

INMOOV-URI: DINÂMICA DE CONTROLE DOS SERVOMOTORES NA REALIZAÇÃO DE MOVIMENTOS SÍNCRONOS EM ROBÔ HUMANOIDE

INMOOV-URI: SERVOMOTORS CONTROL DYNAMICS WHEN PERFORMING SYNCHRONOUS MOVEMENTS IN A HUMANOIDE ROBOT

ALEXANDRE DOS SANTOS ROQUE

Universidade Regional Integrada do Alto Uruguai e das Missões, URI, Santo Ângelo, RS, Brasil
Doutor em Engenharia Elétrica. E-mail: roque@santoangelo.uri.br
<https://orcid.org/0000-0001-5422-5414>

BRUNO KUHN FERNANDES

Universidade Regional Integrada do Alto Uruguai e das Missões, URI, Santo Ângelo, RS, Brasil
Graduando em Engenharia Elétrica. E-mail: brunokfernandes@aluno.santoangelo.uri.br
<https://orcid.org/0009-0000-5005-7361>

Submissão: 13-01-2025 - Aceite: 01-10-2025

RESUMO: Este artigo apresenta o desenvolvimento do robô humanoide InMoov-URI, um projeto baseado em impressão 3D em constante aprimoramento. O foco deste estudo é a dinâmica de controle de servomotores para movimentos síncronos, explorando o framework Robot Operating System, uma ferramenta consolidada na robótica para gerenciar a comunicação entre processos por meio de troca de mensagens. Como metodologia, foi adotado o modelo de controle em malha fechada, beneficiando-se do fato de os motores utilizados já possuírem circuitos integrados, que permitem o feedback de posicionamento, bem como também a adição de sensores de corrente para monitoramento. Os testes foram implementados na linguagem de programação Python, com o framework embarcado em uma plataforma Raspberry Pi, permitindo a evolução do sistema de controle. Os resultados demonstram a eficácia do controle dos servomotores na execução de gestos programados, além de fornecerem dados cruciais sobre o consumo de corrente dos atuadores, essenciais para o dimensionamento adequado da eletrônica de potência. Estes resultados consolidam o robô InMoov-URI como um recurso valioso para o ensino e a pesquisa nas áreas de engenharia e computação em ambiente universitário.

PALAVRAS-CHAVE: Técnicas de controle. Servomotores. Robótica. Framework ROS.

ABSTRACT: This paper presents the development of the InMoov-URI humanoid robot, a 3D-printing-based project undergoing continuous improvement. The focus of this study is the control dynamics of servomotors for synchronous movements, leveraging the Robot Operating System framework, a consolidated tool in robotics for managing inter-process communication through message exchange. The adopted methodology employs a closed-



Esta obra está licenciada com uma Licença Creative Commons
Atribuição-NãoComercial-SemDerivações 4.0 Internacional.

loop control model, benefiting from the integrated circuits within the motors that provide positioning feedback, complemented by the addition of current sensors for monitoring. Tests were implemented using the Python programming language, with the framework embedded in a Raspberry Pi platform, which enables the continuous evolution of the control system. The results demonstrate the effectiveness of servomotor control in executing programmed gestures and provide crucial data on the actuators' current consumption, essential for the proper dimensioning of power electronics. These findings consolidate the InMoov-URI robot as a valuable resource for teaching and research in engineering and computer science within a university setting.

KEYWORDS: Control techniques. Servo motors. Robotics. ROS Framework.

Introdução

Estudos na área de robôs humanoides foram alavancados pela Honda em 1996 com o anúncio do desenvolvimento de um robô que chamou muito a atenção das pessoas (HIRAI, 1998). Mais tarde, após melhoramentos, esse robô ficou conhecido como Asimo, o qual já possui diferentes versões e atualizações. Desde então, estamos presenciando a ascensão dos diversos tipos de robôs humanoides, como o HERMES desenvolvido no MIT (BISCHOFF, 2003), o NASA R5 Valkyrie (RADFORD, 2015), o Toyota T-HR3 (NEWSROOM, 2017), entre tantos outros que estão surgindo.

No entanto, para democratizar o acesso a essa tecnologia, são necessários investimentos de grandes empresas e governos em robôs humanoides de serviço, os quais podem ser usados em diversas áreas, e em muitas outras tarefas críticas para o ser humano. Assim, o uso de robôs humanoides em atividades de ensino e pesquisa depende muito mais do empenho de instituições ou projetos voltados a problemas específicos do usuário (DOS SANTOS *et al*, 2024; OKAGBUE *et al*, 2024).

Contribuindo para a adoção de robôs humanoides, a manufatura aditiva com impressão 3D tem sido de fundamental importância para o projeto e prototipação de diversas peças, de forma rápida e com baixo custo. Esta adoção impulsiona muitas discussões e investimentos no âmbito acadêmico e empresarial, abrindo um leque de oportunidades em diversas áreas do conhecimento. Na robótica não é diferente, pois a prototipação de peças singulares é facilmente realizada com o uso dessa tecnologia. O baixo custo de compra do material de impressão (PLA - um tipo de polímero) e alto rendimento possibilitam que universidades o usem de diversas maneiras, desenvolvendo polias, rodas, engrenagens e diversas partes de equipamentos robóticos.

Para corroborar nesta direção, destaca-se a utilização da manufatura aditiva no processo de montagem de um robô humanoide, baseado no projeto livre *InMoov*, desenvolvido pelo francês Gael Langevin (LANGEVIN, 2019). Assim, esta pesquisa foca no desenvolvimento de uma análise específica sobre a dinâmica de controle dos servomotores e na adoção de um *framework* para possibilitar a realização de movimentos síncronos de diferentes partes do robô. No atual estágio destaca-se a realização de movimentos da mão e do braço direito, em conjunto com a cabeça do robô humanoide. Assim, é impulsionado o desenvolvimento de trabalhos futuros na área de controle e inteligência artificial, principalmente na questão de interatividade, sem a necessidade da recriação e adaptação de todo este mecanismo de atuação. Por conseguinte,

pretende-se criar previamente alguns movimentos para exemplificar o funcionamento do sistema para possíveis pesquisadores que tenham interesse em fazer o uso da plataforma robótica.

Adicionalmente, esta pesquisa destaca a integração de diferentes áreas do conhecimento: a engenharia elétrica, a engenharia mecânica e a computação, com recursos de programação que possibilitam agregar inteligência aos robôs. Aliando conceitos de programação à teoria de sistemas de controle, permite-se a evolução constante de sistemas robóticos, principalmente na área de humanoides. A base fundamental deste estudo é a integração de circuitos eletrônicos e sistemas embarcados (para permitir a programação de funcionalidades do robô) com sistemas sensoriais e sistemas de atuação (servomotores), os quais são responsáveis pela realização de movimentos do conjunto mecânico.

Como objetivo central, destaca-se a continuidade da montagem do humanoide InMoov-URI desenvolvida nos projetos anteriores, integrando o *framework* ROS à plataforma, para a movimentação conjunta dos atuadores servomotores. Em conjunto, são estudados tipos de hardware que podem ser utilizados, tipos de atuadores, sensores e técnicas de controle, para assim possibilitar o desenvolvimento de aplicações com o robô humanoide. Especificamente, são destacados exemplos de uma aplicação prática do *framework* ROS e suas adequações para a plataforma InMoov-URI, com testes funcionais e análises de resultados preliminares.

Enfatizando as contribuições, esta pesquisa se caracteriza como uma alternativa de plataforma robótica, na categoria de humanoides, para o desenvolvimento perene de diversas pesquisas multidisciplinares no âmbito do ensino de graduação, tendo como estudo de caso central a interação com o público em geral. Soluções de problemas relacionados à montagem, aos tipos de atuadores, aos tipos de sistemas de controle já são tratadas e testadas com experimentos práticos, mostrando resultados e direcionamentos para a continuidade do projeto.

