Figura 5, possibilitando a medição de posição angular dos motores.



Fig. 5. Motor utilizado como atuador no protótipo.

Os dois motores são idênticos e possuem as seguintes características:

- Tensão de alimentação de 6V;
- Velocidade de 210 rpm;
- Diâmetro do eixo de 4 mm;
- Comprimento do eixo de 12 mm;
- Velocidade sem carga de 210 rpm / 0,13 A
- Máxima eficiência de 2 kgcm / 170 rpm / 2 W / 0,6 A;
- Máximo torque de 5,2 kgcm / 110 rpm / 3,1 W / 1,1 A;
- Torque de parada de 10 kgcm 3,2 A
- Comprimento total motor e eixo de 73 mm;
- Possui caixa de redução 1:34 (1:34.02);
- Encoder na placa.



Ambos os motores acionam as polias, nas quais o cabo de aço está preso, e que com o movimento de rotação destas, esticam os elementos elásticos, alterando a impedância da junta.

Já o encoder acoplado no motor, possui as seguintes características:

- Incremental;
- Tensão de alimentação de 3,3 V;
- Resolução de $11 \times \text{Redução } 34.02 = 341 \text{ PPR}$;

Para a leitura de posição da junta, foi utilizado um potenciômetro com resistência de $22k\Omega$.

3) *Ponte H*: Para possibilitar o acionamento dos motores com o sinal de controle proveniente do microcontrolador utilizado, foi utilizado um drive Ponte H VN3SP30. Este drive possui as seguintes características:

- Tensão de operação de 5-16 V;
- Chip ST VN3SP30;
- Controla 2 motores CC simultaneamente;
- Corrente de operação máxima de 14 A por canal;
- Corrente de pico de 30 A;
- Tensão lógica de 3,3 V;
- Frequência máxima do PWM de 20 kHz.

Esse drive é projetado para controlar cargas indutivas como relés, solenóides, motores DC e motores de passo, permitindo o controle não só do sentido de rotação do motor, como também da sua velocidade, utilizando um sinal de tensão PWM.

4) *STM32*: O microcontrolador escolhido para alocar o sistema de controle discretizado, foi o STM32F103C8T6, mais especificamente, o kit de desenvolvimento conhecido como "Blue Pill".

De acordo com as características exigidas para a construção do protótipo, é necessário que o microcontrolador atenda a alguns requisitos:

- 1 entrada analógica, com resolução mínima de 8 bits;
- 2 saídas PWM;
- 5 I/O digitais;
- 4 timers;
- Montado em placa de desenvolvimento;

O STM32 é capaz de atender a todos os requisitos necessários para a execução do protótipo, já que possui um módulo ADC integrado e opera com baixos níveis de tensão, sendo simples o condicionamento dos sinais para os níveis requeridos pelo microcontrolador. Os sinais analógicos são convertidos e representados de forma digital através de 12 bits, logo, tem-se 4096 níveis de tensão para a representação das grandezas medidas, o que atende de forma satisfatória a leitura de tensão do potenciômetro da junta.

Outro fator importante, é a montagem em uma placa de desenvolvimento. Esta possui um conversor serial integrado que simplifica a gravação dos programas durante os testes.

Os *timers* presentes têm modos especiais para utilização com encoder, o que torna a leitura de posição dos motores precisa. Com os timers, pode ser gerado sinal PWM, que varia de 0 a 3,3 V, e também é possível ter interrupção.

As saídas digitais são utilizadas para controlar o sentido de rotação dos motores, já que a Ponte H utilizada, controla o sentido de rotação a depender do nível lógico aplicado em seus terminais. Foi necessário também uma entrada digital, para determinar a rigidez de referência, facilitando os testes.

O esquema elétrico do protótipo pode ser visto na Figura 6

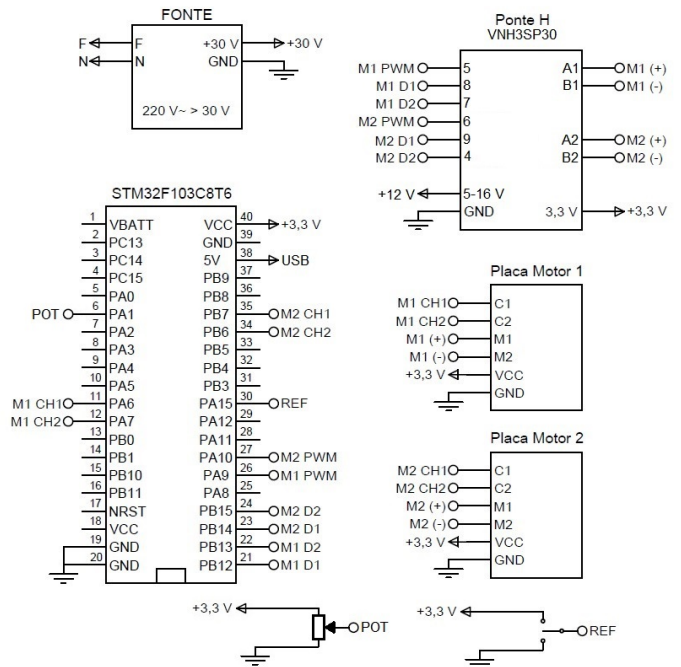


Fig. 6. Esquema elétrico do protótipo.

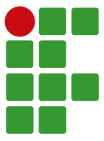
Uma imagem do protótipo final pode ser vista no Anexo I.

B. Projeto e Discretização dos Controladores

Para projetar o sistema de controle, existem as técnicas de controle moderno, que utilizam o modelo matemático da planta na representação de espaço de estados, ou as técnicas de controle clássico, que necessitam do modelo representado em funções de transferência no domínio da frequência. Neste trabalho, embora tenha-se a modelagem matemática em representação de espaço de estados, optou-se por passar este modelo para o domínio da frequência, e projetar o controle por métodos clássicos. Técnicas de controle moderno exigiriam o projeto de um filtro de Kalman digital, o que tornaria a implementação mais complexa nesta fase do projeto.

O diagrama do controle proposto neste trabalho, pode ser visualizado na Figura 7.

De acordo com a impedância de referência, o controlador de impedância calcula o sinal de controle necessário para chegar na referência, o qual passa pelo conversor de posição para



impedância, e assim chega-se na posição de α e β que os motores devem estar. Com estas posições, que são a referência para os controladores de posição, os controladores de posição estipulam qual a tensão que deve ser enviada para os motores, para que estes exerçam o torque necessário para chegar na impedância de referência.

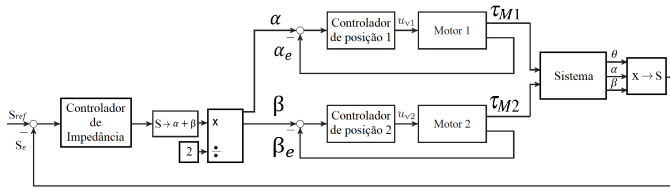


Fig. 7. Diagrama do controle proposto.

De acordo com a Eq. 34, tem-se a equação que relaciona o torque do Motor 1 ao ângulo α sendo:

$$\frac{\alpha}{\tau_{M1}} = \frac{s^4 + as^3 + bs^2 + cs + d}{s^6 + es^4 - fs^3 + gs^2 - h}, \quad (35)$$

onde $a = 2,765 \times 10^{-17}$, $b = 27,24m$, $c = 1,467 \times 10^{-20}$, $d = 16,83\mu$, $e = 28,54m$, $f = 6,019 \times 10^{-35}$, $g = 35,34\mu$ e $h = 2,317 \times 10^{-41}$.

Sabe-se também que a relação de β e o torque do Motor 2 é igual a Eq.35, logo, o controlador de torque de ambos é igual.

Verifica-se que esta planta é de 6ª ordem, possui 6 pólos e 4 zeros, e é criticamente estável, devido a tanto os pólos quanto os zeros possuírem parte real nula.

Existem diversas técnicas de projeto de controle, mas por conta da simplicidade, escolheu-se projetar esses ganhos através do método de Lugar Geométrico das Raízes.

Como a planta não possui parte real, utiliza-se um controlador PD, com realimentação negativa unitária, para que seja inserido no sistema um zeros com parte real negativa. Como não há parte real no sistema, optou-se colocar o zero do controlador PD próximo à origem do eixo, no valor de $-0,01$ rad/s. Sabe-se que o controlador PD é representado por:

$$C(s) = K_p(sT_d + 1) \quad (36)$$

Logo, sabendo a localização do zero, e que a localização deste se dá em $\frac{1}{T_d}$, tem-se que $T_d = 100$ s. Para o valor de K_p , foi encontrado de maneira empírica, como sendo 200, verificando a resposta ao degrau unitário da malha fechada e vendo que valores maiores não causam muita diferença na resposta, ou geram sobressinal. O gráfico de lugar das raízes da malha direta desta planta pode ser visto na Figura 8, na qual não está representado um polo em -10 rad/s.

A resposta ao degrau unitário da malha fechada pode ser vista na Figura 9, no qual pode ser visto que não existe um sobressinal, e o tempo de assentamento em regime permanente é entre dois e

três segundos.

Também na Figura 9, é possível visualizar o sinal de controle, o qual possui saturação em 12 e -12 V, devido a alimentação da Ponte H. Verifica-se que este sinal vai até sua saturação positiva, logo após, vai para um valor negativo próximo de -10 V, e em regime permanente, esse sinal fica oscilando próximo de 0,9 V.

Com o controle dos ângulos α e β definido, é possível então projetar o controlador de impedância. Visualizando o diagrama de controle proposto, consegue-se perceber que a “planta” para projetar o controle de impedância engloba desde o conversor de impedância para posição de α e β , até o conversor de posição para impedância, incluindo o controlador de torque.

A equação de transferência que representa esse sistema está expressa na Eq.37.

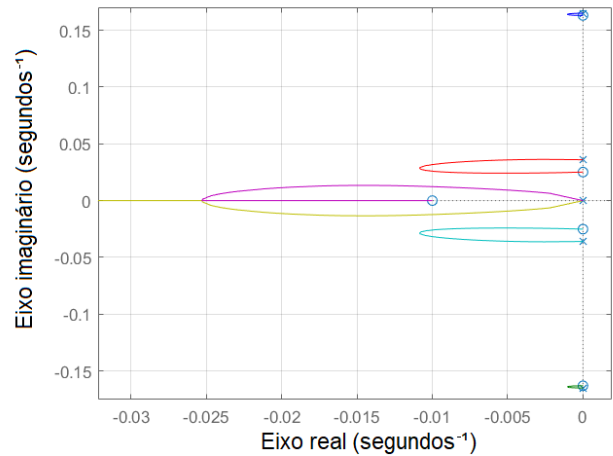


Fig. 8. Lugar das raízes da relação entre α e τ_{M1} .

