

## ROBÔ SEGUIDOR DE LINHA COM CONTROLADOR PID

TIAGO GIACOMELLI ALVES<sup>1</sup>; GUIHERME PEREIRA PAIM<sup>2</sup>; LUCIANO ANACKER  
 LESTON<sup>3</sup>; MARCELO ESPOSITO<sup>4</sup>,

<sup>1</sup>Universidade Federal de Pelotas – [tgiacomelli.cee@ufpel.edu.br](mailto:tgiacomelli.cee@ufpel.edu.br)

<sup>2</sup>Universidade Federal de Pelotas – [guilhermepaim.liberato@gmail.com](mailto:guilhermepaim.liberato@gmail.com)

<sup>3</sup>Universidade Federal de Pelotas – [luleston@gmail.com](mailto:luleston@gmail.com)

<sup>4</sup>Universidade Federal de Pelotas – [marcelo.esposito@ufpel.edu.br](mailto:marcelo.esposito@ufpel.edu.br)

### 1. INTRODUÇÃO

Um sistema de controle em malha fechada é aquele no qual existe uma realimentação da saída proveniente de um sensor que mede a variável de controle, a qual será comparada a uma entrada de referência do sistema (ROSÁRIO, 2005). O emprego deste tipo de sistema tem como objetivo reduzir a sensibilidade às perturbações e mudanças nos parâmetros, melhorar o tempo de resposta em regime transitório, amenizar sobressinais e anular o erro em regime permanente. Pode, no entanto, acarretar em perda da estabilidade caso a técnica de controle utilizada não seja adequada para manter o sistema estável.

Geralmente são utilizados controladores ou compensadores em sistemas com realimentação. Segundo CRUZ (2004), o controlador é um dispositivo cuja finalidade é utilizar o erro resultante de uma comparação entre o valor desejado de uma variável e o seu valor real para calcular o valor da variável de controle. Conforme o autor, os três modos básicos de controle em malha fechada são: proporcional (P), integral (I) e derivativo (D). Quando inseridos em sistemas microprocessados, estes modos de controle podem ser vistos como algoritmos tendo em vista que devem ser parte do código do programa base.

O modo proporcional (P) tem como característica principal aumentar a velocidade da resposta do sistema. Ele é um dos tipos mais elementares e mais utilizados de controlador. Deve-se salientar que ao elevar seu ganho, oscilações indesejáveis poderão surgir, vindo a instabilizar a planta.

O modo integral (I) caracteriza-se por minimizar o erro em regime permanente. Seu funcionamento é baseado na integração da diferença entre a referência e a realimentação da saída (esta diferença é chamada de erro). Este modo pode causar o fenômeno chamado *windup* quando ocorre saturação do controlador, no qual o sinal erro não é anulado. Este tipo de problema geralmente é verificado em uma mudança grande de *setpoint*, na presença de um erro persistente ou em função de alguma perturbação na saída que esteja além da capacidade de controle do sistema. Alguns controladores dispõem de uma função capaz de eliminar o efeito *windup*. Esta função atua interrompendo a integração do erro ao ocorrer à saturação da saída do controlador e reiniciando-a quando a mesma já não estiver mais saturada.

O modo derivativo (D) tem um caráter antecipatório, sendo sua função reagir antecipadamente ao comportamento futuro do sinal erro tendo como base a sua taxa de variação. Ele é empregado no ajuste da resposta transitória do sistema.

Os modos integral (I) e derivativo (D) geralmente são utilizados juntamente com o modo proporcional formando uma estrutura PID ou separadamente, juntando-se somente com o modo proporcional, formando as estruturas PI ou PD. O projeto destes tipos de controladores se baseia na sintonia de seus parâmetros a fim de se obter a resposta desejada na saída do sistema. Na falta de um modelo adequado do processo, realiza-se esta sintonia por tentativa e erro, tornando essa tarefa

enfadonha e demorada. Para os casos em que se dispõe de um modelo do processo, ou ainda, apenas a sua resposta em frequência, métodos de projeto baseados na teoria de controle de processos podem ser utilizados (CRUZ, 2004). Cabe salientar que, mesmo quando sintonizado através de um método matemático, um ajuste fino via tentativa e erro pode ser necessário.

Com o aumento do ganho proporcional, eleva-se a velocidade de resposta do sistema. Com o aumento do parâmetro do integrador as oscilações e o erro em regime permanente são reduzidos, porém aumenta-se o sobressinal percentual. O aumento do parâmetro derivativo torna o sistema mais sensível a ruídos, mas reduz o sobressinal e aumenta a velocidade da resposta.

Comumente em cursos técnicos ou de engenharia, robôs móveis seguidores de linha são utilizados como ferramenta de ensino. O princípio de funcionamento não é demasiadamente complexo e permite trabalhar conceitos de mecânica, eletrônica, programação e sistemas de localização. O conceito se baseia em uma pista que possui uma linha de cor contrastante a qual o robô deverá manter-se próximo. Este tipo de robô geralmente é dotado de sensores ópticos, motores elétricos e microcontroladores. Quando o número de sensores ópticos é reduzido a apenas um, aumenta-se razoavelmente o grau de dificuldade de desenvolvimento do algoritmo computacional.