Este artigo está organizado da seguinte forma: apresentação da base teórica procurando elucidar os principais conceitos envolvidos neste estudo; são destacados os principais objetivos da pesquisa; detalhamento da metodologia de pesquisa aplicada; apresentação dos principais resultados obtidos até o momento; e por fim, uma breve discussão dos principais pontos de relevância acerca dos resultados, enfatizando conclusões e perspectivas de continuidade do estudo.

Base teórica e trabalhos relacionados

A robótica é um campo de estudo multidisciplinar que integra engenharia elétrica, mecânica, de controle e ciência da computação. Seu avanço recente tem sido notável, aproximando cada vez mais a sociedade dos robôs. Nesse contexto, os robôs humanoides, ao exibirem semelhança com os seres humanos, fomentam afinidade e simpatia na interação (MORI, MACDORMAN e KAGEKI, 2012).

Diante deste crescente interesse e desenvolvimento, é fundamental definir as características que distinguem um robô humanoide. Segundo Hernández Ramírez (2016) apud Constantin *et al.* (2015), “Um humanoide é uma espécie de robô que tem aparência e funcionamento como o de um ser humano”. De acordo com a classificação de (HIRAI *et al.*, 1998) e (KANEKO *et al.*, 2008), a Tabela 1 apresenta uma classificação dos tipos de robôs humanoides.

Tabela 1: Classificação dos Robôs Humanoides.

Robôs Humanoides		
Com pernas	Mini	Nao, QRIO, SIGMO
	Pequeno/Médio	REEM, Toyota Partner Robot, Xianxingzhe
	Tamanho Humano	Sophia, Musa, Atlas
	Grande	Elektro, Land Walker
Com rodas	GuRoo, EMIEW, Enon	
Com trilhos	BEAR	
Parte superior	Geoff Peterson, Robonaut, Telenoid R1	

Fonte: (HIRAI et al., 1998) e (KANEKO et al., 2008).

A robótica está em todo lugar e ela vem recebendo uma grande atenção, principalmente em áreas que buscam melhorar a vida do ser humano. Uma área especificamente em destaque é a Robótica Assistiva, que busca fornecer um amparo a pessoas que possuem dificuldade em desempenhar alguma tarefa. Desta forma, o presente estudo também apresenta contribuições neste contexto, com objetivos diversos, e consolidando-se como uma ferramenta de apoio em instituições de ensino.

O InMoov é um projeto baseado em impressão 3D *open source* (projeto livre e aberto), criado em 2012 pelo designer francês Gael Langevin (LANGEVIN, 2019). Segundo o criador, um dos objetivos do projeto é fornecer uma plataforma robótica humanoide acessível a todos, de maneira que os usuários possam personalizar as partes e até mesmo melhorá-las. Desde então, esse projeto tem sido usado no meio acadêmico para a produção de pesquisa científica e experimentos para o aprendizado. A Figura 1 ilustra o robô InMoov original montado por Gael Langevin.

Figura 1 - Plataforma de robótica InMoov



Fonte: Langevin (2019)

Contribuindo na mesma direção, em Homburg *et al.* (2019) é relatado o uso da plataforma InMoov como um tradutor para a linguagem de sinais. Os autores abordam as diferentes opções atuais que existem para esse tipo de mediador de línguas e informam o porquê da escolha do InMoov como plataforma. Uma das razões foi a grande disponibilidade dos componentes individuais que o compõem. Corroborando com o trabalho anteriormente citado, em Acuña (2016), o robô humanoide InMoov é utilizado para ensinar a linguagem de sinais através de imitação. Com a movimentação dos braços e mãos, é possível reproduzir os sinais das Libras de maneira que o aluno possa imitar e, com isso, aprender a se comunicar com essa linguagem. Um estudo da constituição mecânica dos graus de liberdade do robô foi realizado e um algoritmo para o controle dos mesmos foi desenvolvido pelos autores.

Adicionalmente, em Axelsson (2018), a plataforma de robótica InMoov é utilizada para auxílio no aprendizado da língua de sinais para crianças com autismo, utilizando os movimentos do robô para criar gestos a serem replicados pelas crianças submetidas à pesquisa. A pesquisa contribui para a aplicação da plataforma em benefício de pessoas com necessidades especiais e fornece uma estrutura para orientar o desenvolvimento de futuros robôs assistivos.

A plataforma InMoov também foi utilizada em Valencia *et al.* (2016) para a detecção e rastreamento de movimentos através de câmeras instaladas na cabeça do robô. Os autores fizeram o estudo de vários algoritmos de visão computacional e implementaram um algoritmo para ser aplicado no robô, com o desenvolvimento de testes práticos de validação. Outra aplicação da plataforma é apresentada em Parali (2022), onde realizou-se um estudo sobre o controle e sensoriamento dos dedos da mão robótica da plataforma.

Com esta fundamentação, o desenvolvimento de um framework em Python para o projeto InMoov é apresentado por Teixeira (2018), caracterizando um marco inicial deste projeto na Universidade Regional Integrada do Alto Uruguai e das Missões – URI, com a implementação de uma biblioteca de funções para o controle dos movimentos da cabeça do robô. Em seguida, os trabalhos de Ferreira (2019) e Bohnenberger (2020) continuaram o desenvolvimento com a montagem inicial dos braços e do tórax, indicando a importância do uso de um *framework* de controle. Desta forma, o projeto atual vem ajustando e melhorando a montagem física do robô, bem como implementando o *framework* de controle.

Como enfatizado nos estudos anteriores, o desenvolvimento de plataformas robóticas requer a utilização de um sistema de controle dedicado, e o *framework* ROS se demonstra versátil para diversas aplicações neste contexto. Assim, em Mišeikis *et al.* (2020), é apresentado o robô nomeado Lio, uma plataforma robótica personalizada para interação humano-robô capaz de manipular objetos e realizar tarefas de assistência em cuidados pessoais, incluindo operação autônoma com utilização do *framework* ROS. Da mesma forma, em Baek (2022) utilizou-se o *framework* ROS para o desenvolvimento de um robô autônomo para o trabalho no meio agrícola. Já em Mulyanto *et al.* (2020) aplicou-se o *framework* ROS como plataforma principal na criação de um sistema de alerta de colisão para prevenção de acidentes de trânsito. Assim, estas aplicações em áreas distintas demonstram a versatilidade do *framework* ROS e sua aplicabilidade em um projeto de robótica.

Como observado nos trabalhos relacionados e na literatura recente, há a carência de pesquisas com enfoque no uso da plataforma robótica InMoov em conjunto com o *framework* ROS, levando em consideração atividades multidisciplinares e de ensino de robótica. Outra

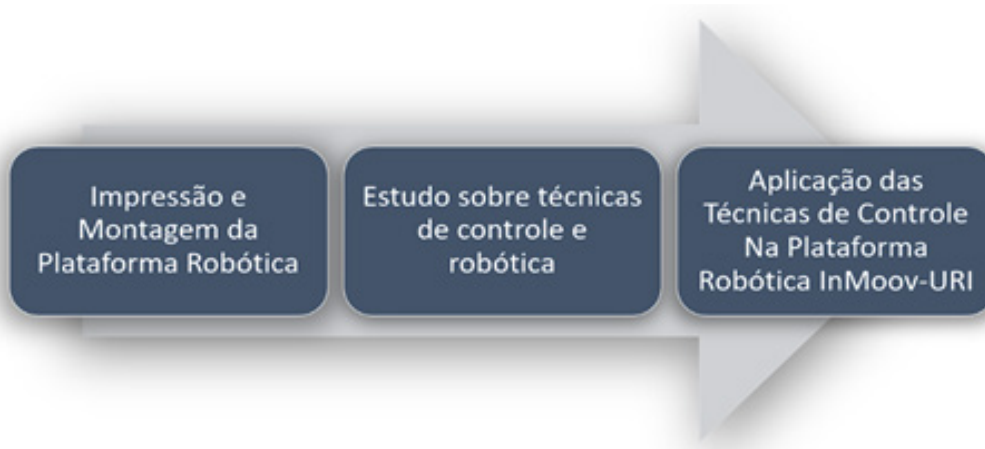
lacuna observada nos estudos diz respeito à pouca documentação referente ao uso do *framework* para controle de servomotores, especificamente considerando ainda a plataforma InMoov. Outra oportunidade de pesquisa encontrada diz respeito à abstração da arquitetura, de modo a permitir o desenvolvimento de diversas aplicações de alto nível com o Robô, e por diferentes áreas. Sendo assim, o projeto nomeado InMoov-URI caracteriza uma importante oportunidade de divulgação de pesquisas que são realizadas na universidade, por meio de seus cursos de computação e engenharia.

