

Introdução

O EduBot é um robô de movimento diferencial autônomo que vem sendo desenvolvido no laboratório LASCAR com propósito educacional e de uso pela própria universidade. O robô móvel possui uma estrutura formada por duas rodas fixas convencionais, controladas de maneira independente, e por duas rodas de apoio que possuem liberdade de movimento total no plano de deslocamento do EduBot. O primeiro passo para programar o robô em mais alto nível de programação é embutir o controle de trajetória em funções que combinam o acionamento dos motores para executar movimentos intuitivos.

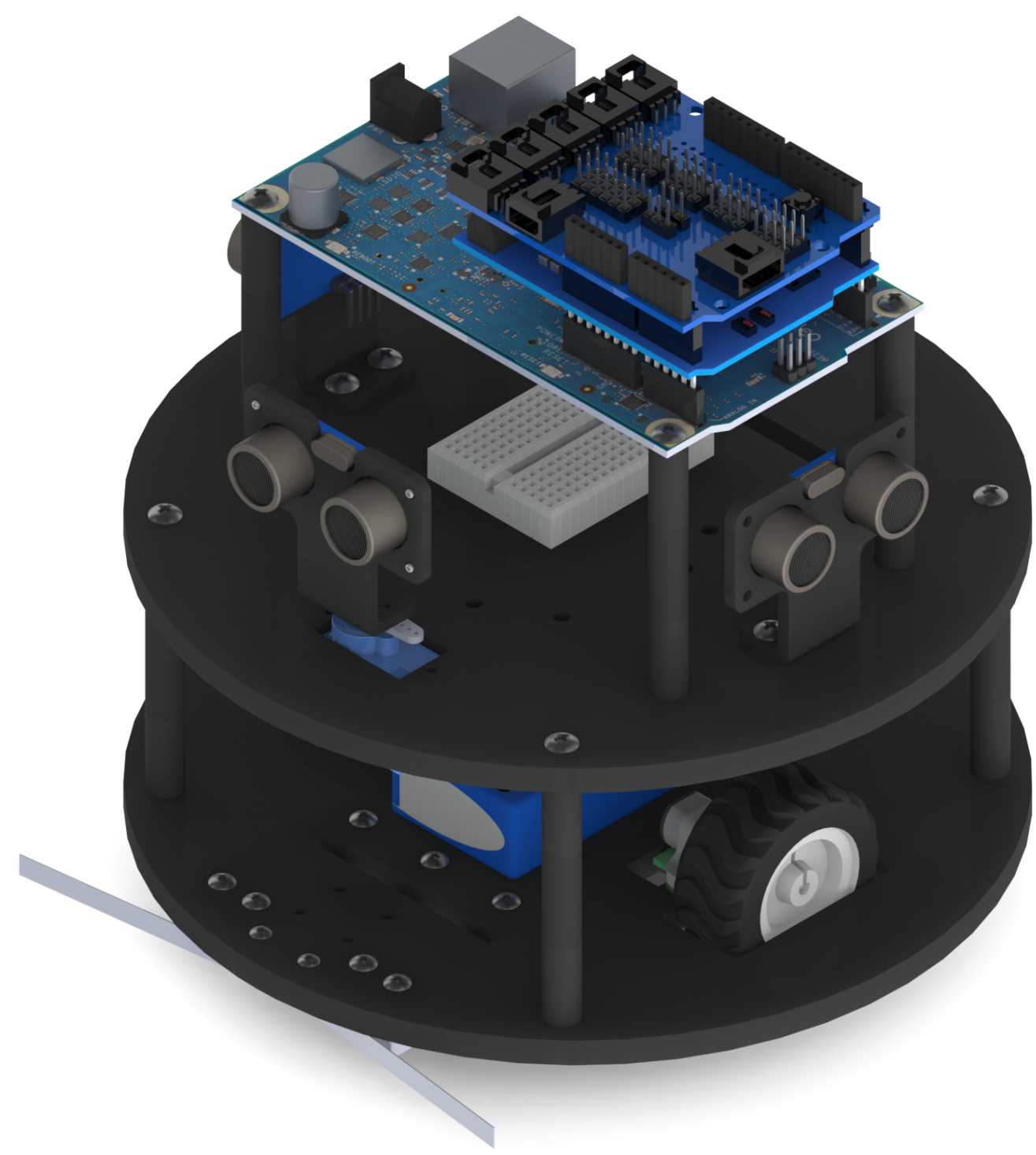


Figura 1: Projeto tridimensional para o EduRobot Intel Galileo Gen2. Fonte: Autores.