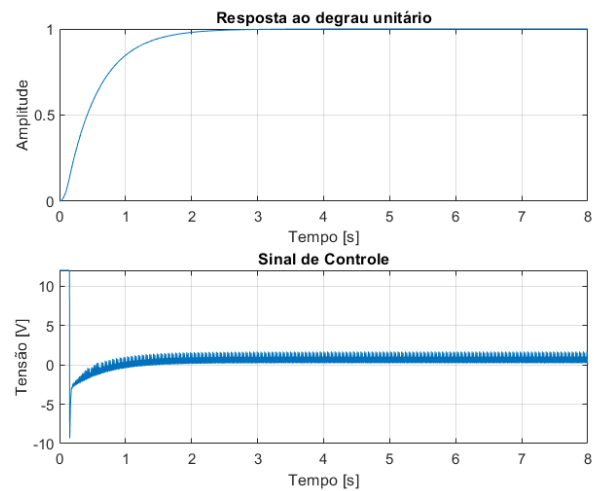
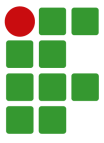


Fig. 9. Resposta ao degrau unitário da malha fechada da relação entre o ângulo α e o torque do Motor 1 Simulado.



$$S_{sis} = \frac{as^2 + bs + c}{s^4 + ds^3 + as^2 + es + c}, \quad (37)$$

onde $a = 2,002 \times 10^5$, $b = 2,454 \times 10^{-10}$, $c = 126,6$, $d = 9,99$ e $e = 0,01219$.

Verifica-se que a planta da impedância é de 4ª ordem, possui 4 pólos e 2 zeros, possuindo 2 pólos e os 2 zeros com parte real nula, e 2 pólos com parte imaginária e parte real não nulos.

Como a planta não possui integrador natural, utiliza-se um controlador PI, com realimentação negativa unitária, para que o sistema consiga garantir referência. Sabe-se que o controlador PI é representado por:

$$C(s) = \frac{K_p}{s} \left(s + \frac{1}{T_i} \right) \quad (38)$$

O controlador PI também insere um zero no sistema, o qual é localizado em $\frac{1}{T_i}$. Logo, optou-se por colocar o zero em $-0,5$ rad/s, ou seja, $T_i = 2$ s. Para o valor de K_p , foi encontrado empiricamente como sendo 3, verificando a resposta ao degrau unitário da malha fechada e vendo que valores maiores não causam muita diferença na resposta, ou geram sobressinal. O gráfico de lugar das raízes da malha direta desta planta pode ser visto na Figura 10.

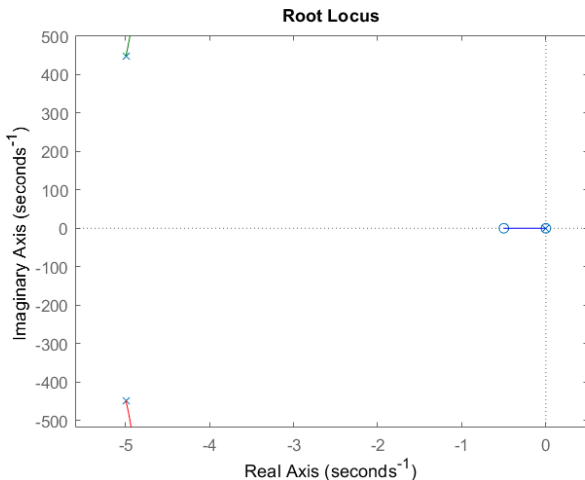


Fig. 10. Lugar das raízes da impedância.

A resposta ao degrau unitário da malha fechada pode ser vista na Figura 11, no qual pode ser visto que não existe um sobressinal, e o tempo de assentamento em regime permanente é entre 1 e 2 segundos.

Também na Figura 11, é possível visualizar o sinal de controle. Verifica-se que este sinal vai até um valor próximo de 3, logo após, diminui, estabilizando em regime permanente em 1.

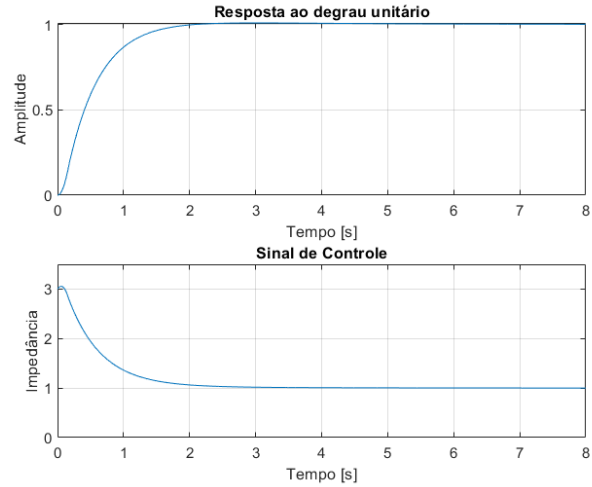


Fig. 11. Resposta ao degrau unitário da malha fechada da impedância Simulado.

Para a discretização e implementação dos controladores digitais no microcontrolador, é necessário calcular o período de amostragem, o qual é encontrado a partir dos polos de malha fechada do sistema.

Tanto para o controlador de impedância, quanto para o de torque, foi utilizando o Teorema da Amostragem de Nyquist o qual necessita-se saber o valor do polo mais rápido da planta em malha fechada. Para a planta da relação entre α e o torque do motor 1, tem-se o pólo mais rápido localizado em -1000 rad/s, e para a planta da impedância, em -4000 rad/s. Utilizando o pólo mais rápido dentre as duas plantas, a frequência de banda é:

$$f_b = \frac{\omega_b}{2\pi} = \frac{4000}{2\pi} = 636,62 \text{ Hz} \quad (39)$$

Para calcular a frequência de amostragem f_s utiliza-se uma constante K . Sendo $K = 10$, a frequência de amostragem é:

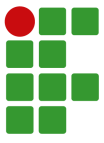
$$f_s = K \cdot f_b = 10 \cdot 636,62 = 6366,2 \text{ Hz} \quad (40)$$

Com a frequência de amostragem, e sabendo que o período é o inverso da frequência, o período de amostragem para as plantas é igual a:

$$T_s = f_s^{-1} = 6366,2^{-1} = 157,08 \mu s \quad (41)$$

Sabendo que T_s tem um valor máximo de $157,08 \mu s$, utilizou-se o período de amostragem no valor de $100 \mu s$ no protótipo.

Com o período de amostragem definido, deve-se discretizar os controladores. Para isso, utilizou-se o método de *backward Euler*. Sabendo que k representa a amostra atual, e $k - 1$ a amostra passada, espaçada pelo tempo de amostragem, para o controlador PI tem-se:



$$u_1(k) = u_1(k-1) + K_p \left(1 + \frac{Ts}{Ti}\right) erro(k) - K_p erro(k-1) \quad (42)$$

$$u_2(k) = -2kr^2\sigma \quad (43)$$

$$u(k) = u_1(k) + u_2(k) \quad (44)$$

E para o controlador PD:

$$u_1(k) = u_1(k-1) + K_p erro(k) - K_p erro(k-1) \quad (45)$$

$$u_2(k) = K_p \frac{Td}{Ts} (leitura(k) - leitura(k-1)); \quad (46)$$

$$u(k) = u_1(k) + u_2(k) \quad (47)$$

O código final implementado no microcontrolador pode ser visto no Anexo II.

IV. RESULTADOS

O protótipo de uma junta simples em configuração antagônica foi construído para provar que é possível controlar a impedância mecânica de partes de robôs. Para isto, foi necessário montar um elemento elástico com característica de alongamento não linear, que neste caso foi o SAT, possuindo esta característica de forma exponencial. Logo, nesta seção, são apresentados os resultados de ensaio do elemento elástico, além dos resultados de controle, os quais foram adquiridos via *STM Studio*, e filtrados no software *Matlab*, por um filtro de 3ª ordem com frequência de corte de 0,05 rad/s, devido aos ruídos presentes nestes resultados.

A. SAT

Para obter a característica de deformação por torque do SAT, foi utilizado uma máquina de ensaios de tração até 100kN da EMIC® para aplicações de tração/compressão em materiais metálicos, modelo DL10000. O resultado deste ensaio pode ser visualizado na Fig. 12, e a equação que melhor se ajusta à curva pode ser vista na Eq. 48.

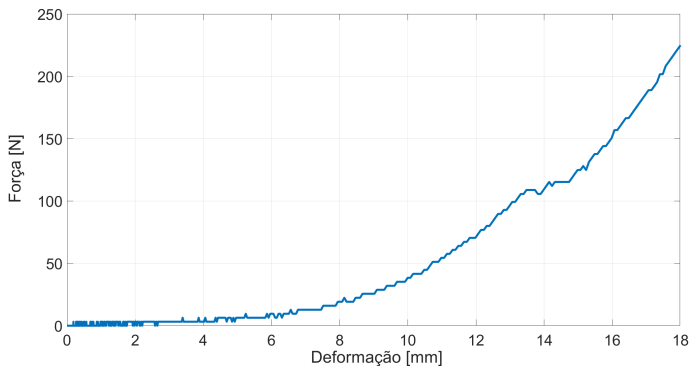


Fig. 12. Característica de tração do SAT.

$$F(x) = 7,598(e^{0,1897x} - 1), R^2 = 0,9946. \quad (48)$$

Com auxílio da Eq. 48, retirada do ajuste feito pelo método dos mínimos quadrados sobre a curva obtida no ensaio de tração, é possível obter as constantes utilizadas nos elementos de transmissão, σ e k , que são respectivamente 0,1897 e 7,598. Estes ganhos foram considerados iguais para ambos elementos elásticos.

B. Controle

Para validar a eficácia dos controladores, alguns testes foram realizados. Dentre estes testes, o primeiro verifica se é possível alterar a impedância de referência. Nos demais testes, foi mantida a impedância de referência como sendo a inicial, pendurando diversas cargas na ponta do eixo da junta, para que o ângulo θ da junta alterasse, alterando a impedância, para verificar se o controlador consegue manter o valor da impedância quando uma perturbação ocorre. Também foi alterado o valor dos ganhos de controle, a fim de analisar os resultados para diferentes condições do sistema de controle.

O quinto teste é feito desligando o controlador de impedância, para comparar o resultado com o teste do controlador proposto, provando que sem esse controle, a referência de impedância não é seguida.