Materiais e métodos

Este estudo foi desenvolvido seguindo a metodologia de prototipagem evolutiva, intrínseca ao processo de desenvolvimento e experimentação de produtos (KONTONYA e SOMMERVILLE, 1998; BOX e Hunter, 2005). Segundo esses autores, protótipos executáveis podem ser implementados de duas formas: prototipagem descartável (*throwaway*) e prototipagem evolutiva. Enquanto a prototipagem descartável auxilia na validação dos requisitos e não é utilizada no produto final, a prototipagem evolutiva minimiza o tempo de desenvolvimento, sendo integrada ao produto. Esta metodologia é empregada na verificação do software de controle embarcado e no refinamento contínuo do protótipo do hardware e da estrutura física do robô humanoide. Essa abordagem iterativa está em consonância com metodologias contemporâneas como o *Design Thinking*, que utiliza a prototipagem para ganhos em criatividade e agilidade, incorporando o *feedback* do usuário para modificações e aperfeiçoamentos (SILVA *et al.*, 2024), bem como também com *frameworks* de desenvolvimento ágil, como o *Scrum*, que se baseiam em curtos ciclos de trabalho e aperfeiçoamento contínuo, como exemplificado no desenvolvimento de jogos educativos (SOUZA *et al.*, 2023).

Para a realização dos objetivos do projeto, seguindo a abordagem evolutiva, foi adotada a seguinte sequência de procedimentos metodológicos descritos na Figura 2.

Figura 2 - Sequência de procedimentos metodológicos



Fonte: Dos autores.

Os objetivos centrais da pesquisa são trabalhados em 4 etapas, os quais são apresentados da seguinte forma: a primeira etapa para a impressão das peças 3D, montagem e ajustes do

robô; a segunda etapa focada no estudos atuadores motores e sensores aplicáveis para o robô; a terceira etapa para estudo e aplicação do *framework* ROS na plataforma, incluindo a integração com novas funcionalidades; e por fim, a quarta etapa, para desenvolvimento do algoritmo e uma interface de testes dos movimentos do robô (utilizando as partes já concluídas do projeto).

Em consonância com esta abordagem, Araújo et al. (2017) salientam que o *framework* ROS se caracteriza por potencializar o desenvolvimento de *softwares* compatíveis com as pesquisas em robótica, pois estimula um desenvolvimento colaborativo dos códigos disponíveis. Seu funcionamento é baseado em um modelo de publicações e subscrições ilustrado na Figura 3.

Figura 3 - Modelo publisher/subscriber utilizado no framework ROS



Fonte: Dos autores

Como apresentado na Figura 3, o *publisher* envia uma mensagem para um *subscriber*, esse modelo atua com o processo de *nodes*. Os nomes e registros dos nós são providenciados por um *Master*, que permite a localização dos nós e realiza o monitoramento das publicações e subscrições. A utilização do ROS permite criar nós separados e utilizá-los como módulos para cada tipo de atividade, comunicando-se por intermédio de um tópico e também por requisições de um *service*.

A realização do projeto baseia-se na utilização de múltiplos dispositivos; as especificações de versões e modelos utilizados estão descritas na Tabela 2.

Tabela 2 – Materiais utilizados

Elemento	Descrição
Computador	Ubuntu 20.04 (Sistema Operacional)
Raspberry Pi 3 B+	Ubuntu Server 20.04 (Sistema Operacional)
<i>Framework</i> ROS	Noetic (Versão da distribuição)
Programação	Python 3.11.3(Linguagem de alto nível)
Placa controladora	PCA9685 - Adafruit
Alimentação	5V, 30A (Fonte Chaveada)

Fonte: o autor.

Resultados e discussões

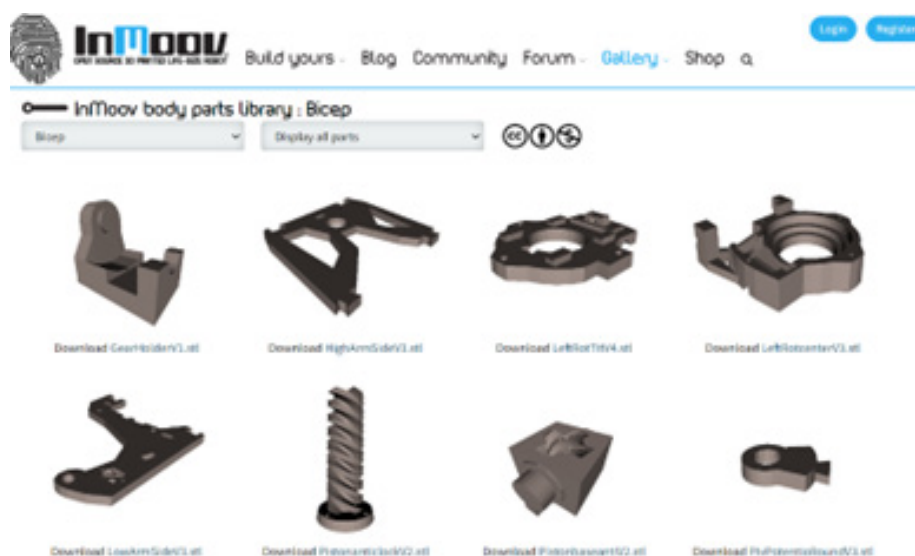
Como descrito na sequência de procedimentos metodológicos, a execução das tarefas dos projetos aborda quatro etapas. A primeira etapa abrange a impressão e montagem da plataforma robótica. A segunda etapa aborda os componentes utilizados no projeto, os atuadores, sensores e dispositivos de controle, que, utilizando esses componentes, se executa a terceira etapa, onde é aplicado o *framework* ROS em conjunto com as técnicas de controle para o funcionamento do projeto. A etapa final é executada para validar os estudos com o ROS, realizando os movimentos do robô e monitorando-os com um sensor de corrente.

Impressão e montagem da plataforma

Durante a primeira parte do projeto, de acordo com a metodologia, foi realizada a impressão das peças e sua montagem. Adicionalmente, foi realizado o estudo do funcionamento da plataforma física e mecânica, disponibilizado no site do projeto original, como pode ser observado na

Figura 4. Todas as peças e procedimentos necessários estão descritos na página *web* <https://inmoov.fr/>.

Figura 4 - Arquivos disponibilizados para a impressão das peças da plataforma InMoov



Fonte: LANGEVIN (2019).

Esse período levou em consideração adaptações e ajustes necessários nas peças, bem como também estudos prévios sobre os atuadores que podem ser usados na plataforma. Como resultado de outros projetos realizados nessa plataforma robótica, apresenta-se em um estágio avançado de sua montagem, como ilustra a Figura 5.

Figura 5 - Plataforma robótica InMoov-URI



Fonte: Dos autores.

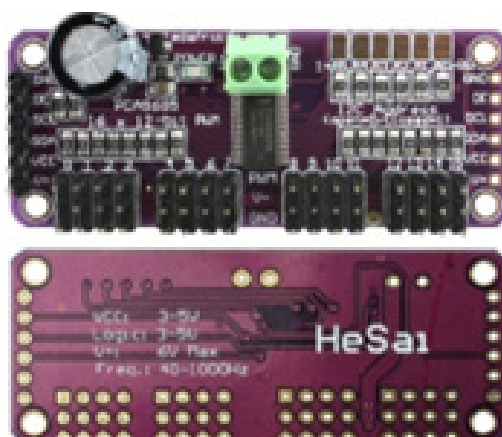
Durante os estudos, diversas atualizações foram realizadas no robô, como a finalização da cintura, que incluiu a adição de um suporte para sua fixação, que permitiu a liberdade nos movimentos do robô, como a movimentação dos braços. Diversos reparos foram realizados, necessários devido à ação do tempo em suas peças plásticas que se tornam frágeis e suscetíveis a quebrar com os esforços realizados pelo robô.