Objetivo

O deslocamento do robô para frente e para trás é feito ao se acionar os corretos pinos da ponte H disponível no EduBot em conjunto com uma entrada de PWM (Modulação de Largura de Pulso) definida individualmente para cada motor. Entretanto, mesmo que definidos ambos PWMs com mesmo valor, interferências elétricas e, principalmente, mecânicas se somam para criar uma derivação na trajetória retilínea idealizada do robô. O objetivo deste trabalho é prevenir um erro muito acentuado na trajetória do robô, regulando a velocidade das rodas individualmente. O bloco PID será alimentado com a informação contida nos contadores dos pulsos de encoder (um para cada roda).

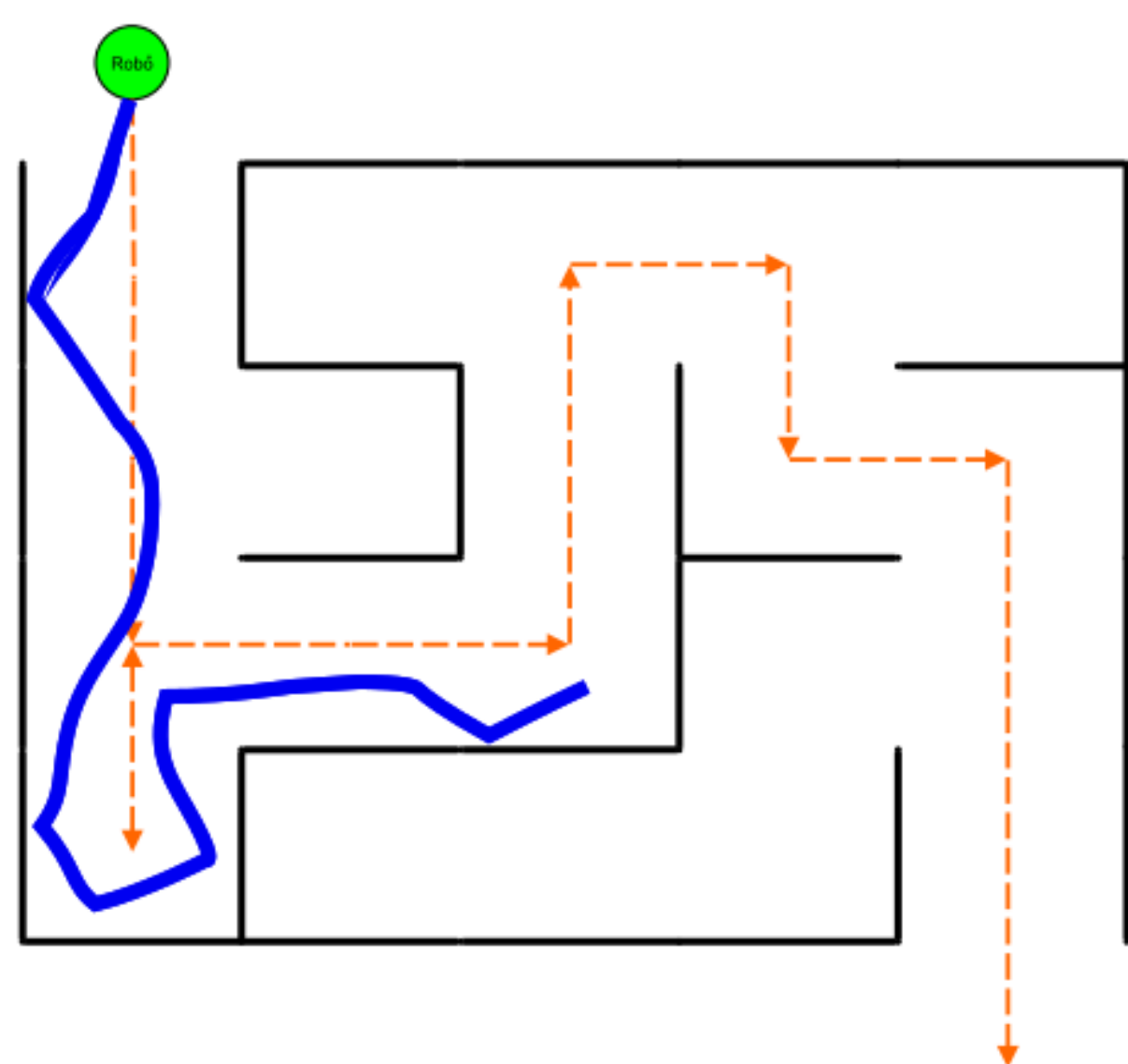


Figura 2: Robô diferencial usando PID (amarelo) e sem PID (azul)

Solução

O desenvolvimento da solução utiliza um PID (controlador proporcional integral derivativo) que recebe de entrada a diferença entre o contador de pulsos do encoder direito com o esquerdo. O proporcional atualiza a saída do PID de acordo com a atual entrada, contudo a verdadeira correção do erro na trajetória está na contribuição da integral, que leva em conta os erros acumulados anteriormente. A saída é posta em escala e utilizada para incrementar ou decrementar o PWM inserido em cada roda, controlando, assim a velocidade de rotação que cada uma executa. Um segundo PID é usado em paralelo com o controle de alinhamento do robô diferencial, sendo responsável pelo controle da distância percorrida. Utiliza-se a média entre os contadores dos encoders como entradas, que fisicamente é o deslocamento do ponto central do EduBot, visto que ambas as rodas estão a mesma distância do centro. Este segundo PID previne acelerações bruscas que seriam resultantes de atribuições manuais aos valores de PWM dos motores, além de estabilizar o robô na posição determinada, mesmo que este exceda a distância a ser percorrer. Através deste podemos usar comandos de movimentação que utilizam apenas a distância a ser percorrida como parâmetro, expressa em pulsos de encoder.