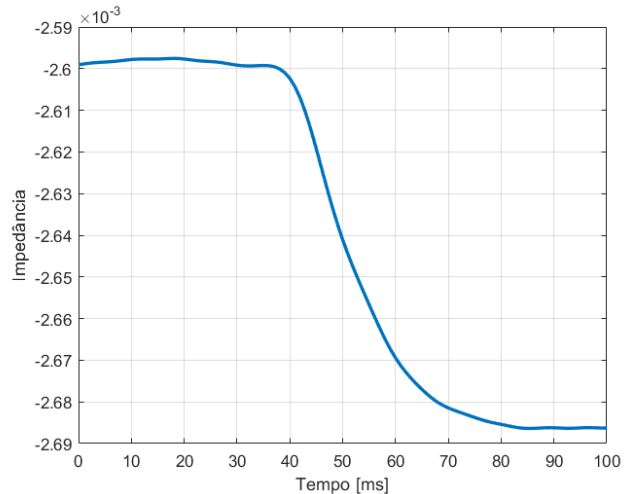
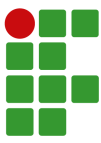


Fig. 13. Mudança de referência de impedância.

No primeiro teste realizado, foi alterado a referência de impedância, mudando o valor da impedância inicial de $-0,00259 \text{ Nm}^3/\text{rad}^2$ para um valor 4,5% maior, ou seja, de $-0,00271 \text{ Nm}^3/\text{rad}^2$, utilizando os ganhos dos controladores propostos na Seção III, porém, alterando o K_p do controlador PD para 10, que foi um ganho obtido empiricamente com o protótipo. O resultado deste teste pode ser visualizado na Figura 13. O motivo de não escolher um valor maior de impedância para a alteração se dá devido ao torque máximo dos motores ser muito baixo em



comparação com o torque necessário para esticar os elementos elásticos.

Como pode ser visto, a mudança de referência se deu com um tempo de assentamento próximo de 0,03 segundos. Outro ponto que verifica-se olhando a imagem, é que a referência não é atingida em 100% de seu valor, tanto na referência inicial, quanto na referência alterada. Para a referência inicial, há um erro próximo de 0,4%, e na referência alterada, um erro próximo de 0,8%.

Iniciando os testes com a impedância de referência sendo a inicial, e pendurando pesos na ponta do eixo da junta, no primeiro teste, foi utilizando os ganhos dos controladores propostos anteriormente, onde o controlador PD possui $Kp = 10$ e $Td = 100$ s, e os ganhos do controlador PI como $Kp = 3$ e $Ti = 2$ s. Foi colocado um peso de 237 g na pontado eixo, próximo a 1 segundo, e logo após, próximo de 2 segundos, retirado.

Como pode-se ver na Figura 14, a posição da junta variou rapidamente, chegando a um limite próximo de -10° , e ao ser retirado o peso, retornou com um sobrepasso próximo de 4° , estabilizando na posição 0° após meio segundo. Nesta mesma figura, é possível ver que o sinal de impedância fica variando o tempo todo entre $-0,00259 \text{ Nm}^3/\text{rad}^2$ e $-0,0026 \text{ Nm}^3/\text{rad}^2$, ou seja, fica variando e causando um erro de aproximadamente 0,4% da impedância de referência. Ao retirar a carga, a impedância teve um pico que alterou seu valor para próximo de $-0,00258 \text{ Nm}^3/\text{rad}^2$, o que se deve à posição dos motores ter tido uma pequena pausa em sua oscilação neste momento.

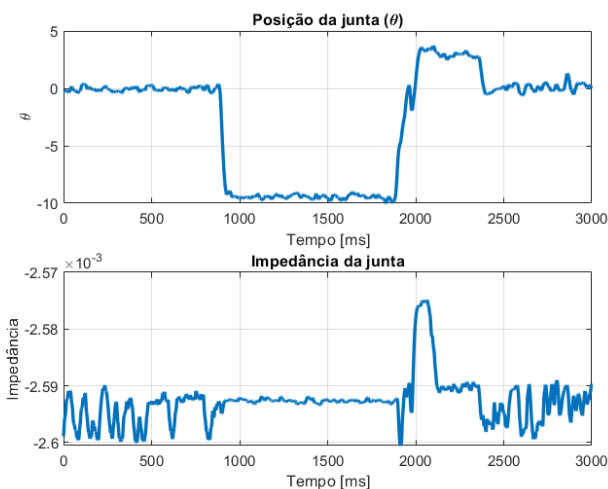


Fig. 14. Ângulo θ e impedância: Teste 1.

Olhando para a Figura 15, no sinal de controle do controlador de impedância, verifica-se que o sinal de controle diminui sua oscilação quando a carga é aplicada, e quando a mesma é retirada, o sinal volta a aumentar sua oscilação. Na mesma figura, olhando para a posição dos motores, verifica-se que estes não se mantem em uma posição fixa, onde o Motor 1 varia positivamente, e o Motor 2 negativamente.

O mesmo comportamento de os motores não conseguirem manter-se em uma posição fixa, é visto em todos os demais testes, com outros ganhos e outras cargas. A partir deste comportamento, constata-se que a maior parte das oscilações presentes no resultados, é devido ao fato de os motores não pararem de oscilar.

O motivo de os motores ficarem oscilando, se dá predominantemente a circunstância de que os encoders utilizados só conseguem medir posições de valores inteiros, criando uma zona morta de posição dos motores. Outro fator que influência nesta oscilação, são os atritos desconhecidos que existem tanto no motor quanto na redução, além do próprio momento de inércia dos motores que é desconhecido.

Porém, como pôde ser visto, o objetivo de manter a impedância controlada em regime permanente, ficou variando próximo do valor de referência, independente da posição dos motores ser estável ou não.

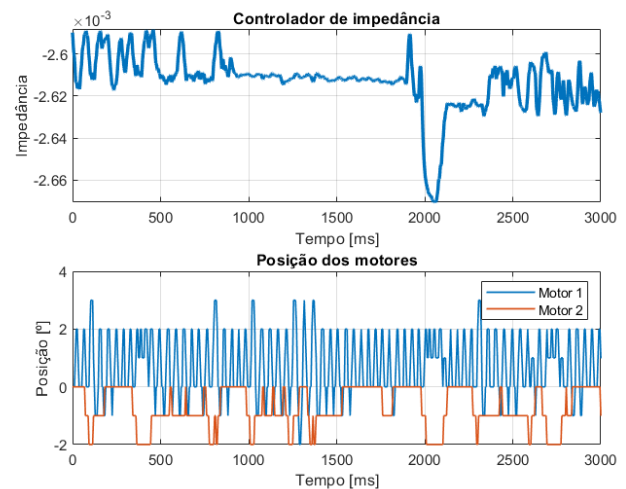
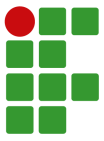


Fig. 15. Sinal do controlador de impedância e posição dos motores: Teste 1.

Para o segundo teste, manteve-se os ganhos dos controladores, porém, aumentou-se a o peso da carga utilizada no teste, para um valor de 607 g. A carga é aplicada a ponta do eixo em 0,4 segundos, e retirada em 1,8 segundos.

Nos resultados do segundo teste, verifica-se novamente o mesmo comportamento do primeiro teste em relação a posição dos motores, porém desta vez, o Motor 2 ficou fixo em -1° , e apenas o Motor 1 ficou variando sua posição angular. Como pode-se ver, a posição θ teve uma variação maior, devido ao maior peso, porém, mesmo assim, a impedância da junta ainda conseguiu se manter próxima a referência, embora tenha muito ruído. Novamente o sinal de controle diminui sua variação quando a carga é aplicada.

Constata-se com os Teste 1 e Teste 2 que mesmo com uma carga maior, a impedância varia dentre os mesmos valores, comprovando que o controlador se comporta igualmente para diversas cargas.



O terceiro teste é realizado da mesma forma que o primeiro, porém, desta vez, os ganhos do controlador de impedância são alterados para $K_p = 0,3$ e $T_i = 0,2$ s. A carga é a mesma de 237 g, sendo aplicada em 0,5 segundos e retirada próximo de 2,5 segundos.

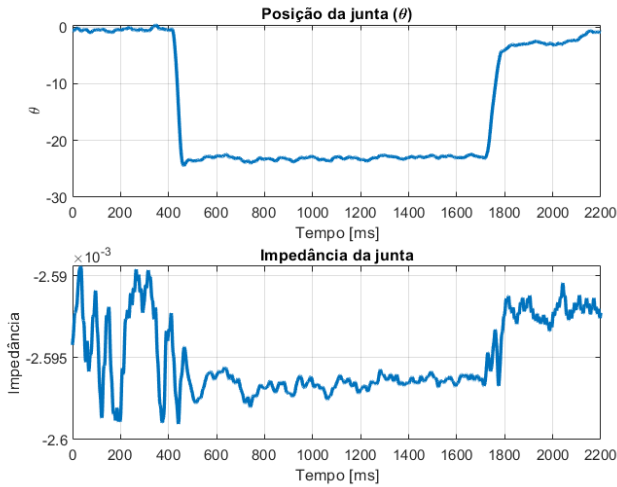


Fig. 16. Ângulo θ e impedância: Teste 2.

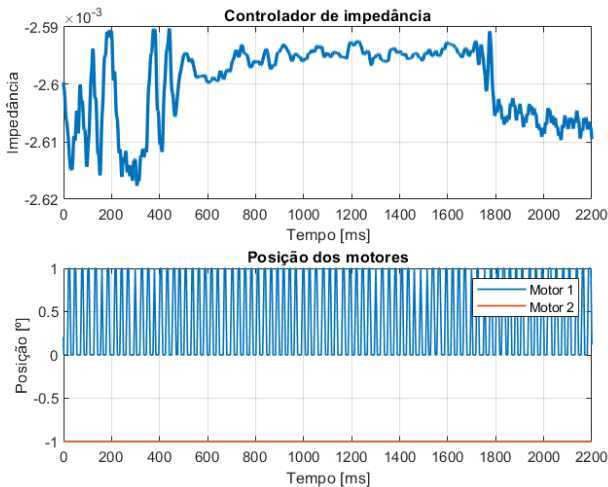


Fig. 17. Sinal do controlador de impedância e posição dos motores: Teste 2.