A continuidade da montagem do robô é uma tarefa fundamental para que os estudos realizados sobre a plataforma sejam realizados. Assim, após os ajustes necessários, os esforços serão direcionados para o controle e sensoriamento da plataforma InMoov-URI utilizando o *framework* ROS.

Estudo sobre sensores, atuadores e plataforma de controle

Até o momento, o dispositivo utilizado para acionamento dos servomotores é o módulo PCA 9685. Esse dispositivo possui filtro de ruídos na entrada da porta serial, também controla a saída de corrente, não permite que ocorram *glitches* (falhas que geram pulsos aleatórios que geram aceleração ou folga de posicionamento) na saída para os servos logo após ser ligado. A seguir, a Figura 6 ilustra o dispositivo usado no acionamento dos motores.

Figura 6 - Driver de servomotores PCA9685

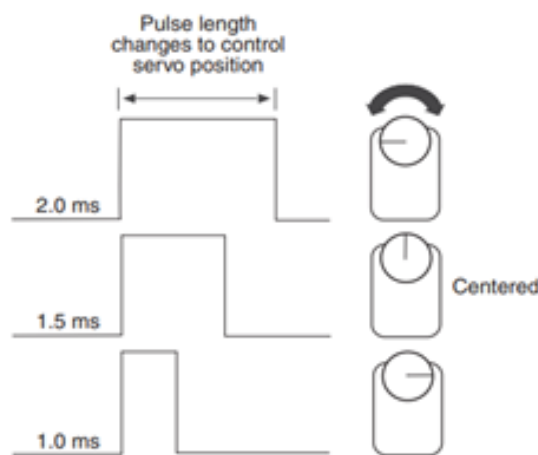


Fonte: Adafruit. Disponível em: <https://www.adafruit.com/product/815>.

A placa extensora PC9685 possui 16 canais de PWM na placa para controlar os servomotores, sendo a solução ideal para a quantidade de atuadores presentes na plataforma de robótica InMoov, é possível a utilização de um Arduino e Raspberry Pi juntamente com a placa extensora, logo esses dispositivos são compatíveis com o Protocolo de comunicação I2C (DO NASCIMENTO e NETO, 2021).

Os servomotores utilizados no robô para movimentação de seus membros possuem ângulo de atuação de 180 graus, são responsáveis pelos esforços mecânicos necessários pela plataforma na execução de gestos. O sinal de controle de um servomotor é realizado através da repetição constante de um pulso, esse pulso deve ser estável e repetido entre 30 a 50 vezes em um segundo, a duração dos pulsos que variam entre 1 a 2 ms é o que determina a posição (MCCOMB, 2011), a variação da duração do pulso e o resultado no ângulo do servomotor está exemplificado na Figura 7.

Figura 7 - Resposta do servomotor a diferentes comprimentos de pulsos.



Fonte: MCCOMB (2011, p. 251).

Conforme a Figura 7, a entrada é enviada a partir de um controle de pulsos, podendo ser utilizado o controle PWM (*Pulse Width Modulation*) para realizar a modulação desse pulso. Sendo o pulso modulado o sinal enviado para o servomotor, este é comparado com o sinal que corresponde à posição atual do eixo do servomotor, caracterizando a malha de controle do servomotor como uma malha fechada. Assim, ocorre uma constante correção do movimento de acordo com o feedback obtido através da variação de resistência do potenciômetro acoplado ao eixo do servomotor. De acordo com Scheffer, et al. (2018, p. 19) um sistema em malha fechada se define por:

Os sistemas de controle em malha fechada, também chamados de sistemas com realimentação, são sistemas influenciados pela saída. Nesses sistemas, a saída é comparada à entrada do sistema, chamada de referência ou sinal desejado, e a diferença entre elas é chamada de erro. O sistema é projetado de forma a manter o erro próximo de zero, ou seja, o sistema mantém a saída no nível desejado.

Neste contexto, para atingir a posição angular desejada no servomotor, ocorre a realimentação do sinal de saída através do potenciômetro acoplado ao eixo do servomotor, e através da comparação da saída (ângulo do servomotor) e da entrada (ângulo a ser atingido)

é obtido o “erro”, o qual é necessário para o sistema controlador atuar para atingir a posição desejada. Quando a entrada e saída são iguais, o erro passa a não existir mais e o servomotor se torna estático até ser enviado um comando diferente ou ser atribuída uma força externa ao atuador. A malha de controle do servomotor é apresentada na Figura 8.

Figura 8 - Malha de controle de um servomotor



Fonte: Dos autores.

Como destacado neste estudo, o controle de malha em sinal fechado é o controle no qual o sinal de saída depende totalmente da ação de controle (FRANKLIN, 2013). Assim, os atuadores motores, para realizarem suas funções, dependem do sistema de controle previamente programado, sendo esse controle ajustado com informações dos sensores (ARAUJO, 2007). Essa característica torna o sistema mais preciso e mais tolerante a perturbações externas, tornando-o mais dinâmico.

A plataforma InMoov apresenta uma grande variação na atuação dos seus movimentos, em que cada tipo de movimento requer um servomotor capaz de suprir o torque necessário. Para obter esse resultado, procura-se utilizar o modelo correto para cada finalidade. Na Tabela 3 apresentam-se os modelos de servomotores utilizados e suas especificações.

Tabela 3 - Especificações dos servomotores utilizados na plataforma

Modelo de servomotor	Torque (Operando em 4,8 volts)	Material das Engrenagens	Peso	Servomotor
MG996R	9,4 kgf·cm	Metal	55 g	
TD-8130MG	29,5 kgf·cm	Metal	56 g	
HITEC HS-805B	19,8 kgf·cm	Nylon	152 g	
SG90	1,2 kgf·cm	Nylon	9 g	

Fonte: Dos autores.

Para desenvolver a programação do conjunto de hardware é necessário um dispositivo embarcado, similar a um computador, porém de pequeno porte. Atualmente existem diversos modelos de plataformas de prototipagem e programação, que permitem o desenvolvimento de um algoritmo de processamento central que efetiva a comunicação e o controle dos servomotores. No presente projeto, para fins de desenvolvimento dos algoritmos de controle, o mini computador de placa única Raspberry Pi 3B+ é utilizado no projeto utilizando o sistema Linux Ubuntu em conjunto com o *framework* ROS. A seguir, a Figura 9 ilustra o mini computador de placa única utilizado no projeto.

Figura 9 - Raspberry Pi 3 Model B+

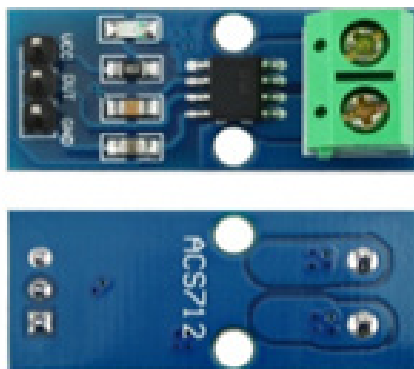


Fonte: Raspberry Pi Foundation. Disponível em: <https://www.raspberrypi.com/products/>.

Esse mini computador contribui para o desenvolvimento de um software que compõe o sistema embarcado de controle do robô, pois possui características de robustez, com um processador *quad-core* de 64 bits com 1,4 GHz de *clock*, LAN sem fio de banda dupla, Bluetooth 4.2 / BLE, Ethernet veloz e suporte Power-over-Ethernet. Com a Raspberry Pi 3 B+, é possível conectar o robô a outros dispositivos e utilizar o ROS para realizar a comunicação entre eles.