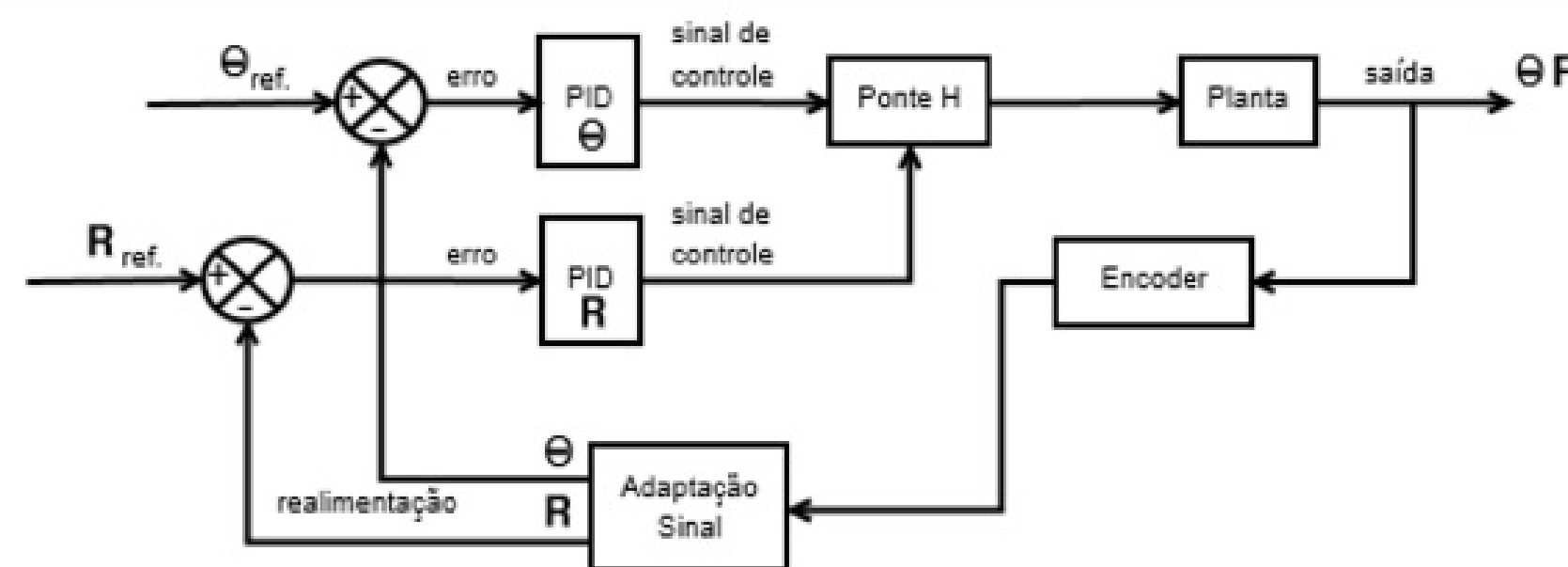


Figura 3: Diagrama do PID[2]

As saídas dos PIDs são utilizadas em conjunto para determinar não só a velocidade, mas o sentido dos motores. Existe uma função que recebe o incremento a ser efetuado nos setpoints de cada respectivo PID e movimenta o robô automaticamente, sem necessidade de atualizar o sentido ou velocidade de cada motor manualmente. Isso só é possível porque, considerando um modelo teórico, ao rotacionar-se o robô, a média dos contadores dos encoders continua a mesma, portanto, podemos usar o setpoint do primeiro PID para definir diretamente o ângulo de rotação do robô em relação a referência inicial. Analogamente, quando se movimenta o robô para frente, a diferença entre os contadores é a mesma.

