

Always standing robot

O projeto consiste num *robot* autónomo capaz de se equilibrar de forma automática – com a implementação do controlo Proporcional Integral e Derivativo (PID) em apenas duas rodas e com o auxílio de diversos sensores. Estes sensores asseguram a locomoção numa trajetória, desviando-se de obstáculos, e garantem o equilíbrio e o controlo PID da velocidade imposta.

1. CONTROLADOR PID

O PID é uma metodologia de controlo linear, cuja lei de controlo é baseada no erro da variável a controlar. Foi pela primeira vez apresentado por N. Minorsky em 1922. No entanto, atualmente, continua a ser o método de controlo mais usado em todo o mundo, tanto por parte da indústria em sistemas de controlo industrial, como na maioria dos controladores comercialmente disponíveis e em robótica móvel. Neste ponto é descrito o controlador PID, as suas principais aplicações e características.

A popularidade dos controladores PID pode ser atribuída com base em vários fatores: são matematicamente simples, de fácil compreensão, fiáveis, e requerem uma baixa capacidade e um custo computacional. Têm tanto destaque, que são muitas as técnicas não lineares utilizadas para o ajuste dos seus parâmetros. Existem mesmo autores que referem que, em processos com perturbações imprevistas e frequentes, o método PID bem ajustado é o que apresenta um melhor desempenho e robustez, exceto nos sistemas com atraso [1]. Por um lado, é verdade que o controlo PID está massivamente estudado na literatura [2][3] e é uma das técnicas de controlo mais populares na indústria porque, além das vantagens acima referidas, responde com desempenhos suficientes na maioria dos processos com requisitos pouco exigentes. Por outro lado, também é verdade que o avanço no desenvolvimento das técnicas de controlo não linear e da análise da complexidade deste tipo de sistemas, tem potenciado o controlo de sistemas até requisitos de desempenho totalmente fora do alcance das técnicas PID lineares, independentemente do seu método de ajuste e da configuração com que se apresente. Portanto, não se realça tanto o possível abandono da utilização desta técnica devido ao número de processos que pode efetivamente controlar, mas antes porque os processos tendem a ser cada vez mais complexos, e pela facilidade de implementação de algumas técnicas não lineares, que são flexíveis ao ponto de regular o desempenho desejado até às solicitações mais exigentes.

Nos métodos práticos de sintonia, o primeiro passo para a utilização de um controlador PID é a escolha do tipo de controlador a usar. Este pode conter apenas a componente de ação proporcional (P), a ação proporcional-integral (PI), a ação proporcional-derivativa (PD) ou então a ação proporcional-integral-derivativa (PID). Posteriormente é necessário fazer o ajuste dos vários parâmetros do controlador. Este ajuste consiste na dedução, tendo em conta a resposta do sistema, quando este

é sujeito a entradas específicas com valores que vão permitir o cálculo dos referidos parâmetros. Este procedimento possui a vantagem de não exigir a necessidade de conhecer o modelo do sistema, sendo este, muitas vezes, difícil de determinar.

Já no método analítico procede-se à sintonia dos modos PID para uma aplicação específica, de modo a que determinados critérios de desempenho sejam verificados. Normalmente, este método é usado sempre que se conhece a função de transferência do sistema.

Estrutura do PID

Na Figura 1 é possível observar a configuração de um controlador PID em malha fechada.

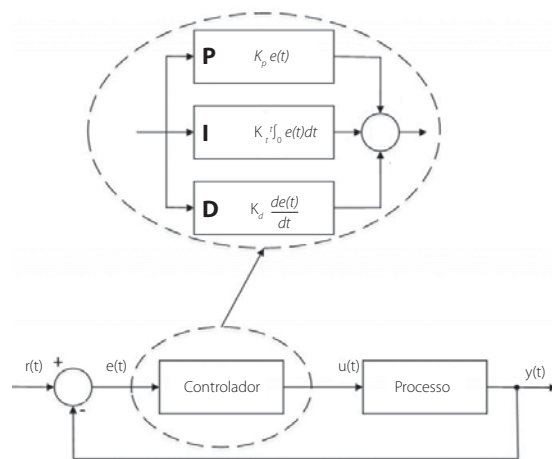


Figura 1. Diagrama de blocos do controlador PID.

O objetivo deste tipo de controlador passa por manter a saída do processo $y(t)$, no valor desejado ou então no valor de referência dado por $u(t)$, eliminando continuamente o erro $e(t)$.