O sensor utilizado para realizar a leitura de corrente consumida pelo sistema é o modelo ACS712, apresentado na Figura 10. Este dispositivo utiliza o princípio do efeito *Hall* para detectar o campo magnético gerado por uma corrente elétrica que passa pelo condutor próximo a ele, e retorna em tensão medida em volts um valor proporcional à corrente que passa pela trilha condutora presente no circuito do ACS712.

Figura 10 - Sensor de Corrente ACS712 20A AC/DC com Efeito Hall



Fonte: Usina Info. Disponível em: <https://www.usinainfo.com.br/sensor-de-corrente-arduino/sensor-de-corrente-ac712-20a-ac-dc-com-efeito-hall-5701.html#idTab5>.

O módulo sensor ACS712 é projetado para fornecer uma leitura analógica proporcional à corrente medida. Existem diferentes versões do sensor ACS712 disponíveis, com faixas de corrente diferentes, como 5A, 20A e 30A, que possuem sensibilidade diferente e estão dispostas na Tabela 4 a seguir.

Tabela 4 - Especificações das variações de ACS712-05B

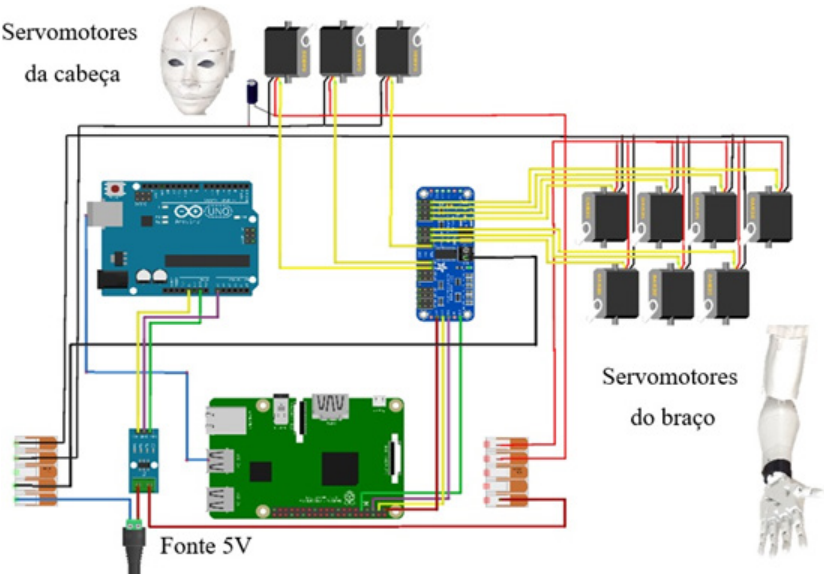
Modelo	Capacidade	Sensibilidade
ACS712-05B	±5A	185 mV/A
ACS712-20B	±20A	100 mV/A
ACS712-30A	±30A	66 mV/A

Fonte: Datasheet do fabricante (ALLEGRO, 2022).

Neste sentido, devido a Raspberry Pi 3 possuir apenas entradas de sinal digital em seu conjunto de pinos GPIO, é necessário utilizar um conversor ADC para a coleta de dados do sensor de corrente ACS712. A utilização do Arduino UNO foi escolhida para realizar essa conversão, por possuir um ADC em sua plataforma. É possível conectar o Arduino UNO à Raspberry Pi por meio do cabo USB e realizar uma conexão serial por meio do protocolo UART e enviar os dados do sensor de corrente. Outra possibilidade é a utilização de um conversor ADC, pois testes demonstram que um conversor de 16 bits como o ADS1115 tem um desempenho superior em relação ao conversor ADC de 10 bits presente no Arduino (CONSTANTIN *et al.*, 2023), o que resulta em leituras mais precisas e coerentes.

As conexões realizadas entre o hardware utilizado na plataforma estão apresentadas em formato de diagrama na Figura 11. O diagrama apresenta a utilização da Raspberry Pi 3 B+ em conjunto com a placa PCA9685 responsável pelo controle dos servomotores da cabeça e do braço do robô, também apresenta a utilização do sensor de corrente ACS712 em união com o Arduino UNO R3 responsável por realizar a conversão da leitura analógica para um sinal digital que é enviado para a Raspberry PI 3 B+.

Figura 11 - Diagrama da arquitetura de interconexão de componentes da plataforma robótica



Fonte: Dos autores.

As conexões da plataforma devem seguir o diagrama de componentes para o funcionamento da plataforma de robótica InMoov-URI. Os servomotores são alimentados diretamente da fonte de 5 volts, onde o sinal positivo é conectado ao sensor de corrente ACS712 de maneira invasiva. Em paralelo com a alimentação, possui um capacitor de 1000 μ F; esse capacitor permite que os servomotores atuem de maneira adequada. Para realizar o controle dos atuadores, os sinais de controle de cada servomotor são conectados à placa PCA9685, a qual utiliza o protocolo de comunicação I2C com a Raspberry Pi 3 B+.

Com esses componentes montados na plataforma física, foi possível iniciar os testes com os servomotores presentes na cabeça e no braço do robô, contemplando realizar movimentos e gestos, além da leitura da curva de corrente elétrica dos atuadores durante a realização de movimentos. Para a realização dos testes, a utilização do *framework* ROS para comunicar os componentes da plataforma é apresentada a seguir.

Algoritmo, Técnica, *Framework* ROS e Interface de controle

Destaca-se que, enquanto sistema de comunicação entre os componentes de um projeto, o ROS suporta diversas ferramentas e aplicações que podem ser estruturadas pelos usuários em uma rede integrada, cuja compreensão exige o entendimento dos conceitos, apresentados a seguir.

Os *nodes* são os processos dentro do *framework* ROS, que, utilizando os métodos de comunicação do ROS, trocam dados e executam as tarefas. Podem utilizar as bibliotecas *ros.h* ou *rospy*, linguagem de programação C++ e Python, respectivamente (LENTIN, 2015).

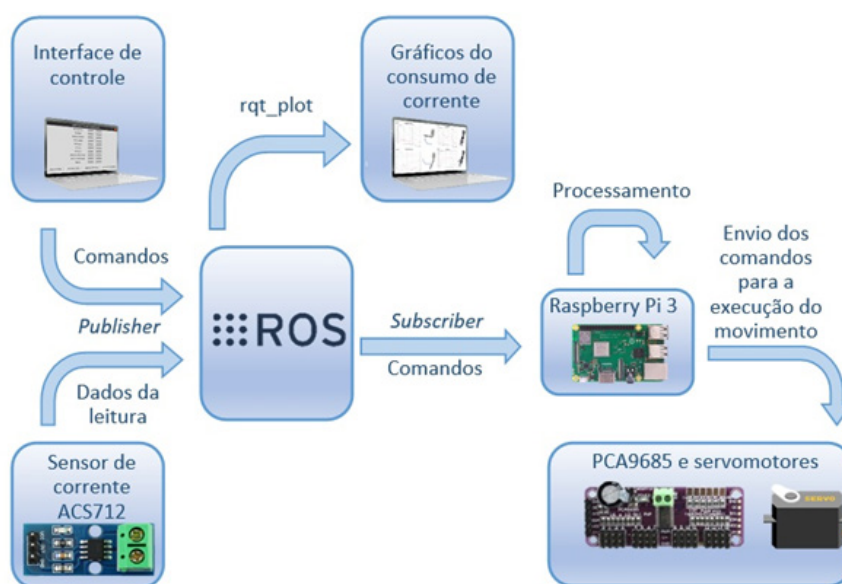
Com base nas informações apresentadas por Lentin (2015), o ROS Master desempenha o papel de registrar e localizar os nomes dos demais nós. Sem um ROS Master, os nós não conseguirão se descobrir, trocar mensagens ou acessar serviços. Em uma configuração distribuída, é recomendado executar o ROS Master em um computador central, permitindo que os outros nós remotos se encontrem ao se comunicar com esse mestre.