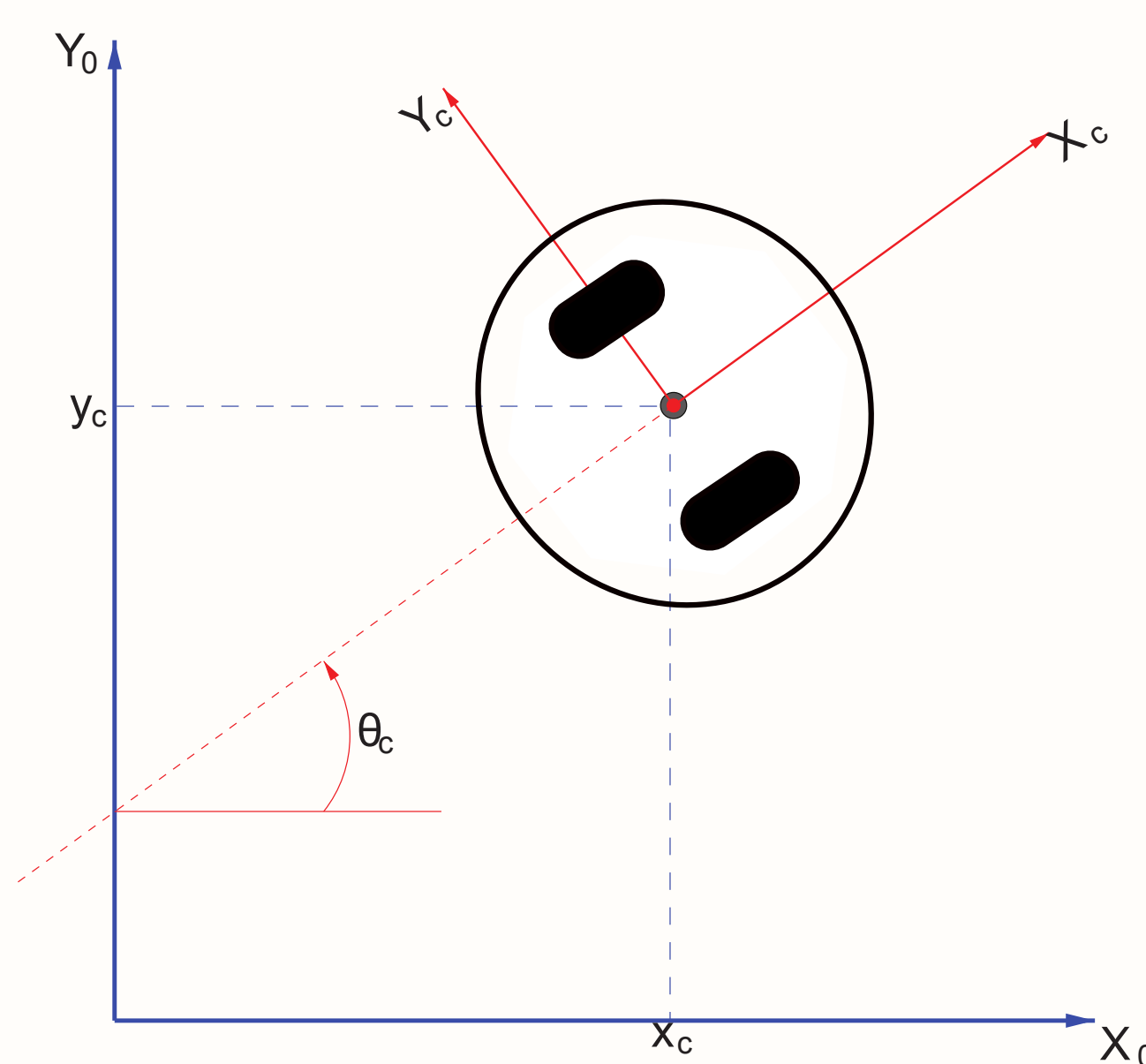


Figura 4: Orientação do robô em relação ao sistema inercial.[1]

O modelo cinemático do robô móvel do tipo unicycle está baseado nas seguintes considerações: o robô é constituído por um chassi rígido e rodas não-deformáveis, que se movem em um plano horizontal. Assume-se que o plano das rodas permanece na vertical durante a movimentação, com a rotação se dando em torno de um eixo horizontal cuja orientação é fixa ou variável com relação a um sistema de coordenadas associado ao corpo do robô, $\{X_c, Y_c, \theta_c\}$, descrito em relação ao sistema inercial $\{X_0, Y_0, \theta_0\}$, com θ sendo o ângulo que define a orientação do robô. Tal esquematização teórica permite uma abordagem utilizando coordenadas polares funcionando da seguinte maneira: Cada movimento é definido através da passagem de dois parâmetros, um de ângulo e outro de distância. Os PIDs têm seus setpoints alterados de acordo com os parâmetros passados, então, automaticamente corrigem a posição do robô para se enquadrar no setpoint.

Problemas com a abordagem

Mesmo utilizando o controle apresentado, o robô eventualmente irá acumular algum erro devido a eventos não considerados, exemplificados com o escorregamento de uma das rodas sobre a superfície. No caso, deve-se estabelecer uma nova referência a partir de elementos físicos detectados através de outros sensores. A aquisição de uma nova posição de referência é equivalente a reiniciar todo o processo de deslocamento.