Novamente, o comportamento oscilatório da posição angular dos motores se repete, porém desta vez, o sinal de controle não diminui sua oscilação quando a carga é aplicada. A impedância fica oscilando em torno de $-0,002594 \text{ Nm}^3/\text{rad}^2$, ou seja, está com um pequeno erro de 0,15%. Verifica-se também, que a posição da junta está parecida com a do Teste 1, o que era esperado devido ter-se utilizado a mesma carga e a impedância estar em um valor parecido.

No quarto teste, o mesmo procedimento se repetiu, com a mesma carga, mas alterando os ganhos do controlador de

impedância para $K_p = 30$ e $T_i = 20$ s. A carga é a mesma de 237 g, sendo aplicada próximo de 0,7 segundos e retirada próximo de 4 segundos.

Outra vez, o comportamento oscilatório da posição angular dos motores se repete, e neste caso, após 3 segundos, o Motor 2 para de oscilar se mantendo em -5° . Em virtude disto, o sinal de controle é bem oscilatório, e não diminui sua oscilação quando a carga é aplicada. A impedância fica oscilando em torno de $-0,002592 \text{ Nm}^3/\text{rad}^2$ quando a carga é aplicada, o que produz um erro de 0,08%. Verifica-se novamente que a posição da junta está parecida com a dos testes 1 e 3, próxima de -10° .

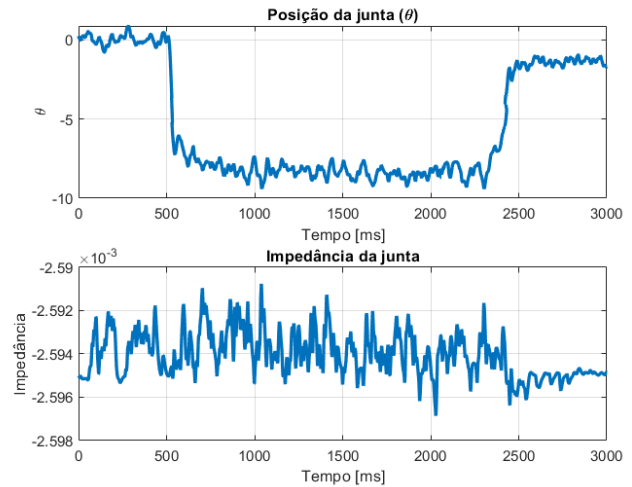


Fig. 18. Ângulo θ e impedância: Teste 3.

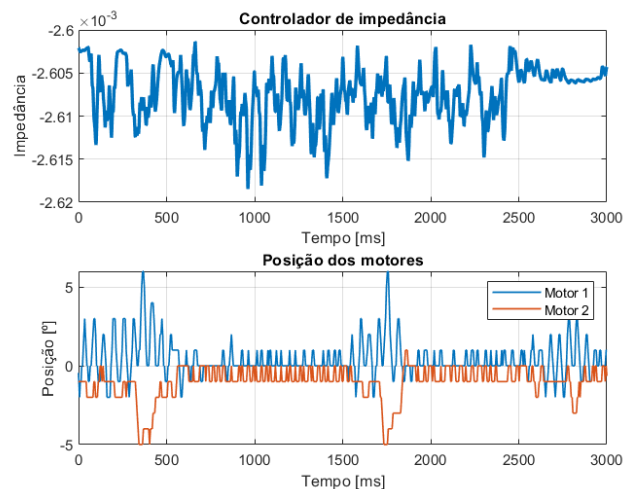
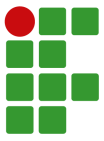


Fig. 19. Sinal do controlador de impedância e posição dos motores: Teste 3.

Constata-se que entre o Teste 1, Teste 3 e Teste 4, o que teve pior resultado foi o Teste 1, devido a ser o que teve maior erro na impedância em relação a impedância de referência, e o único que apresentou sobrepasso. Já o Teste 4, foi o que teve o menor erro



na impedância, o que se deve ao fato de ter-se aproximado da origem o zero inserido pelo controlador PI, deixando o sistema de controle mais rápido.

No quinto teste, é comparado o comportamento da impedância dos testes 1, 3 e 4, com um teste sem o controlador de impedância, apenas mantendo o controlador de posição, e deixando a referência dos motores em 0°, para que a impedância seja a inicial.

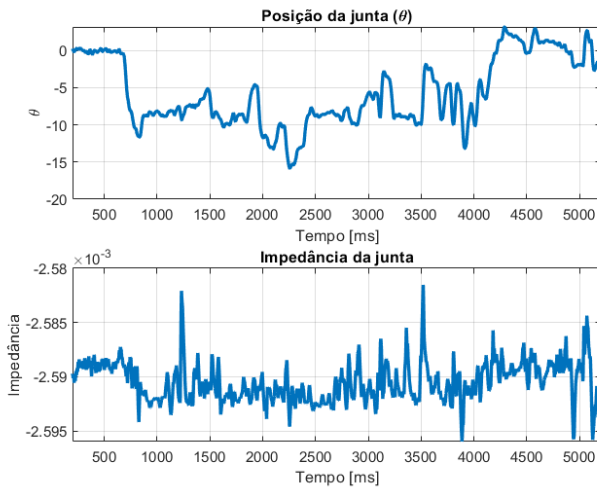


Fig. 20. Ângulo θ e impedância: Teste 4.

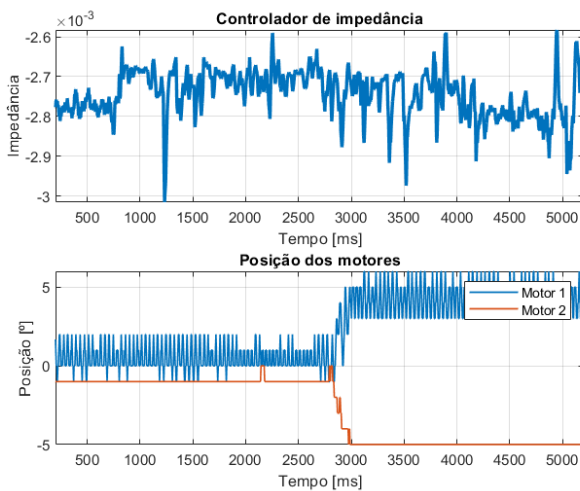


Fig. 21. Sinal do controlador de impedância e posição dos motores: Teste 4.

Conforme a Figura 22, pode-se visualizar que para os Teste 1, Teste 3 e Teste 4, a impedância conseguiu se manter controlada próximo ao valor de referência, já para o caso sem controle, ela foi alterada quando a carga foi aplicada próximo de 0,4 segundos, percebe-se que iniciou-se uma oscilação, pois o peso fez os motores se moverem, e ao tentar retornar, ficaram oscilando, e que essa oscilação não foi em torno da impedância

inicial deste gráfico, o que mostra que a referência não foi seguida. Quando a carga é retirada, próximo de 1,6 segundos, a impedância volta ao seu valor inicial.

O gráfico da impedância sem controlador de impedância também comprova que os motores que geram todo o ruído presente nos resultados, pois nos momentos que os motores estão em zero, quase nenhum ruído aparece, apenas o ruído das conexões elétricas.

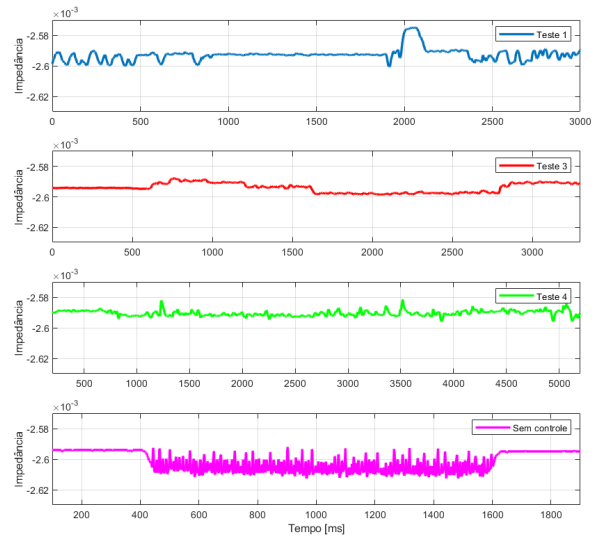


Fig. 22. Comparação entre ganhos com uma carga de 237 g aplicada a ponta do eixo da junta.

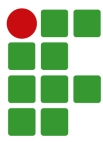
V. CONCLUSÃO

Para o projeto de um controlador de impedância de uma junta robótica em configuração antagonista, primeiramente deve-se modelar matematicamente a junta em sua configuração escolhida, para que toda a dinâmica do sistema seja exposta, e de acordo com as especificações do protótipo, o sistema de controle possa ser desenvolvido.

Neste trabalho, a modelagem matemática foi realizada através da Equação de Lagrange, chegando-se no modelo em espaço de estados, e o convertendo para funções de transferência, para que o sistema de controle pudesse ser projetado por métodos clássicos de controle, que neste caso foi o método de Lugar Geométrico das Raízes (LGR).

Um dos critérios para que seja possível controlar a impedância de uma junta em configuração antagonista, é o de que essa possa ser alterada de maneira não linear, o que leva a ser utilizado um elemento elástico com características de deformação exponencial, conhecido como SAT.

Com o sistema de controle projetado e discretizado, e com o protótipo montado, testes práticos foram empregados para comprovar a eficácia do controle de impedância. Os testes comprovaram que é possível realizar o controle de impedância de



acordo com o protótipo montado.

Como dificuldades encontradas, destaca-se os encoders utilizados, cuja resolução se mostrou baixa para este projeto, lendo apenas posições angulares de valor inteiro, o que fez com que para conseguir manter o sinal de controle, os motores ficassem variando a posição, o que fez com que surgisse muito ruído no sistema.

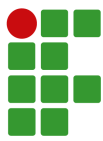
Outra dificuldade encontrada foi o torque máximo dos motores utilizados, o qual para poder variar mais a impedância, deveria ser maior.

Para a continuidade do trabalho, sugere-se que sejam utilizados encoders com maior resolução, e motores com maior torque. Caso seja optado por utilizar os mesmos motores, pode-se trocar os elementos elásticos por outros com menor rigidez, ou até mesmo com a resposta de força/deformação que não seja exponencial, desde que esta seja não linear.

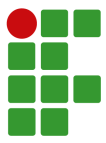
Outra sugestão, é a de melhorar a modelagem matemática do sistema, cuja possui diversos atritos que não foram considerados, além de momentos de inércia que neste caso foram escolhidos, mas que com um estudo mais aprofundado, e equipamentos adequados, poderiam ser calculados. Além disso, a implementação de um filtro de Kalman discreto, possibilita que o sistema de controle possa ser projetado por métodos modernos.