Para isso, o controlador aplica ininterruptamente a ação de controlo à entrada do processo. Esta ação é composta pela soma dos termos que o constituem. Ou seja, pela soma do termo proporcional, integral e derivativo do erro, sendo que o erro é a diferença entre o sinal de entrada e o sinal de saída, dada pela expressão (1).

$$e(t) = u(t) - y(t) \quad (1)$$

O processo de sintonia do controlo começa pelos requisitos de desempenho do sistema. O controlo de desempenho do sistema é geralmente medido pela inserção de uma função em degrau, definida como setpoint, sendo posteriormente medida a resposta da variável do processo. Geralmente, a resposta é quantificada pelas características da onda de resposta. O tempo de subida ou *Rise Time* (T_r) é o tempo que o sistema leva para ir de 10% a 90% do estado estacionário, ou valor final. A sobree-longação máxima ou o *Percent Overshoot* (M_p) é o valor em que a variável do processo ultrapassa o valor final, expresso como

uma percentagem do valor final. O tempo de estabelecimento ou o *Settling Time* (T_s) é o tempo necessário para a variável do processo atingir uma determinada percentagem (normalmente 2%) do valor final.

Por fim, o erro em regime permanente ou o *Steady-State Error* (e_{ss}) é a diferença final entre as variáveis do processo e o *setpoint* [3]. De modo a se ajustarem estes parâmetros para que o desempenho do sistema seja o ideal, é necessário alterar os parâmetros do controlador.

Ou seja, é preciso alterar o ganho proporcional (K_p), o ganho integrativo (K_i) e o ganho derivativo (K_d) [3]. Assim sendo, tem-se:

$$u(t) = \text{Ação de Controlo} = K_p \cdot e(t) + K_i \cdot \int_0^t e(t)dt + K_d \cdot \frac{de(t)}{dt} \quad (2)$$

Ação Proporcional

Este tipo de controlo é caracterizado pela seguinte expressão:

$$u(t) = K_p \cdot e(t) \xrightarrow{L} U(s) = K_p \cdot E(s) \quad (3)$$

A ação proporcional, como o próprio nome indica, age proporcionalmente ao erro $e(t)$ entre a entrada e a saída do sistema. Sintonizando este parâmetro, quanto maior for o seu valor, menor será o erro em regime permanente.

Ou seja, a precisão do sistema em malha fechada é otimizada. O erro $e(t)$ será diminuído com o aumento de K_p , no entanto nunca poderá ser anulado. Em contrapartida, quanto maior for o ganho de K_p , mais instável o sistema pode ficar.

Ação Proporcional-Integral

Este tipo de controlo é caracterizado pela seguinte expressão:

$$u(t) = K_p \cdot [e(t) + \frac{1}{T_i} \int_0^t e(t)dt] \xrightarrow{L} U(s) = K_p \cdot (1 + \frac{1}{T_i \cdot s}) E(s) \quad (4)$$

Onde T_i é a designação de tempo integral e representa o tempo necessário para que haja uma igualdade entre a ação integral e a ação proporcional.

A ação integral age proporcionalmente à integral do erro do sistema. Esta é responsável por garantir um erro igual a zero em regime permanente para as entradas em degrau, quando o sistema em malha fechada for internamente estável, e rejeitar perturbações aplicadas na entrada do processo [3].

Na prática, após aplicar este tipo de controlador, o erro em regime permanente é eliminado, independentemente do sistema que se pretende controlar.

No entanto, o tempo de estabelecimento irá aumentar, piorando a estabilidade do sistema. Com isto será necessário reduzir o ganho proporcional, de modo a equilibrar a resposta do sistema.

Ação Proporcional-Derivativa

Este tipo de controlo é caracterizado pela expressão que destacamos na linha seguinte:

$$u(t) = K_p \cdot [e(t) + T_D \frac{de(t)}{dt}] \xrightarrow{L} U(s) = K_p \cdot (1 + T_D \cdot s) \quad (5)$$