O presente trabalho apresenta a implementação de um robô móvel seguidor de linha utilizando um algoritmo de controle Proporcional-Integral-Derivativo (PID) juntamente com a estratégia de controle tipo *split-range*. Mesmo não sendo fornecidas informações prévias sobre a pista ao robô a trajetória deve ser seguida sem a perda de estabilidade, ou seja, o robô deve realizar todo o percurso mantendo-se sobre a linha.

## 2. METODOLOGIA

Neste trabalho, optou-se por uma abordagem empírica em todas as etapas. Para a confecção do robô fez-se uso do kit educacional *LEGO MINDSTORMS EDUCATION 9797*. Desta plataforma foi utilizado um sensor óptico, dois servomotores, um bloco lógico programável NXT, bem como partes fixas e peças para a montagem estrutural do robô.

O sensor óptico utilizado serve tanto para identificar a intensidade de luz em um ambiente, como para medir a intensidade de luz em superfícies coloridas, permitindo assim a distinção entre claro e escuro. Quando posicionado próximo a superfícies claras indica valores altos de leitura tendo em vista que elas refletem uma maior quantidade de luz. Da mesma forma, ao ser posicionado próximo a superfícies escuras o mesmo indica valores baixos de leitura. Desta forma, estes valores (altos e baixos) de leitura podem ser manipulados para detectar a localização do robô. A leitura do sensor é realizada em porcentagem (0-100%). Os valores gerados por este sensor são apenas relativos, não correspondem a qualquer unidade de medida.

Os servomotores proveem a capacidade de movimento ao robô. Eles são compostos de um motor e um sensor de rotação (*encoder*). O *encoder* é ligado juntamente com o motor ao bloco programável NXT. Este sensor possibilita a monitoração precisa da execução de movimentos pelo robô. É possível medir a posição angular do eixo do motor em graus, assim como o número de rotações completas.

O bloco programável NXT possui como funcionalidades: comunicação USB e *Bluetooth*, três portas de saída para a ligação dos servomotores, quatro entradas para a aquisição de dados dos sensores, *display* LCD e botões externos para a configuração e interface com o usuário. A avaliação das características eletrônicas internas do NXT não faz parte deste trabalho, sendo então estas informações negligenciadas.

A pista utilizada nos testes foi projetada pelo grupo do Laboratório de Sistemas de Controle da UFPel (LSC). A pista em estilo *banner* com um metro e meio de largura e dois metros de comprimento teve seu projeto baseado na Pista de Kart Arena Sapiens de Florianópolis/SC. Esta pista possui o fundo totalmente branco e a trajetória na cor preta. Para a programação do robô e implementação do sistema de controle foram utilizados o *software* Simulink do ambiente computacional Matlab e uma *toolbox* disponibilizada para a programação do bloco NXT (MATHWORKS, 2012).

A estratégia de localização baseou-se nas respostas do sensor óptico. Mantendo-se o robô com o sensor posicionado parte sobre a pista e parte sobre a linha, ou seja, parte sobre a área clara (área branca da pista) e parte sobre a linha preta (trajetória a ser seguida) o valor obtido foi de 41%. De modo similar foram obtidos também os valores percentuais nas posições totalmente expostas a área clara, a qual resultou em 69% e totalmente sobre a linha, a qual resultou em 37%. Através destas constatações e, tendo como sinal de referência um valor percentual representativo do contraste entre a pista e a linha, pôde-se realizar a identificação da posição do robô com o uso de apenas um sensor óptico, trazendo como única desvantagem o posicionamento fixo do mesmo em relação a um lado da linha.

O modelo de robô construído conta com apenas três rodas, sendo duas delas ligadas a servomotores individuais. A rota que o robô segue é controlada através da variação das posições angulares no tempo de um ou de outro servomotor, permitindo a variação da quantidade de movimento linear de cada roda, assim como a mudança de direção para esquerda ou para a direita. A terceira roda serve para dar sustentação e possui giro livre em 360 graus ao redor dela mesma, possibilitando desta forma manobras rápidas para ambos os lados.

Tendo em vista o uso de dois atuadores e com o objetivo de simplificar o algoritmo de controle optou-se pelo emprego da estratégia de controle *split-range* ao invés da atuação de controle independente em cada servomotor. Foram escolhidas como variáveis de controle as potências dos servomotores, sendo estas definidas em porcentagens. O primeiro servomotor recebe o valor da potência de trabalho diretamente da saída do controlador PID. O segundo servomotor recebe o complemento desta, não excedendo o total de 100% de potência se somadas.