REFERÊNCIAS

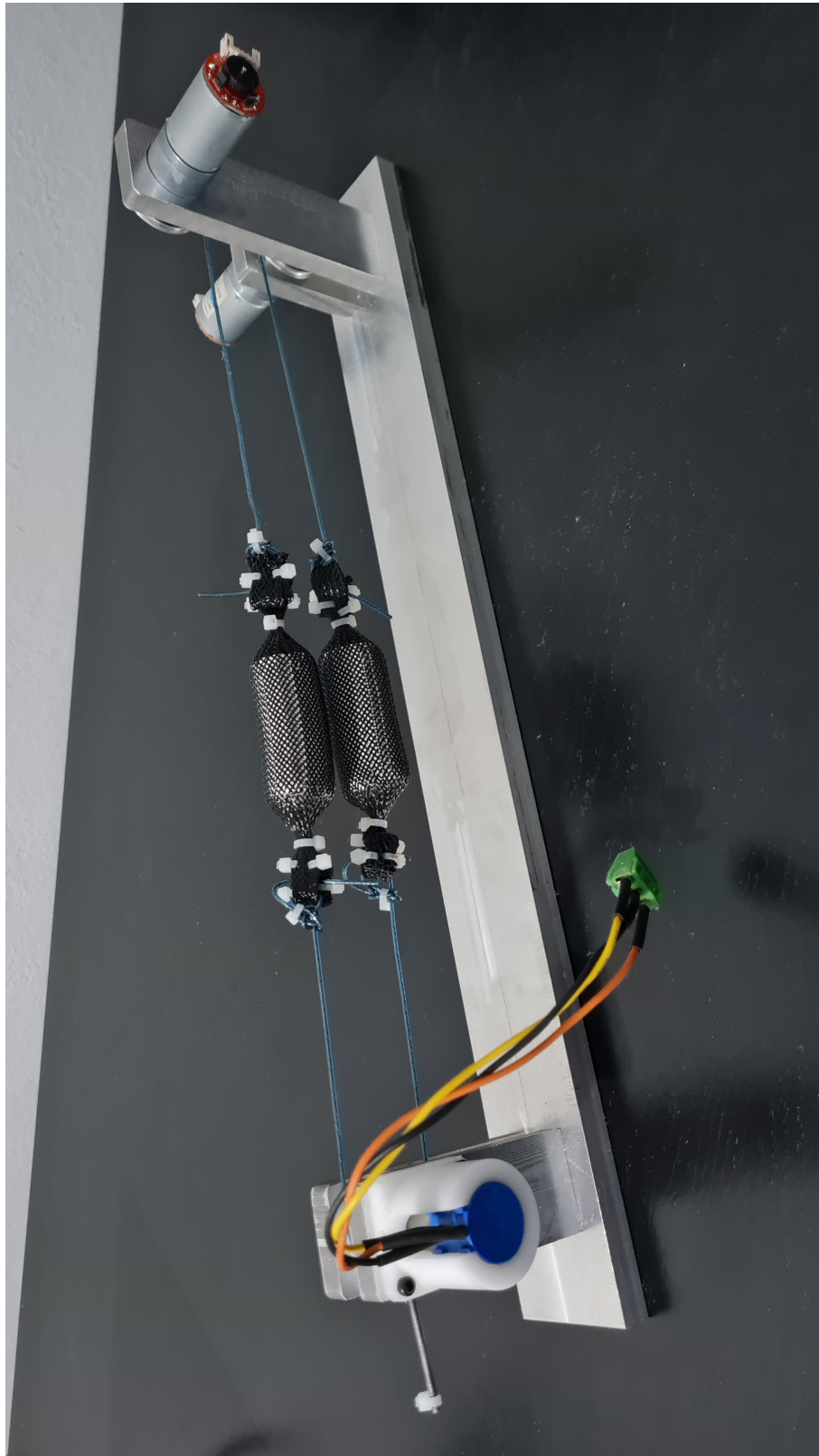
- [1] A. Pereira, E. de Oliveira Simonetto, “Indústria 4.0: conceitos e perspectivas para o Brasil”, *Revista da Universidade Vale do Rio Verde*, vol. 16, no. 1, 2018.
- [2] A. Tödtheide, T. Lilge, S. Haddadin, “Antagonistic impedance control for pneumatically actuated robot joints”, *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 161–168, 2015.
- [3] N. Hogan, “Impedance control: An approach to manipulation—Part I: Theory”, *ASME J Dyn Syst Meas Control*, vol. 107, pp. 1–7, 1985.
- [4] C. Ott, *Cartesian impedance control of redundant and flexible-joint robots*, Springer, 2008.
- [5] A. Sabanovic, K. Ohnishi, *Motion control systems*, John Wiley & Sons, 2011.
- [6] P. L. Gatti, *Applied structural and mechanical vibrations: theory, methods and measuring instrumentation*, CRC Press, 1999.
- [7] A. Khalid, P. Kirisci, Z. H. Khan, Z. Ghrairi, K.-D. Thoben, J. Pannek, “Security framework for industrial collaborative robotic cyber-physical systems”, *Computers in Industry*, vol. 97, pp. 132–145, 2018.
- [8] K. Goris, J. Saldien, B. Vanderborght, D. Lefeber, “How to achieve the huggable behavior of the social robot Probo? A reflection on the actuators”, *Mechatronics*, vol. 21, no. 3, pp. 490–500, 2011.
- [9] W. D. Stiehl, C. Breazeal, K.-H. Han, J. Lieberman, L. Lalla, A. Maymin, J. Salinas, D. Fuentes, R. Toscano, C. H. Tong, *et al.*, “The huggable: a therapeutic robotic companion for relational, affective touch”, in *ACM SIGGRAPH 2006 emerging technologies*, pp. 15–es, 2006.
- [10] M. H. Raibert, H. B. Brown Jr, M. Chepponis, “Experiments in balance with a 3D one-legged hopping machine”, *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 75–92, 1984.
- [11] S. Collins, A. Ruina, R. Tedrake, M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers”, *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.
- [12] B. Vanderborght, R. Van Ham, B. Verrelst, M. Van Damme, D. Lefeber, “Overview of the Lucy project: Dynamic stabilization of a biped powered by pneumatic artificial muscles”, *Advanced Robotics*, vol. 22, no. 10, pp. 1027–1051, 2008.
- [13] K. Sreenath, H.-W. Park, I. Poulakakis, J. W. Grizzle, “A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL”, *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.
- [14] M. Hutter, C. D. Remy, M. A. Hoepflinger, R. Siegwart, “Efficient and versatile locomotion with highly compliant legs”, *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 449–458, 2012.
- [15] Y. Huang, B. Vanderborght, R. Van Ham, Q. Wang, M. Van Damme, G. Xie, D. Lefeber, “Step length and velocity control of a dynamic bipedal walking robot with adaptable compliant joints”, *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 598–611, 2012.
- [16] A. Albu-Schaffer, M. Fischer, G. Schreiber, F. Schoppa, G. Hirzinger, “Soft robotics: what cartesian stiffness can obtain with passively compliant, uncoupled joints?”, in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 4, pp. 3295–3301, IEEE, 2004.
- [17] E. Ohashi, K. Ohnishi, “Variable compliance control based on soft-landing trajectory for hopping robot”, in *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, vol. 1, pp. 117–122, IEEE, 2004.
- [18] C. Mitsantisuk, K. Ohishi, S. Katsura, “Variable mechanical stiffness control based on human stiffness estimation”, in *2011 IEEE International Conference on Mechatronics*, pp. 731–736, IEEE, 2011.
- [19] F. Petit, A. Albu-Schäffer, “Cartesian impedance control for a variable stiffness robot arm”, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4180–4186, IEEE, 2011.
- [20] T. Wimböck, C. Ott, G. Hirzinger, “Immersion and invariance control for an antagonistic joint with nonlinear mechanical stiffness”, in *49th IEEE conference on decision and control (CDC)*, pp. 1128–1135, IEEE, 2010.

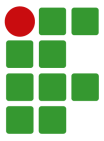


- [21] S. Wolf, G. Hirzinger, “A new variable stiffness design: Matching requirements of the next robot generation”, in *2008 IEEE International Conference on Robotics and Automation*, pp. 1741–1746, IEEE, 2008.
- [22] M. Okada, S. Ban, Y. Nakamura, “Skill of compliance with controlled charging/discharging of kinetic energy”, in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 3, pp. 2455–2460, IEEE, 2002.
- [23] S. A. Migliore, E. A. Brown, S. P. DeWeerth, “Biologically inspired joint stiffness control”, in *Proceedings of the 2005 IEEE international conference on robotics and automation*, pp. 4508–4513, IEEE, 2005.
- [24] T. Miura, T. Shirai, T. Tomioka, “Proposal of joint stiffness adjustment mechanism SAT”, in *JSME Conference on Robotics and Mechatronics*, vol. 2, p. 29, 2002.
- [25] T. Shirai, N. Tanaka, T. Tomioka, “Mechanism and characteristic of non-linear spring SAT”, *Robotics Society of Japan Scholarly Lecture Presentation’03*, 2003.
- [26] Y. Jiang, S. Ding, D. Zhao, W. Ji, “Straight-Line Tracking Control of an Agricultural Vehicle with Finite-Time Control Technique”, *Asian Journal of Control*, vol. 17, no. 6, pp. 2218–2228, 2015.
- [27] R. Mebarki, V. Lippiello, “Image-based control for aerial manipulation”, *Asian Journal of Control*, vol. 16, no. 3, pp. 646–656, 2014.
- [28] Z. Miao, Y. Wang, “Adaptive control for simultaneous stabilization and tracking of unicycle mobile robots”, *Asian Journal of Control*, vol. 17, no. 6, pp. 2277–2288, 2015.
- [29] M. T. Mason, J. K. Salisbury Jr, “Robot hands and the mechanics of manipulation”, , 1985.
- [30] M. W. Spong, “Modeling and control of elastic joint robots”, , 1987.
- [31] K. Shimamoto, D. Suzuki, K. Ohnishi, “Bilateral control for 4-DOF manipulator with a tendon-driven spherical joint mechanism”, in *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*, pp. 290–295, IEEE, 2014.
- [32] B. Vanderborght, A. Albu-Schäffer, A. Bicchi, E. Burdet, D. G. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh, *et al.*, “Variable impedance actuators: A review”, *Robotics and autonomous systems*, vol. 61, no. 12, pp. 1601–1614, 2013.
- [33] G. Palli, C. Natale, C. May, C. Melchiorri, T. Wurtz, “Modeling and control of the twisted string actuation system”, *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 664–673, 2012.
- [34] G. Palli, M. Hosseini, L. Moriello, C. Melchiorri, “Modeling and identification of a variable stiffness joint based on twisted string actuators”, in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1757–1762, IEEE, 2015.
- [35] H. Kobayashi, R. Ozawa, “Adaptive neural network control of tendon-driven mechanisms with elastic tendons”, *Automatica*, vol. 39, no. 9, pp. 1509–1519, 2003.
- [36] G. Palli, *Model and control of tendon actuated robots*, Ph.d. dissertation, University of Bologna, 2007.
- [37] H. Goldstein, C. Poole, J. Safko, *Classical Mechanics*, Addison-Wesley series in physics, Addison Wesley, 2002, URL: <https://books.google.com.br/books?id=EE-wQgAACAAJ>.
- [38] L. A. Aguirre, *Controle de Sistemas Amostrados*, E-papers Serviços Editoriais Ltda., 2010.
- [39] R. C. Monzani, *Controladores Analógicos e Digitais - uma análise comparativa*, Graduação, Universidade Estadual de Londrina, Londrina, 2010.
- [40] M. S. Fadali, A. Visioli, *Digital control engineering: analysis and design*, Academic Press, 2013.