Conclusões

O uso de um controlador Proporcional Integral Derivativo se provou prático e flexível, possibilitando estudos dos mais simples aos complexos, sendo ele a escolha certa na correção da trajetória de um robô diferencial. Embora os ganhos do controlador que são utilizados (ganho proporcional, integral e derivativo) tenham sido obtidos empiricamente, através de uma pesquisa na literatura para a primeira tentativa e subsequentes testes com pequenas alterações nas constantes, o erro na trajetória do EduBot[3] foi diminuído com a implementação do trabalho. Pelo ponto de vista educacional, as funções desenvolvidas pelo trabalho podem ser usadas para aperfeiçoar os algoritmos desenvolvidos por discentes para o EduBot.

Referências

- [1] T. T. T. Barros. Modelagem e implementação no ros de um controlador para manipuladores móveis. 2014.
- [2] A. N. Bonella. Proposta de um serviço para um robô móvel não holonômico com acionamento diferencial. 2015. Trabalho de final de curso - Engenharia Elétrica - UFRGS.
- [3] C. S. S. G. Junior, E. H. Maciel, R. Ventura, B. Henriques, and C. E. Pereira. Edubot: Proposta de um robô móvel não-holonômico com acionamento diferencial integrado com framework soa. *Simpósio Brasileiro de Automação Inteligente*, 2015.