De acordo com Lentin (2015), os nós se comunicam entre si por meio de mensagens. As mensagens são simplesmente estruturas de dados contendo campos digitados, que podem armazenar um conjunto de informações e serem enviadas para outro nó. Existem tipos primitivos padrão (inteiro, ponto flutuante, booleano, etc.) que são suportados pelas mensagens do ROS. Além disso, podemos construir nossos próprios tipos de mensagens usando esses tipos padrão.

O ROS transporta mensagens através de canais nomeados chamados tópicos. Quando um nó envia uma mensagem por meio de um tópico, é considerado que ele está “publicando” (*Publishing*) o tópico, enquanto um nó recebe uma mensagem por meio de um tópico, é considerado que ele está “se inscrevendo” (*Subscribing*) no tópico (LENTIN, 2015).

A utilização do ROS é apresentada no diagrama da Figura 12, para a comunicação entre os componentes. Os servomotores são controlados por meio de um sinal de controle, o qual será enviado pelo dispositivo PCA9685. Esse dispositivo é controlado por uma Raspberry Pi 3, a qual se comunica com o *framework* ROS e pode receber mensagens dos outros dispositivos conectados.

Figura 12 – Fluxograma do algoritmo de controle e sensoriamento



Fonte: Dos autores.

Os movimentos do robô são comandados a partir de uma interface gráfica, que em um computador conectado ao ROS, o computador executa um *node* que publica os comandos da interface em um tópico, nesse tópico a Raspberry Pi consome os dados para realizar o controle dos servomotores. Também nesse computador, é executado o *rqt_plot*, uma ferramenta do ROS para monitorar tópicos e mostrar os dados em formato de gráficos.

A leitura do sensor de corrente é obtida a partir do Arduino, o qual cria um *node* no ROS e publica os valores do sensor de corrente ACS712, utilizando o *rqt_plot* os dados da leitura de corrente são expostos de maneira gráfica e permitem uma visualização clara dos dados e um monitoramento constante em tempo de operação do sistema.

A realização dos movimentos do robô atua com a utilização dos servomotores, os quais possuem um sensor interno, como um potenciômetro, que informa sua posição atual, e utilizando o método de controle PID para minimizar a diferença entre o ângulo desejado e o sinal de controle. Desse modo, o servomotor atinge a posição desejada com precisão e estabilidade. Utilizando esse método, o hardware interno dos servomotores tem conhecimento de que ângulo ele se encontra, sem a necessidade de se realizar ajustes por meio de sensores, pois esses já se encontram embarcados no servomotor. Esse controle é facilitado ainda mais quando se utiliza a biblioteca desenvolvida pelo próprio fabricante, utilizou-se a biblioteca *adafruit_servokit*, que é compatível com a PCA9685. A seguir, a Figura 13 ilustra um exemplo de utilização da biblioteca de controle dos servomotores *adafruit_servokit* em conjunto com a utilização da biblioteca *rospy* para a utilização dos recursos do ROS.

Figura 13 – Código fonte para a utilização da PCA9685 em conjunto com o ROS

```
1  #!/usr/bin/env python3
2  import rospy
3  from std_msgs.msg import Float64MultiArray
4  from adafruit_servokit import ServoKit
5
6  kit = ServoKit(channels=16)
7
8  # Função de callback para receber e separar os valores
9  def callback(data):
10
11     # Obtém o valor do ângulo (elemento 0 do array) e o índice (elemento 1 do array)
12     angulo = data.data[0]
13     indice = data.data[1]
14
15     # Envia o comando para a PCA9685 controlar o servomotor correspondente
16     kit.servo[int(indice)].angle = int(angulo)
17
18 # Inicialização do nó ROS
19 rospy.init_node('pca_control')
20
21 # Subscrição ao tópico onde será publicada a mensagem contendo o angulo e indice
22 rospy.Subscriber('/servo_control', Float64MultiArray, callback)
23
24 # Mantém o nó ROS ativo
25 rospy.spin()
```

Fonte: Dos autores.

Observa-se na Figura 13, linha 6, que para mover o servomotor é utilizada a classe *ServoKit*, que permitirá que os servomotores sejam controlados. Para realizar o controle, utiliza-se o ângulo objetivo do movimento e o índice correspondente do servomotor a ser controlado. O código inicia um *Ros Node* (Figura 13, linha 19) chamado 'pca_control', onde aguarda mensagens a serem publicadas no *Ros Topic* '/servo_control' (Figura 13, linha 22) criado. As mensagens publicadas devem conter o ângulo e o índice para realizar a atuação a partir da placa PCA9685. Com esse nó funcionando como *subscriber*, é possível controlar os servomotores utilizando outros nós com a função *publisher*.

Para o atual projeto escolheu-se desenvolver uma interface gráfica, que utiliza a função *publisher* para enviar as mensagens ao nó 'pca_control'. Essa interface é utilizada por um computador conectado ao mesmo Mestre ROS que a Raspberry Pi, permitindo que os nós se comuniquem entre os dispositivos. A criação da interface permite uma abordagem intuitiva do controle dos servomotores; optou-se por adicionar controles separados para cada servomotor e também botões para realizar ações pré-definidas.

Em continuidade, a Figura 14 apresenta uma imagem da interface desenvolvida, que mostra a movimentação dos controles deslizantes, que resultam na atuação do respectivo servomotor. Essa aplicação desenvolvida serve de ferramenta de testes, ajustes e configuração dos limites de operação dos atuadores do robô, auxiliando na posterior criação de gestos mais complexos. Além dos movimentos separados para cada servomotor, os movimentos pré-configurados são acionados nos botões: Abrir Mão, Fechar Mão, Baixar Bíceps e Levantar Bíceps.

Figura 14 - Interface desenvolvida para o controle dos servomotores



Fonte: Dos autores.

Os gestos criados nos botões da interface (Figura 14) serviram de parâmetros para os testes iniciais de consumo de corrente e da criação de gestos na plataforma. A utilização de gestos com servomotores se movendo simultaneamente retrata a realidade da utilização da plataforma, os movimentos de abrir e fechar a mão foram criados para cumprir essa finalidade, já o gesto de levantar e baixar o bíceps foi criado para testar o funcionamento de um servomotor em um movimento que necessita de maior torque.

Utilizando esta interface de testes, em conjunto com o sensor de corrente ACS712, é possível obter a corrente em amperes que os servomotores estão consumindo durante a execução dos movimentos. A obtenção do consumo dos servomotores é uma informação valiosa para o projeto, porque servirá de recurso para os estudos posteriores na plataforma. Para a coleta dos dados, utilizou-se o Arduino para obter as leituras do sensor, com a utilização do microcontrolador, e com a média de 400 amostras a cada 20 milissegundos, a qual é convertida em corrente. Isso implica uma frequência de amostragem interna do ADC do Arduino de aproximadamente 9,6 kHz (9600 amostras por segundo) para a aquisição dos dados brutos, com a disponibilização de uma leitura média atualizada a cada 50 Hz (1/20 ms). Embora essa taxa de amostragem interna do ADC seja o padrão para o Arduino UNO e permita detectar uma boa gama de variações, a etapa de média de 400 amostras tende a suavizar os picos de corrente de curtíssima duração. Essa abordagem foi considerada suficiente para a análise do consumo energético médio e dinâmico dos servomotores nos gestos programados. Contudo, para uma caracterização mais precisa e detalhada da dinâmica de transientes e picos abruptos, é recomendado futuramente registrar os dados brutos sem a média ou empregar técnicas de amostragem e processamento que preservem melhor esses eventos de alta frequência. O resultado do processamento dos dados foi publicado em um tópico no ROS, permitindo a visualização utilizando as ferramentas padronizadas do *framework*.