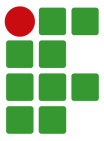
ANEXO I



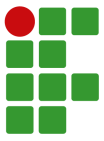


ANEXO II

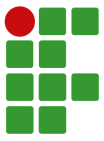
```
1  /* USER CODE BEGIN Header */
2  /**
3   * *****
4   * @file           : main.c
5   * @brief          : Main program body
6   * *****
7   * @attention
8   *
9   * Copyright (c) 2022 STMicroelectronics.
10  * All rights reserved.
11  *
12  * This software is licensed under terms that can be found in the LICENSE file
13  * in the root directory of this software component.
14  * If no LICENSE file comes with this software, it is provided AS-IS.
15  *
16  * *****
17  */
18  /* USER CODE END Header */
19  /* Includes -----*/
20  #include "main.h"
21
22  /* Private includes -----*/
23  /* USER CODE BEGIN Includes */
24  #include <string.h>
25  #include <stdio.h>
26  #include "math.h"
27  /* USER CODE END Includes */
28
29  /* Private typedef -----*/
30  /* USER CODE BEGIN PTD */
31
32  /* USER CODE END PTD */
33
34  /* Private define -----*/
35  /* USER CODE BEGIN PD */
36  /* USER CODE END PD */
37
38  /* Private macro -----*/
39  /* USER CODE BEGIN PM */
40
41  /* USER CODE END PM */
42
43  /* Private variables -----*/
44  ADC_HandleTypeDef hadcl;
45
46  TIM_HandleTypeDef htim1;
47  TIM_HandleTypeDef htim2;
48  TIM_HandleTypeDef htim3;
49  TIM_HandleTypeDef htim4;
50
51  /* USER CODE BEGIN PV */
52
53  /* USER CODE END PV */
54
55  /* Private function prototypes -----*/
56  void SystemClock_Config(void);
57  static void MX_GPIO_Init(void);
58  static void MX_TIM3_Init(void);
59  static void MX_TIM4_Init(void);
60  static void MX_TIM1_Init(void);
61  static void MX_TIM2_Init(void);
62  static void MX_ADC1_Init(void);
63  /* USER CODE BEGIN PFP */
64
65  /* USER CODE END PFP */
66
67  /* Private user code -----*/
68  /* USER CODE BEGIN 0 */
69
70  /*----- Variáveis da leitura dos encoders
71  -----*/
72  uint32_t counterM1 = 0; // Variável de aquisição em bits sem sinal do encoder do
```



```
Motor 1
73 int16_t countM1 = 0; // Variável de aquisição em bits com sinal do encoder do Motor 1
74 int16_t positionM1 = 0; // Variável de aquisição de posição do encoder do Motor 1
75 int16_t positionM1_ant = 0;
76
77 uint32_t counterM2 = 0; // Variável de aquisição em bits sem sinal do encoder do
Motor 2
78 int16_t countM2 = 0; // Variável de aquisição em bits com sinal do encoder do Motor 2
79 int16_t positionM2 = 0; // Variável de aquisição de posição do encoder do Motor 2
80 int16_t positionM2_ant = 0;
81
82 /*-----*/
83
84 /*----- Constantes do sistema
-----*/
85
86 volatile double k = 7.598;
87 volatile double r = 0.03;
88 volatile double sigma = 0.1897;
89
90 /*-----*/
91
92 /*----- Variáveis do controlador
-----*/
93
94 volatile double Sref = -0.00259441308; // Declaração da constante de referência da
rigidez
95
96 volatile double Vmax = 5; // Declaração da constante de tensão máxima do sinal de
controle
97
98 volatile double ts = 0.0001; // Declaração da constante de tempo de amostragem
99
100 //// Declaração das variáveis de erro entre a saída e o medido ////
101 volatile double erro_1 = 0.0;
102 volatile double erro_ant_1 = 0.0;
103
104 volatile double erro_2 = 0.0;
105 volatile double erro_ant_2 = 0.0;
106
107 volatile double erro_s = 0.0;
108 volatile double erro_ant_s = 0.0;
109 ///////////////////////////////////////////////////////////////////
110
111 /////// Declaração das variáveis do sinal de controle ///////
112 volatile double u_1 = 0.0;
113 volatile double u1_1 = 0.0;
114 volatile double u1_ant_1 = 0.0;
115 volatile double u2_1 = 0.0;
116 volatile double u3_1 = 0;
117
118 volatile double u_2 = 0.0;
119 volatile double u1_2 = 0.0;
120 volatile double u1_ant_2 = 0.0;
121 volatile double u2_2 = 0.0;
122 volatile double u3_2 = 0;
123
124 volatile double u_s = 0.0;
125 volatile double u1_s = 0.0;
126 volatile double u1_ant_s = 0.0;
127 volatile double u2_s = 0.0;
128 volatile double u3_s = -0.00259441308;
129 ///////////////////////////////////////////////////////////////////
130
131 volatile double upwm_1 = 0.0; // Declaração da variável de valor absoluto para o PWM
do motor 1
132 volatile double upwm_2 = 0.0; // Declaração da variável de valor absoluto para o PWM
do motor 2
133
134 /////// Declaração dos parâmetros do controlador PID ///////
135 volatile double Kp = 10;
```

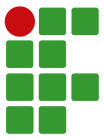


```
136 volatile double Ti = 0;
137 volatile double Td = 100;
138 //volatile double Tn = Td/11.5;
139 volatile double Tn = 0;
140
141 volatile double Kp_s = 3;
142 volatile double Ti_s = 2;
143 volatile double Td_s = 0;
144 volatile double Tn_s = 0;
145 ///////////////////////////////////////////////////////////////////
146
147 /*-----*/
148
149 /*----- Variáveis de conversão da rigidez
150 -----*/
151
152 volatile double SP = 0;
153 volatile double PS = -0.00259441308;
154 volatile double PS_ant = -0.00259441308;
155
156 /*-----*/
157
158 /*----- Variáveis da posição da Junta
159 -----*/
160
161 volatile double positionJ = 0;
162 volatile double positionJ_ant = 0;
163
164 /*-----*/
165
166 /*----- Variáveis do filtro digital
167 -----*/
168
169 volatile double a = 1;
170 volatile double b = 1;
171 volatile double positionJf = 0.0;
172
173 /*-----*/
174
175 /* USER CODE END 0 */
176
177 /**
178  * @brief The application entry point.
179  * @retval int
180  */
181 int main(void)
182 {
183     /* USER CODE BEGIN 1 */
184
185     /* MCU Configuration-----*/
186
187     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
188     HAL_Init();
189
190     /* USER CODE BEGIN Init */
191
192     /* USER CODE END Init */
193
194     /* Configure the system clock */
195     SystemClock_Config();
196
197     /* USER CODE BEGIN SysInit */
198
199     /* USER CODE END SysInit */
200
201     /* Initialize all configured peripherals */
```

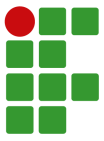


```
202     MX_GPIO_Init();
203     MX_TIM3_Init();
204     MX_TIM4_Init();
205     MX_TIM1_Init();
206     MX_TIM2_Init();
207     MX_ADC1_Init();
208     /* USER CODE BEGIN 2 */
209
210     // MX_NVIC_Init(); // Inicialização das Interrupções (NECESSÁRIO??? - Verificar se é
211     // necessário setar prioridade)
212
213     HAL_TIM_Base_Start_IT(&htim2); // Inicialização da Interrupção do Timer2
214
215     HAL_TIM_PWM_Start (&htim1, TIM_CHANNEL_2); // Inicialização do Timer1 - Canal 2 em
216     modo PWM
217     HAL_TIM_PWM_Start (&htim1, TIM_CHANNEL_3); // Inicialização do Timer1 - Canal 3 em
218     modo PWM
219
220     HAL_TIM_Encoder_Start_IT(&htim3, TIM_CHANNEL_ALL); // Inicialização do Timer3 no
221     modo Encoder
222     HAL_TIM_Encoder_Start_IT(&htim4, TIM_CHANNEL_ALL); // Inicialização do Timer4 no
223     modo Encoder
224
225     /* USER CODE END 2 */
226
227     /* Infinite loop */
228     /* USER CODE BEGIN WHILE */
229     while (1)
230     {
231         /* USER CODE END WHILE */
232
233         /* USER CODE BEGIN 3 */
234         /*----- Leitura dos encoders
235         -----*/
236
237         counterM1 = htim3.Instance->CNT; // Contagem em 32 bits
238         countM1 = (int16_t)counterM1; // Contagem em 16 bits, para ter o sinal de
239         rotação negativo
240         positionM1 = countM1/4; // Conversão para graus de acordo com o valor do
241         timer utilizado para a variável counter
242
243         counterM2 = htim4.Instance->CNT; // Contagem em 32 bits
244         countM2 = (int16_t)counterM2; // Contagem em 16 bits, para ter o sinal de
245         rotação negativo
246         positionM2 = countM2/4; // Conversão para graus de acordo com o valor do
247         timer utilizado para a variável counter
248
249
250         /*-----*/
251
252         // Aquisição da porta ADC (Posição da Junta) //
253         HAL_ADC_Start(&hadc1);
254         HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
255         positionJ = HAL_ADC_GetValue(&hadc1);
256         positionJ = positionJ - 3036;
257         positionJ = -positionJ * 0.9;
258         //
259
260         // Filtro digital //
261         positionJf = (b * positionJ - a * positionJf) * 0.1 + positionJf;
262         positionJ_ant = positionJf;
263         //
264
265         // MUDANÇA DE REFERÊNCIA //
266         if(HAL_GPIO_ReadPin(REF_GPIO_Port,REF_Pin) == GPIO_PIN_SET){
267             Sref = -2*k*r*r*sigma;
268         }
269         else if (HAL_GPIO_ReadPin(REF_GPIO_Port,REF_Pin) == GPIO_PIN_RESET){
270             Sref = -2*k*r*r*sigma*1.045;
271         }
272         //

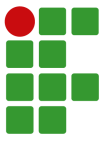
```