Onde T_D é a designação de tempo derivativo e representa a antecipação da ação derivativa relativamente à ação proporcional. A ação derivativa age proporcionalmente à derivada do erro do sistema, e é responsável por melhorar o seu desempenho. Devido à dinâmica do processo existe um atraso entre a variação do sinal de controlo e a sua influência no sinal de saída. Posto isto, um controlador deste tipo pode ser interpretado como se o controlo atuasse proporcionalmente sobre a previsão do sinal de erro. Esta previsão é feita extrapolando a curva do erro, utilizando a sua tangente no instante de tempo t . Este tipo de controlo não pode ser usado sozinho, pois vai ser proporcional à taxa de variação do erro. A ação de controlo derivativa, quando adicionada a um controlador proporcional, propicia um meio de obter um controlador com alta sensibilidade. Uma vantagem de se usar o controlo em questão, é que este responde à taxa de variação do erro e pode produzir uma correção significativa antes do valor do erro se tornar demasiado grande. Assim, o controlo derivativo antecipa o erro e inicia uma ação corretiva mais cedo, tendendo a aumentar a estabilidade do sistema [3].

Ação Proporcional-Integral-Derivativa

Este tipo de controlo é a junção das três ações de controlo mencionadas anteriormente (P, PI e PD), e é caracterizado pela seguinte expressão:

$$u(t) = K_p \cdot [e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_D \cdot \frac{de(t)}{dt}] \xrightarrow{L} U(s) = K_p \cdot (1 + \frac{1}{T_i \cdot s} + T_D \cdot s) E(s) \quad (6)$$

Num controlador PID, o parâmetro integral é utilizado para eliminar o erro em regime permanente, proveniente de grandes variações dos parâmetros do sistema. O parâmetro derivativo, com o seu efeito estabilizador, provoca o aumento do ganho e a redução das oscilações, o que leva a uma resposta mais rápida quando comparado com o controlador P e PI [3].

Para facilitar a sintonia de um sistema, apenas com base no efeito que cada parâmetro produz na resposta, pode-se resumir esses mesmos efeitos na Tabela 1, usada no *robot* desenvolvido. No entanto, esta tabela só deve ser usada como referência, pois uma alteração num parâmetro produz efeitos nos restantes.

Ganho	Tempo de Subida (T_r)	Overshoot (M_p)	Tempo de Estab. (T_s)	Erro Regime Permanente (e_{ss})	Estabilidade
$\uparrow K_p$	Diminui	Aumenta	Aumenta Pouco	Diminui	Piora
$\uparrow K_i$	Diminui Pouco	Aumenta	Aumenta	Diminui Muito	Piora
$\uparrow K_d$	Diminui Pouco	Diminui	Diminui	Pouca Variação	Melhora

Tabela 1. Comportamento do sistema em função dos ganhos PID [3].

2. ARQUITETURA DO ROBOT

A arquitetura desenvolvida para o *robot* está representada através da Figura 2. Esta consiste num Arduino Uno, responsável pela aquisição de dados dos três sensores incluídos no *robot* e de acordo com as instruções de programação (em linguagem C/C++).

O Arduino Uno é uma placa de microcontrolador baseado no ATmega328. Dispõe de 14 pinos digitais de entrada/saída (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal de quartzo 16 MHz, um conector USB, um socket ICSP e um botão de *reset*.

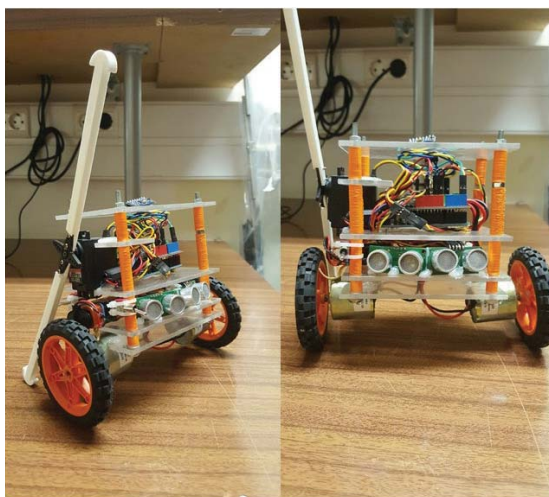
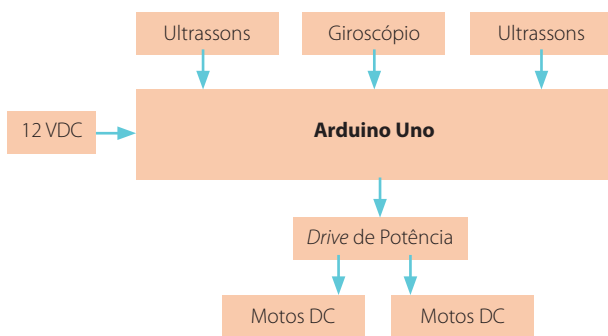


Figura 2. Arquitetura do *robot*.