Aplicação de testes na plataforma e análise dos resultados

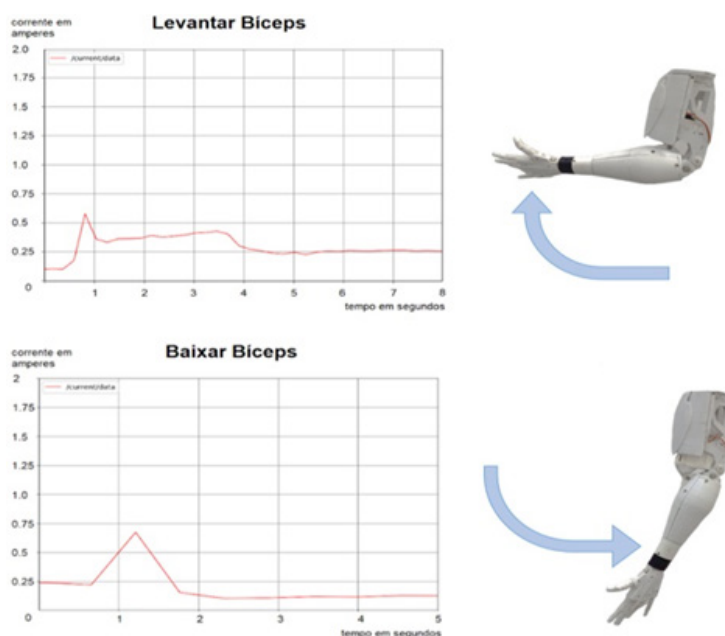
Para um sistema robótico que busca a sincronia de movimentos entre múltiplos atuadores, a latência inerente ao sistema de controle representa um fator crítico. Durante o desenvolvimento e os testes do InMoov-URI, a cadeia de comunicação foi constantemente considerada, abrangendo a Interface Gráfica, o ROS Master, o Raspberry Pi (atuando como *subscriber*), a comunicação via protocolo I2C com a placa PCA9685 e, finalmente, a resposta dos servomotores. Observou-se que potenciais atrasos em qualquer ponto dessa cadeia podem impactar a precisão e a estabilidade dos movimentos síncronos programados, resultando em pequenas defasagens entre os atuadores. Embora a medição quantitativa da latência em cada estágio não tenha sido o foco principal deste trabalho, a otimização do código Python e a configuração do ROS foram realizadas com vistas a minimizar tais defasagens, permitindo a execução satisfatória dos gestos.

Desta forma, seguindo os procedimentos metodológicos, foram definidos alguns cenários de testes para a arquitetura de controle especificada. A seguinte sequência de testes foi realizada:

- A movimentação de levantar e abaixar o bíceps que utiliza um servomotor TD-8130MG.
- Atuação simultânea de cinco servomotores do modelo MG996R nos gestos de abrir e fechar a mão.
- Verificação do consumo de corrente da plataforma durante a realização de cada teste.

Os resultados da leitura do sensor de corrente durante os testes com o bíceps e a mão do robô, em formato de gráfico, estão expostos na Figura 15 e na Figura 16, respectivamente. O gráfico foi criado utilizando uma ferramenta do ROS chamada “*rqt_plot*”. Essa ferramenta monitora o tópico onde os dados do sensor de corrente estão sendo publicados e cria os gráficos com esses dados em relação ao tempo.

Figura 15 - Testes de corrente com o bíceps.

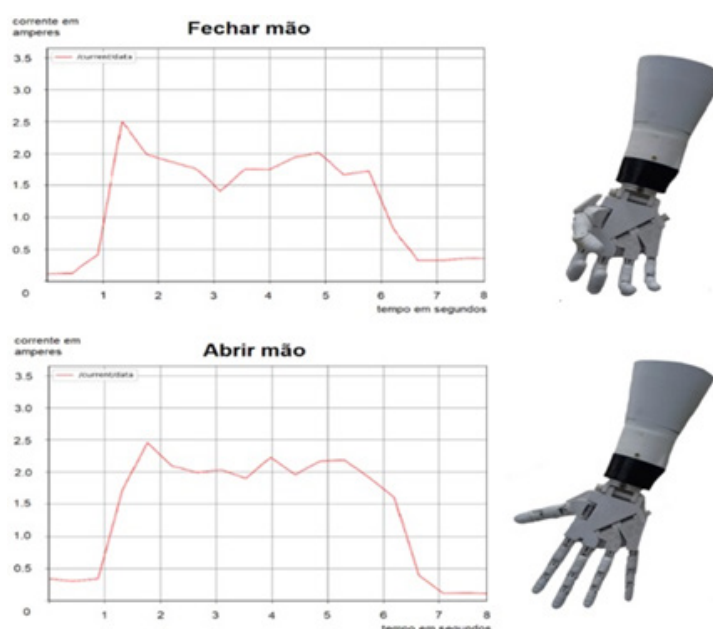


Fonte: Dos autores.

A análise dos gráficos relativos ao movimento de levantar o servomotor proporciona informações essenciais para a compreensão do comportamento dessa plataforma. Um pico inicial de corrente de 0,6 A é notado, indicando uma demanda significativa de energia na fase inicial do movimento. Essa observação se alinha com os princípios da física e da mecânica, uma vez que o início do movimento requer a superação da inércia, como discutido por Halliday e Resnick (2017).

A necessidade de superar essa inércia inicial resulta em um aumento no consumo de energia, conforme refletido pelo pico registrado na corrente. À medida que o servomotor continua a ascender, a corrente se estabiliza em torno de 0,3 A durante o restante do movimento. Isso sugere que, após vencer a resistência inicial, menos energia é necessária para manter o movimento constante. Após a conclusão do movimento, é observado que o braço permanece elevado, resultando em um consumo constante de 0,25 A.

Figura 16 - Testes de corrente com a mão.



Fonte: Dos autores.

A análise do movimento de abertura e fechamento da mão, que envolve cinco servomotores, também fornece informações essenciais. Durante o início do movimento, é observado um pico inicial de corrente atingindo 2,5 A, indicando uma demanda energética elevada. Esse consumo está relacionado ao movimento síncrono de múltiplos servomotores em conjunto com a superação de resistências iniciais. Ao longo do movimento, a corrente se estabiliza em um consumo médio de 2 A, demonstrando um padrão consistente de demanda energética. Essa observação ajuda a compreender as características de consumo dos servomotores envolvidos no movimento da mão, contribuindo para uma otimização mais eficiente na implementação de movimentos complexos.

Conclusões e perspectivas futuras

Ao longo do projeto foram exploradas a montagem e programação de robôs humanoides, tendo como base a plataforma InMoov. A impressão 3D se mostrou uma abordagem eficiente para criar peças mecânicas, permitindo aprimoramentos contínuos no robô InMoov-URI. A integração do framework ROS (*Robot Operating System*) possibilitou a integração de todos os componentes eletrônicos utilizados na plataforma, com controle e sensoramento do robô humanoide utilizando o método de comunicação *publish/subscribe*. O ROS se mostrou uma ferramenta robusta para implementar movimentos e gestos complexos no robô. Durante os testes práticos de controle, verificou-se que o objetivo central do trabalho de agregar e possibilitar movimentos síncronos dos atuadores foi atingido com sucesso. Assim, foram gerados e analisados gráficos de desempenho que permitiram acompanhar os movimentos e observar os limites físicos e eletrônicos da plataforma, considerando tanto o consumo de corrente como a variação angular dos movimentos realizados. No entanto, é fundamental reconhecer que o sistema atual apresenta limitações. Restrições de torque foram observadas em movimentos que exigiam maior carga, assim, a robustez mecânica das peças impressas em 3D requer contínuo aperfeiçoamento para suportar o uso prolongado, além da otimização de eventuais atrasos de comunicação, que é um desafio a ser abordado. Por conseguinte, este projeto nos proporcionou uma visão abrangente das possibilidades de aplicação da robótica com humanoides, incluindo atividades de ensino e pesquisa, sendo possível aplicar e estudar conceitos clássicos de engenharia e computação de forma prática.

Olhando para o futuro, vislumbramos uma série de perspectivas altamente promissoras para mitigar as limitações identificadas e expandir as capacidades do InMoov-URI. A transição para o framework ROS 2, com suas características avançadas e melhorias de desempenho, destaca-se como um caminho viável para otimizar a comunicação e o controle, impactando diretamente na precisão e responsividade dos movimentos. Devido à grande quantidade de atuadores e à necessidade de gerenciar eficientemente as demandas de torque, uma análise detalhada do consumo energético se apresenta como um passo crucial para otimizar a autonomia operacional do robô. Nesse cenário, a exploração de estratégias de conservação de energia, como a implementação de técnicas de hibernação, emerge como um campo de pesquisa altamente pertinente, contribuindo para a longevidade e confiabilidade do sistema.