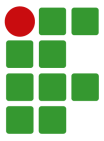
```
263     }
264     /* USER CODE END 3 */
265 }
266
267
268 /*----- Interrupção para Controle
-----*/
269
270 void TIM2_IRQHandler(void)
271 {
272
273     //////////// Conversões entre saídas e rigidez ////////////
274     PS = (-k*r*r*sigma) * (exp(r*sigma*(positionJf + positionM2)) + exp(r*sigma*(
positionM1-positionJf))); // Não linear
275     ////////////
276
277
278     //////////// Cálculo do sinal de controle do PID (rigidez) ////////////
279     erro_s = Sref - PS;
280
281     u1_s = u1_ant_s + (Kp_s*(1 + (ts/Ti_s))*erro_s) - (Kp_s*erro_ant_s);
282     //u2_s = Kp_s*(Td_s/ts)*(PS - PS_ant);
283
284     u_s = u1_s - u2_s + u3_s;
285     ////////////
286
287     //////////// Conversões entre rigidez e saídas (linearizado) ////////////
288     SP = (-u_s - (2*k*r*r*sigma))/(k*r*r*r*sigma*sigma); // Linear
289     SP = SP/2; // alpha + beta / 2
290     ////////////
291
292     //////////// Cálculo do sinal de controle do PID (motores) ////////////
293     erro_1 = SP - positionM1;
294
295     //u1_1 = u1_ant_1 + (Kp*(1 + (ts/Ti))*erro_1) - (Kp*erro_ant_1);
296     u1_1 = u1_ant_1 + (Kp*(1)*erro_1) - (Kp*erro_ant_1);
297     u2_1 = Kp*(Td/ts)*(positionM1 - positionM1_ant);
298
299     u_1 = u1_1 - u2_1 + u3_1;
300
301
302     erro_2 = SP - positionM2;
303
304     //u1_2 = u1_ant_2 + (Kp*(1 + (ts/Ti))*erro_2) - (Kp*erro_ant_2);
305     u1_2 = u1_ant_2 + (Kp*(1)*erro_2) - (Kp*erro_ant_2);
306     u2_2 = Kp*(Td/ts)*(positionM2 - positionM2_ant);
307
308     u_2 = u1_2 - u2_2 + u3_2;
309
310     ////////////
311
312     ///// Atualização de todas as variáveis utilizadas para o controle /////
313     u1_ant_1 = u1_1;
314     erro_ant_1 = erro_1;
315     positionM1_ant = positionM1;
316
317     u1_ant_2 = u1_2;
318     erro_ant_2 = erro_2;
319     positionM2_ant = positionM2;
320
321     u1_ant_s = u1_s;
322     erro_ant_s = erro_s;
323     PS_ant = PS;
324     ////////////
325
326
327
328     //////////// Direção de rotação dos motores ////////////
329     if (u_1 > 0){
330     HAL_GPIO_WritePin(DIRECAO1M1_GPIO_Port, DIRECAO1M1_Pin, GPIO_PIN_SET);
331     HAL_GPIO_WritePin(DIRECAO2M1_GPIO_Port, DIRECAO2M1_Pin, GPIO_PIN_RESET);
332     upwm_1 = u_1/Vmax;
333     upwm_1 = upwm_1*10000;
```



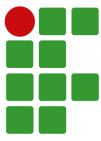
```
334     }
335     if (u_1 < 0){
336     HAL_GPIO_WritePin(DIRECAO2M1_GPIO_Port, DIRECAO2M1_Pin, GPIO_PIN_SET);
337     HAL_GPIO_WritePin(DIRECAO1M1_GPIO_Port, DIRECAO1M1_Pin, GPIO_PIN_RESET);
338     upwm_1 = -u_1/Vmax;
339     upwm_1 = upwm_1*10000;
340     }
341     if (u_1==0){
342     HAL_GPIO_WritePin(DIRECAO2M1_GPIO_Port, DIRECAO2M1_Pin, GPIO_PIN_SET);
343     HAL_GPIO_WritePin(DIRECAO1M1_GPIO_Port, DIRECAO1M1_Pin, GPIO_PIN_SET);
344     upwm_1 = 0;
345     }
346
347     if(u_2>0){
348     HAL_GPIO_WritePin(DIRECAO1M2_GPIO_Port, DIRECAO1M2_Pin, GPIO_PIN_SET);
349     HAL_GPIO_WritePin(DIRECAO2M2_GPIO_Port, DIRECAO2M2_Pin, GPIO_PIN_RESET);
350     upwm_2 = u_2/Vmax;
351     upwm_2 = upwm_2*10000;
352     }
353     if(u_2<0){
354     HAL_GPIO_WritePin(DIRECAO2M2_GPIO_Port, DIRECAO2M2_Pin, GPIO_PIN_SET);
355     HAL_GPIO_WritePin(DIRECAO1M2_GPIO_Port, DIRECAO1M2_Pin, GPIO_PIN_RESET);
356     upwm_2 = -u_2/Vmax;
357     upwm_2 = upwm_2*10000;
358     }
359     if(u_2==0){
360     HAL_GPIO_WritePin(DIRECAO2M2_GPIO_Port, DIRECAO2M2_Pin, GPIO_PIN_SET);
361     HAL_GPIO_WritePin(DIRECAO1M2_GPIO_Port, DIRECAO1M2_Pin, GPIO_PIN_SET);
362     upwm_2 = 0;
363     }
364     ////////////////////////////////////////////////////
365
366     htim1.Instance->CCR2 = upwm_1; // Saída PWM (M1)
367
368     htim1.Instance->CCR3 = upwm_2; // Saída PWM (M2)
369
370
371     HAL_TIM_IRQHandler(&htim2); // Finaliza a interrupção do timer
372
373 }
374
375 /*-----*/
376
377
378 /**
379  * @brief System Clock Configuration
380  * @retval None
381  */
382 void SystemClock_Config(void)
383 {
384     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
385     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
386     RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};
387
388     /** Initializes the RCC Oscillators according to the specified parameters
389     * in the RCC_OscInitTypeDef structure.
390     */
391     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
392     RCC_OscInitStruct.HSEState = RCC_HSE_ON;
393     RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
394     RCC_OscInitStruct.HSIState = RCC_HSI_ON;
395     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
396     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
397     RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
398     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
399     {
400         Error_Handler();
401     }
402
403     /** Initializes the CPU, AHB and APB buses clocks
404     */
405
```



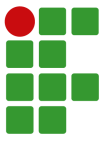
```
406     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
407                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
408     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
409     RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
410     RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
411     RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
412
413     if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
414     {
415         Error_Handler();
416     }
417     PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_ADC;
418     PeriphClkInit.AdcClockSelection = RCC_ADCCLK2_DIV6;
419     if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
420     {
421         Error_Handler();
422     }
423 }
424
425 /**
426  * @brief ADC1 Initialization Function
427  * @param None
428  * @retval None
429  */
430 static void MX_ADC1_Init(void)
431 {
432
433     /* USER CODE BEGIN ADC1_Init 0 */
434
435     /* USER CODE END ADC1_Init 0 */
436
437     ADC_ChannelConfTypeDef sConfig = {0};
438
439     /* USER CODE BEGIN ADC1_Init 1 */
440
441     /* USER CODE END ADC1_Init 1 */
442
443     /** Common config
444     */
445     hadcl.Instance = ADC1;
446     hadcl.Init.ScanConvMode = ADC_SCAN_DISABLE;
447     hadcl.Init.ContinuousConvMode = DISABLE;
448     hadcl.Init.DiscontinuousConvMode = DISABLE;
449     hadcl.Init.ExternalTrigConv = ADC_SOFTWARE_START;
450     hadcl.Init.DataAlign = ADC_DATAALIGN_RIGHT;
451     hadcl.Init.NbrOfConversion = 1;
452     if (HAL_ADC_Init(&hadcl) != HAL_OK)
453     {
454         Error_Handler();
455     }
456
457     /** Configure Regular Channel
458     */
459     sConfig.Channel = ADC_CHANNEL_1;
460     sConfig.Rank = ADC_REGULAR_RANK_1;
461     sConfig.SamplingTime = ADC_SAMPLETIME_239CYCLES_5;
462     if (HAL_ADC_ConfigChannel(&hadcl, &sConfig) != HAL_OK)
463     {
464         Error_Handler();
465     }
466     /* USER CODE BEGIN ADC1_Init 2 */
467
468     /* USER CODE END ADC1_Init 2 */
469
470 }
471
472 /**
473  * @brief TIM1 Initialization Function
474  * @param None
475  * @retval None
476  */
477 static void MX_TIM1_Init(void)
478 {
```