Os sensores incluídos no *robot* são um giroscópio que é o elemento que vai dar a conhecer ao microcontrolador qual a posição em que o *robot* se encontra, sendo um elemento fundamental para a implementação do controlo PID por *software*. O giroscópio é um sensor que tem na sua saída valores que, depois de tratados, apresentam valores das três coordenadas (X, Y e Z). O giroscópio utilizado neste projeto é o GY-61; dois sensores ultrassons utilizados para detetar obstáculos no ambiente envolvente ao *robot*. O sensor de ultrassons utilizado é o SRF05.

No projeto são usados dois motores DC que permitem o movimento (direção e velocidade), segundo as instruções indicadas pelo Arduino Uno. O controlo dos motores é assegurado por um *drive* de potência L298, que possui uma arquitetura em quatro quadrantes, ou seja, ponte H de transístores, sendo um conversor de potência DC-DC. Assim, converte a tensão fixa que existe numa variável, podendo assim determinar o sentido da corrente, a polaridade da tensão e a tensão aplicada a um determinado componente. Esta ponte H permite não só utilizar as

duas direções como também, como já referido acima, controlar os motores por PWM, isto é, impor diferentes velocidades e sentidos.

As instruções desenvolvidas em linguagem de programação C/C++ resultam da implementação de um controlo PID, conjugando os valores adquiridos pelo giroscópio, erros atuais, diferenças de erros, erro acumulado, geração de sinais PWM e direção para cada motor DC. O projeto é alimentado através de baterias externas de 12 V com 900 mA. O processo funcional do *robot* está presente na Figura 3.

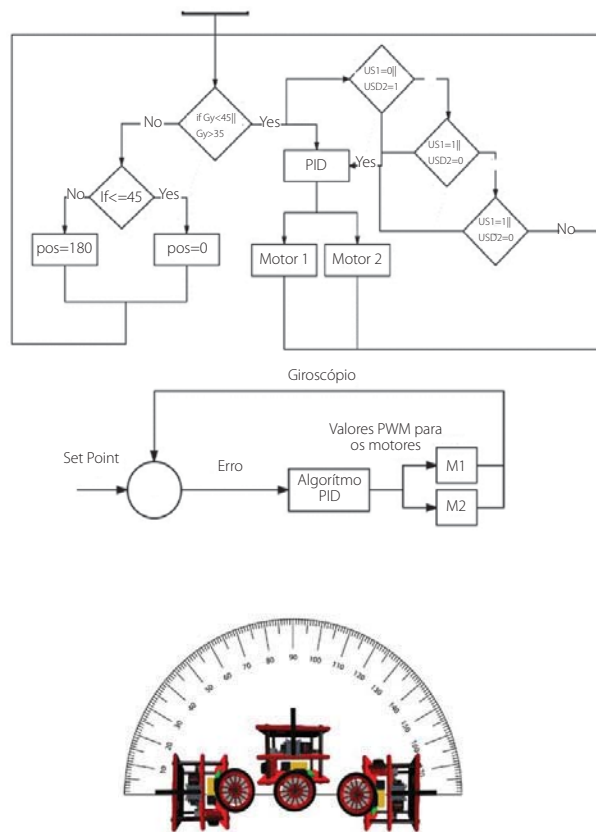


Figura 3. Fluxogramas desenvolvidos.

3. CONCLUSÃO

Este trabalho tinha como objetivo principal, a implementação do controlo PID para assegurar o equilíbrio do *robot*. Foram atingidos resultados bastante satisfatórios que garantiram a estabilidade, equilíbrio e locomoção do *robot*, mesmo perante obstáculos. Desenvolveram-se algoritmos implementando o controlo PID para o controlo da velocidade dos dois motores de forma independente, em função dos valores adquiridos pelo giroscópio.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] F. Castrillón, M. Osorio, R. Vasquez, "Comparison between different methods for tuning PID controllers", *Proceedings of the 2006 International Conference on Dynamics, Instrumentation and Control CDIC*. Querétaro, México, 2006.
- [2] C. Graham Goodwin, F. Stefan Graebe, E. Mario Salgado, "Control System Design", Prentice Hall, 2001.
- [3] André Santos, Projeto de Controladores PID Difusos Adaptativos, Tese de Mestrado, ISEP, setembro 2016.