Referências

ACUÑA, Fausto et al. Humanoid interpreter for teaching basic sign language. In: 2016 IEEE International Conference on Automatica (ICA-ACCA). IEEE, 2016. p. 1-7.

ALLEGRO, MicroSystems. Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor. 2022. Disponível em: <https://www.allegromicro.com/-/media/files/datasheets/acs712-datasheet.pdf>. Acesso em: jun. 2024.

ARAUJO, F. B. U. Sistemas de Controle. UFRN, Natal-RN, 2007.

ARAÚJO, Mateus Amarante et al. Aplicação do ROS no controle de movimento de robôs equipados com motores Dynamixel em Linux de Tempo Real. 2017.

AXELSSON, Minja et al. Designing an inmoov robot to teach assistive sign language to children with autism. 2018.

BAEK, Eu-Tteum; IM, Dae-Yeong. ROS-based unmanned mobile robot platform for agriculture. *Applied Sciences*, v. 12, n. 9, p. 4335, 2022.

BISCHOFF, Rainer; GRAEFE, Volker. Hermes—an intelligent humanoid robot designed and tested for dependability. In: *Experimental Robotics VIII*. Springer, Heidelberg, 2003. p. 64-74.

BOHNENBERGER, Rafael W.; ROQUE, Alexandre S. Estudo sobre dimensionamento do circuito de potência e controle dos servomotores do Robô InMoov-URI. Trabalho de Conclusão de Curso. Universidade Regional Integrada do Alto Uruguai e das Missões. 2020.

BOX, George EP; HUNTER, J. Stuart; HUNTER, William G. Statistics for experimenters. In: *Wiley Series in Probability and Statistics*. Wiley Hoboken, NJ, USA, 2005.

CONSTANTIN, George et al. Monitoring the Current Provided by a Hall Sensor Integrated in a Drive Wheel Module of a Mobile Robot. *Machines*, v. 11, n. 3, p. 385, 2023.

DO NASCIMENTO, Guilherme Araujo Machado; NETO, Manoel Morais Lemos; DA SILVA, Wyllian Bezerra. Uma Aplicação Didática do Protocolo I2C em Sistemas de Comunicação / A Didactic Application of the I2C Protocol in Communication Systems. *Brazilian Journal of Development*, v. 7, n. 10, p. 94837-94853, 2021.

DOS SANTOS, J. R. S., BALBINO, A. C., SENA, A., SANTOS, R., da CRUZ SOUZA, J. W., & PAVANI FILHO, A. Testes e Desenvolvimento Preliminar de Módulos e Bibliotecas para Comunicação em Português para o Robô Humanoide Arthur FrankSteam Utilizando PLN. In *Escola Regional de Computação Bahia, Alagoas e Sergipe (ERBASE)* (pp. 185-194). SBC, 2024.

FERREIRA, Eduardo. Desenvolvimento de um Framework para a realização de movimentos complexos no InMoov-URI. Trabalho de Conclusão de Curso. Universidade Regional Integrada do Alto Uruguai e das Missões, Santo Ângelo, 2019.

FRANKLIN, Gene F.; POWELL, J. David; EMAMI-NAEINI, Abbas. *Sistemas de controle para engenharia*. Bookman Editora, 2013.

HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. *Fundamentals of Physics, Volume 1*. John Wiley & Sons, 2017.

HERNÁNDEZ RAMÍREZ, Óscar. Diseño y programación de un robot humanoide de bajo coste, Universidad de Alicante. Dep. de Física, Ing. de Sistemas y Teoría de la Señal, Espanha, 2016.

HIRAI, Kazuo, et al. "The development of Honda humanoid robot."; *Proceedings*. 1998.

HOMBURG, Daniel et al. RoboTalk-Prototyping a Humanoid Robot as Speech-to-Sign Language Translator. In: Proceedings of the 52nd Hawaii International Conference on System Sciences. 2019.

LENTIN, Joseph; CACACE, J. Mastering ROS for robotics programming. Birmingham: Packt Publishing, v. 3, 2015.

KANEKO, Kenji et al. Humanoid robot HRP-3. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008. p. 2471-2478.

KOTONYA, Gerald; SOMMERVILLE, Ian. Requirements engineering: processes and techniques. Wiley Publishing, 1998.

LANGVIN, Gael. InMoov Open-source 3D Printed Life-size Robot. Disponível em: <http://inmoov.fr/>, 2019. Acesso em: 25 ago. 2024.

MCCOMB, Gordon. Robot Builder's Bonanza. 4. ed. New York: McGraw-Hill Education, 2011.

MIŠEIKIS, Justinas et al. Lio-a personal robot assistant for human-robot interaction and care applications. IEEE Robotics and Automation Letters, v. 5, n. 4, p. 5339-5346, 2020.

MORI, Masahiro; MACDORMAN, Karl F.; KAGEKI, Norri. The uncanny valley [from the field]. IEEE Robotics & Automation Magazine, v. 19, n. 2, p. 98-100, 2012.

MULYANTO, Agus et al. Implementation 2D Lidar and Camera for detection object and distance based on RoS. JOIV: Int Journal on Informatics Visualization, v. 4, n. 4, p. 231-236, 2020.

NEWSROOM, Toyota Global. Toyota Unveils Third Generation Humanoid Robot T-HR3. (21 November 2017). Retrieved March, v. 21, p. 2018, 2017.

OKAGBUE, E. F., MUHIDEEN, S., ANULIKA, A. G., Nchekwubemchukwu, I. S., Chinemerem, O. G., Tsakuwa, M. B., ... & MWASE, C. An in-depth analysis of humanoid robotics in higher education system. *Education and Information Technologies*, 29(1), 185-217, 2024.

PARALI, Levent; ALI, S. A. R. I.; MEHMET, E. S. E. N. Design of a 3D Printed Open Source Humanoid Robot. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, v. 11, n. 2, p. 411-420, 2022.

RADFORD, Nicolaus A. et al. Valkyrie: Nasa's first bipedal humanoid robot. Journal of Field Robotics, v. 32, n. 3, p. 397-419, 2015.

ROMERO, R. A. F. et al. Robótica Móvel. 1. Ed. – Rio de Janeiro: LTC, 2014.

SCHEFFER, Eduardo., NERY, Eduardo G., SEIXAS, Jordana L. Teoria de controle e servomecanismo. 1 ed. Porto Alegre: Editora Sagah. 2018.

SCHERZ, Paul; MONK, Simon. Practical electronics for inventors. McGraw-Hill, 2013.

SILVA, B. R., ALVARADO, E. F., da SILVA, N. R., de OLIVEIRA, R. R., & GONÇALVES, W. Prototipagem como Metodologia Ativa de Ensino: a experiência acadêmica e o processo de aprendizagem. Em Congresso de Inovação, Ciência e Tecnologia do IFSP (Vol. 15, No. 1), 2024.

SOUZA, G. G., de ANDRADE, V. B., XEXÉO, G., Silva, F., & de OLIVEIRA, S. Superbactérias: um jogo educativo sobre o risco do uso inadequado de antibióticos. Em Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames) (pp. 1317-1322). SBC, 2023.

TEIXEIRA, Gustavo. Desenvolvimento do Framework para a Sincronização dos Movimentos da Cabeça do InMoov-URI. Trabalho de Conclusão de Curso. Universidade Regional Integrada do Alto Uruguai e das Missões, Santo Ângelo, 2018.

VALENCIA, Nicolás Ortiz et al. Movement detection for object tracking applied to the InMoov robot head. In: 2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA). IEEE, 2016. p. 1-6.

WAZLAWICK, Raul. Metodologia de Pesquisa para Ciência da Computação, 2ª Edição. Elsevier Brasil, 2014.