### 3. RESULTADOS E DISCUSSÃO

A Figura 1a evidencia o comportamento do robô sobre a pista (Figura 1b). Considerando-se as características do robô e do modo de controle utilizado, constatou-se que o *setpoint* (valor desejado) foi atingido sem oscilações abruptas (baixo tempo de acomodação) e com um razoável erro de regime permanente. O ajuste do controlador permitiu percorrer a linha de cor preta sem mudanças indesejadas ao longo do percurso, mesmo no caso de curvas fechadas com ângulos inferiores a noventa graus.

Após analisar a movimentação do robô em relação a curvas de diferentes ângulos, notou-se que o mesmo poderia se mover demasiadamente sobre a área

escura e ir de encontro com a área clara no lado oposto da linha, perdendo assim a sua localização. Corrigiu-se este problema ao fazer o robô permanecer mais sobre a área branca, neste caso foi definido como *setpoint* 50% (Figura 1c) ao invés do valor de 41% anterior. Para evitar derrapagens ou saídas de rota repentinas a potência dos motores foi limitada em 35%. Este valor apresentou os melhores resultados experimentais, considerando-se a rapidez do robô a cada volta completada na pista sem a perda de estabilidade. Obteve-se o melhor resultado com o valor de 80 para a parte proporcional (P), 12 para o integrador (I) e 5 para o derivativo (D). Identificou-se também que qualquer fonte de luz (lâmpadas, luz solar, sombras, etc.) e até mesmo a altura do sensor pode perturbar a quantidade de luz sensível para a mesma superfície, aumentando desta forma os esforços de controle ou gerando instabilidade.

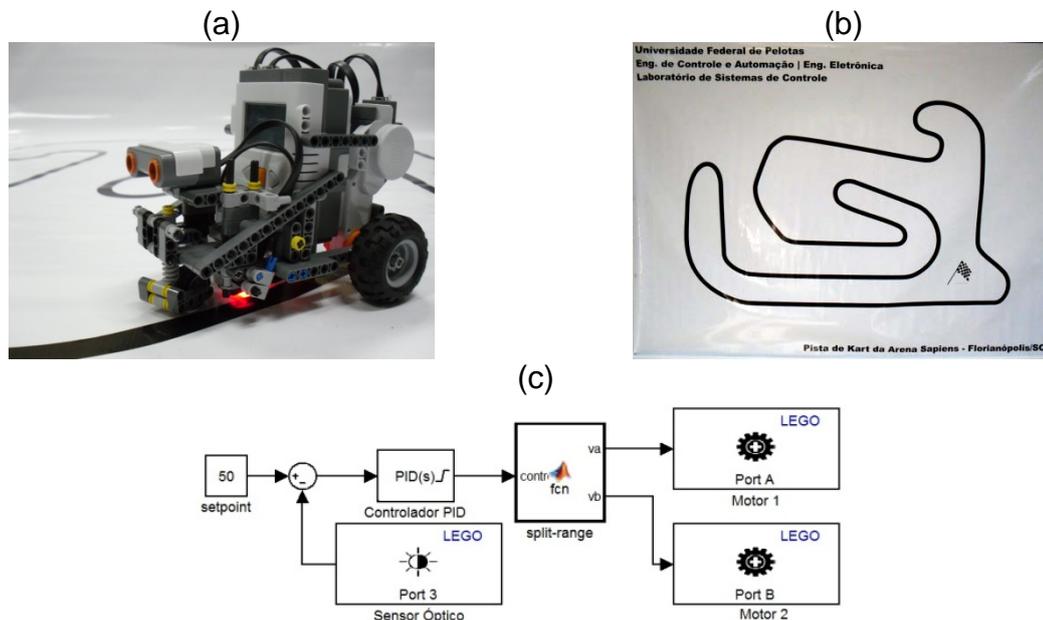


Figura 1: a) comportamento do robô sobre a pista; b) pista de testes utilizada no trabalho; d) Malha de controle do robô seguidor de linha.

#### 4. CONCLUSÕES

Utilizou-se com sucesso conhecimentos de “controle clássico” e de estratégias de controle, desenvolvidas em disciplinas específicas dos cursos de Engenharia Eletrônica e de Engenharia de Controle e Automação. O projeto de um robô com comportamento autônomo em relação à trajetória a ser seguida foi atendido mesmo com o uso de um método de tentativa e erro para a sintonia do controlador. Como trabalho futuro justificar-se-ia a utilização de técnicas de sintonia baseadas na teoria de sistemas de controle (projeto de compensadores).

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

- ROSÁRIO, J.M. **Princípios de Mecatrônica**. São Paulo: Prentice Hall, 2005.  
 CRUZ, J. J. **Entendendo e Ajustando Malhas de Controle**. São Paulo, 2004.  
 MATHWORKS. Simulink Support Package for LEGO MINDSTORMS NXT hardware. Acessado em 27 de setembro de 2013. Online. Disponível em: <http://www.mathworks.com/matlabcentral/fileexchange/37896-simulink-support-package-for-lego-mindstorms-nxt-hardware-r2012b>