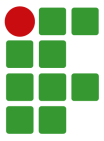
```
479
480     /* USER CODE BEGIN TIM1_Init 0 */
481
482     /* USER CODE END TIM1_Init 0 */
483
484     TIM_ClockConfigTypeDef sClockSourceConfig = {0};
485     TIM_MasterConfigTypeDef sMasterConfig = {0};
486     TIM_OC_InitTypeDef sConfigOC = {0};
487     TIM_BreakDeadTimeConfigTypeDef sBreakDeadTimeConfig = {0};
488
489     /* USER CODE BEGIN TIM1_Init 1 */
490
491     /* USER CODE END TIM1_Init 1 */
492     htim1.Instance = TIM1;
493     htim1.Init.Prescaler = 71;
494     htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
495     htim1.Init.Period = 9999;
496     htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
497     htim1.Init.RepetitionCounter = 0;
498     htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
499     if (HAL_TIM_Base_Init(&htim1) != HAL_OK)
500     {
501         Error_Handler();
502     }
503     sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
504     if (HAL_TIM_ConfigClockSource(&htim1, &sClockSourceConfig) != HAL_OK)
505     {
506         Error_Handler();
507     }
508     if (HAL_TIM_PWM_Init(&htim1) != HAL_OK)
509     {
510         Error_Handler();
511     }
512     sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
513     sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
514     if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
515     {
516         Error_Handler();
517     }
518     sConfigOC.OCMode = TIM_OCMODE_PWM1;
519     sConfigOC.Pulse = 0;
520     sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
521     sConfigOC.OCNPolarity = TIM_OCNPOLARITY_HIGH;
522     sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
523     sConfigOC.OCIdleState = TIM_OCIDLESTATE_RESET;
524     sConfigOC.OCNIdleState = TIM_OCNIDLESTATE_RESET;
525     if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_2) != HAL_OK)
526     {
527         Error_Handler();
528     }
529     if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_3) != HAL_OK)
530     {
531         Error_Handler();
532     }
533     sBreakDeadTimeConfig.OffStateRunMode = TIM_OSSR_DISABLE;
534     sBreakDeadTimeConfig.OffStateIDLEMode = TIM_OSSI_DISABLE;
535     sBreakDeadTimeConfig.LockLevel = TIM_LOCKLEVEL_OFF;
536     sBreakDeadTimeConfig.DeadTime = 0;
537     sBreakDeadTimeConfig.BreakState = TIM_BREAK_DISABLE;
538     sBreakDeadTimeConfig.BreakPolarity = TIM_BREAKPOLARITY_HIGH;
539     sBreakDeadTimeConfig.AutomaticOutput = TIM_AUTOMATICOUTPUT_DISABLE;
540     if (HAL_TIMEx_ConfigBreakDeadTime(&htim1, &sBreakDeadTimeConfig) != HAL_OK)
541     {
542         Error_Handler();
543     }
544     /* USER CODE BEGIN TIM1_Init 2 */
545
546     /* USER CODE END TIM1_Init 2 */
547     HAL_TIM_MspPostInit(&htim1);
548
549 }
550
551 /**
```



```
552     * @brief TIM2 Initialization Function
553     * @param None
554     * @retval None
555     */
556 static void MX_TIM2_Init(void)
557 {
558
559     /* USER CODE BEGIN TIM2_Init 0 */
560
561     /* USER CODE END TIM2_Init 0 */
562
563     TIM_ClockConfigTypeDef sClockSourceConfig = {0};
564     TIM_MasterConfigTypeDef sMasterConfig = {0};
565
566     /* USER CODE BEGIN TIM2_Init 1 */
567
568     /* USER CODE END TIM2_Init 1 */
569     htim2.Instance = TIM2;
570     htim2.Init.Prescaler = 71;
571     htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
572     htim2.Init.Period = 100-1;
573     htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
574     htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
575     if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
576     {
577         Error_Handler();
578     }
579     sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
580     if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
581     {
582         Error_Handler();
583     }
584     sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
585     sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
586     if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
587     {
588         Error_Handler();
589     }
590     /* USER CODE BEGIN TIM2_Init 2 */
591
592     /* USER CODE END TIM2_Init 2 */
593
594 }
595
596 /**
597  * @brief TIM3 Initialization Function
598  * @param None
599  * @retval None
600  */
601 static void MX_TIM3_Init(void)
602 {
603
604     /* USER CODE BEGIN TIM3_Init 0 */
605
606     /* USER CODE END TIM3_Init 0 */
607
608     TIM_Encoder_InitTypeDef sConfig = {0};
609     TIM_MasterConfigTypeDef sMasterConfig = {0};
610
611     /* USER CODE BEGIN TIM3_Init 1 */
612
613     /* USER CODE END TIM3_Init 1 */
614     htim3.Instance = TIM3;
615     htim3.Init.Prescaler = 0;
616     htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
617     htim3.Init.Period = 65535;
618     htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
619     htim3.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
620     sConfig.EncoderMode = TIM_ENCODERMODE_TI12;
621     sConfig.IC1Polarity = TIM_ICPOLARITY_FALLING;
622     sConfig.IC1Selection = TIM_ICSELECTION_DIRECTTI;
623     sConfig.IC1Prescaler = TIM_ICPSC_DIV1;
624     sConfig.IC1Filter = 0;
```



```
625     sConfig.IC2Polarity = TIM_ICPOLARITY_FALLING;
626     sConfig.IC2Selection = TIM_ICSELECTION_DIRECTTI;
627     sConfig.IC2Prescaler = TIM_ICPSC_DIV1;
628     sConfig.IC2Filter = 0;
629     if (HAL_TIM_Encoder_Init(&htim3, &sConfig) != HAL_OK)
630     {
631         Error_Handler();
632     }
633     sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
634     sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
635     if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
636     {
637         Error_Handler();
638     }
639     /* USER CODE BEGIN TIM3_Init 2 */
640
641     /* USER CODE END TIM3_Init 2 */
642
643 }
644
645 /**
646  * @brief TIM4 Initialization Function
647  * @param None
648  * @retval None
649  */
650 static void MX_TIM4_Init(void)
651 {
652     /* USER CODE BEGIN TIM4_Init 0 */
653
654     /* USER CODE END TIM4_Init 0 */
655
656     TIM_Encoder_InitTypeDef sConfig = {0};
657     TIM_MasterConfigTypeDef sMasterConfig = {0};
658
659     /* USER CODE BEGIN TIM4_Init 1 */
660
661     /* USER CODE END TIM4_Init 1 */
662     htim4.Instance = TIM4;
663     htim4.Init.Prescaler = 0;
664     htim4.Init.CounterMode = TIM_COUNTERMODE_UP;
665     htim4.Init.Period = 65535;
666     htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
667     htim4.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
668     sConfig.EncoderMode = TIM_ENCODERMODE_TI12;
669     sConfig.IC1Polarity = TIM_ICPOLARITY_FALLING;
670     sConfig.IC1Selection = TIM_ICSELECTION_DIRECTTI;
671     sConfig.IC1Prescaler = TIM_ICPSC_DIV1;
672     sConfig.IC1Filter = 0;
673     sConfig.IC2Polarity = TIM_ICPOLARITY_FALLING;
674     sConfig.IC2Selection = TIM_ICSELECTION_DIRECTTI;
675     sConfig.IC2Prescaler = TIM_ICPSC_DIV1;
676     sConfig.IC2Filter = 0;
677     if (HAL_TIM_Encoder_Init(&htim4, &sConfig) != HAL_OK)
678     {
679         Error_Handler();
680     }
681     sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
682     sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
683     if (HAL_TIMEx_MasterConfigSynchronization(&htim4, &sMasterConfig) != HAL_OK)
684     {
685         Error_Handler();
686     }
687     /* USER CODE BEGIN TIM4_Init 2 */
688
689     /* USER CODE END TIM4_Init 2 */
690
691 }
692
693
694 /**
695  * @brief GPIO Initialization Function
696  * @param None
697  * @retval None
```



```
698     */
699     static void MX_GPIO_Init(void)
700     {
701         GPIO_InitTypeDef GPIO_InitStructure = {0};
702
703         /* GPIO Ports Clock Enable */
704         __HAL_RCC_GPIOC_CLK_ENABLE();
705         __HAL_RCC_GPIOA_CLK_ENABLE();
706         __HAL_RCC_GPIOB_CLK_ENABLE();
707
708         /*Configure GPIO pin Output Level */
709         HAL_GPIO_WritePin(GPIOB, DIRECAO1M1_Pin|DIRECAO2M1_Pin|DIRECAO1M2_Pin|DIRECAO2M2_Pin
710         , GPIO_PIN_RESET);
711
712         /*Configure GPIO pins : DIRECAO1M1_Pin DIRECAO2M1_Pin DIRECAO1M2_Pin DIRECAO2M2_Pin
713         */
714         GPIO_InitStructure.Pin = DIRECAO1M1_Pin|DIRECAO2M1_Pin|DIRECAO1M2_Pin|DIRECAO2M2_Pin;
715         GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
716         GPIO_InitStructure.Pull = GPIO_NOPULL;
717         GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
718         HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);
719
720         /*Configure GPIO pin : REF_Pin */
721         GPIO_InitStructure.Pin = REF_Pin;
722         GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
723         GPIO_InitStructure.Pull = GPIO_NOPULL;
724         HAL_GPIO_Init(REF_GPIO_Port, &GPIO_InitStructure);
725     }
726     /* USER CODE BEGIN 4 */
727
728     /* USER CODE END 4 */
729
730     /**
731     * @brief This function is executed in case of error occurrence.
732     * @retval None
733     */
734     void Error_Handler(void)
735     {
736         /* USER CODE BEGIN Error_Handler_Debug */
737         /* User can add his own implementation to report the HAL error return state */
738         __disable_irq();
739         while (1)
740         {
741         }
742         /* USER CODE END Error_Handler_Debug */
743     }
744
745     #ifdef USE_FULL_ASSERT
746     /**
747     * @brief Reports the name of the source file and the source line number
748     * where the assert_param error has occurred.
749     * @param file: pointer to the source file name
750     * @param line: assert_param error line source number
751     * @retval None
752     */
753     void assert_failed(uint8_t *file, uint32_t line)
754     {
755         /* USER CODE BEGIN 6 */
756         /* User can add his own implementation to report the file name and line number,
757         ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
758         /* USER CODE END 6 */
759     }
760     #endif /* USE_FULL_ASSERT */
761
```

OSCAR HENRIQUE REICHOW

**PROJETO E IMPLEMENTAÇÃO DE UMA JUNTA ROBÓTICA EM CONFIGURAÇÃO
ANTAGÔNICA COM CONTROLE DE IMPEDÂNCIA**

Este trabalho foi julgado adequado para obtenção do título em Bacharel em Engenharia Elétrica, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Jaraguá do Sul, 01 de Julho de 2022.



Prof. Dr. Rodrigo Trentini Preuss
Orientador
IFSC – Campus Jaraguá do Sul – Rau



Documento assinado digitalmente
GABRIEL HERMANN NEGRI
Data: 22/07/2022 20:29:04-0300
Verifique em <https://verificador.itl.br>

Prof. Dr. Gabriel Hermann Negri
IFSC – Campus Caçador

Prof. Dr. Alexandre Campos
UDESC – Campus Balneário Camboriú



Assinaturas do documento



Código para verificação: **G02VT04H**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:



ANIBAL ALEXANDRE CAMPOS BONILLA (CPF: 008.XXX.599-XX) em 21/07/2022 às 12:50:03

Emitido por: "SGP-e", emitido em 30/03/2018 - 12:34:23 e válido até 30/03/2118 - 12:34:23.

(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTIwMjJfMDAwMzE3NzJfMzE4MjRfMjAyMI9HMDJWVDA0SA==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00031772/2022** e o código **G02VT04H** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.