

AJUSTE DE PARÂMETROS DE CONTROLE PARA UM MOTOR DE CORRENTE CONTÍNUA

Vinícius Augusto Bittencourt dos Santos¹; Anfranserai Morais Dias²

1. Bolsista PROBIC, Universidade Estadual de Feira de Santana, Graduando em Engenharia de Computação, email: vinigustobittencourt@gmail.com
2. Orientador, Universidade Estadual de Feira de Santana, Departamento de Tecnologia, email: anfranserai@ecomp.uefs.br

PALAVRAS-CHAVES: Teoria de Controle; Controle PD; Motor CC

INTRODUÇÃO

Um dos sistemas de sensoriamento utilizados no projeto “Construção de Uma Colônia de Robôs Autônomos para Reconhecimento, Busca e Inspeção”, é constituído por duas câmeras dispostas em forma de uma cabeça com dois olhos. Visando obter um sistema antropomórfico de visão estéreo, a cabeça será provida de cinco motores de corrente contínua (CC), cada um responsável por um movimento específico. O objetivo do projeto é controlar os movimentos de cada motor posicionado no pescoço e nos olhos. Para isso é necessário fazer uma modelagem deste sistema dinâmico em equações diferenciais para em seguida obter sua função de transferência, onde esta relaciona o sinal de saída ao sinal de entrada do sistema, e assim realizar o estudo do comportamento do mesmo.

O objetivo deste trabalho é desenvolver um algoritmo de controle para que este possa ser embarcado em um microcontrolador da família PIC18, cuja função é controlar os movimentos do pescoço e dos olhos (câmeras) do robô. O controle tem garantir que o sistema atue de forma rápida, e que as câmeras não oscilem quando estiverem na posição desejada.

Este relatório está dividido da seguinte forma, na seção Material e Métodos é apresentada a metodologia utilizada para obtenção do projeto do controlador, a arquitetura do projeto, juntamente com o desenvolvimento do algoritmo de controle e do *software* em JAVA. Na seção Resultados e Discussão, são apresentados os resultados iniciais da pesquisa, bem como, a análise dos mesmos. Por fim, são apresentadas as conclusões obtidas até o momento da pesquisa.

MATERIAL E MÉTODOS

A teoria de controle é aplicada em diversas áreas da engenharia, tais como: processos industriais, robótica, veículos espaciais, mísseis guiados, sistemas de pilotagem de aviões e até em nossas residências (CRAIG, 1989) (OGATA, 2003). Inicialmente, foi feita uma revisão bibliográfica do estado da arte, visando aperfeiçoar os conhecimentos na área de controle e robótica. Em seguida foi feita a modelagem da planta (sistema a ser controlado) em equações diferenciais, obtendo assim, seu modelo dinâmico. Foi aplicada a transformada de Laplace na equação diferencial, visando obter a relação da saída do sistema com sua entrada, ou seja, foi obtida a função de transferência do motor (DORF, 2001) (Ferreira, 2006).

Posteriormente, foi realizado o projeto de controle utilizando um controlador do tipo Proporcional-Derivativo (PD) por realimentação de velocidade. Os testes nesta fase foram realizados por meio de simulação no *software* MATLAB, a fim de validar o projeto e de

realizar a sintonia do controlador, ou seja, encontrar seus ganhos. O MATLAB foi utilizado por conter vários pacotes com rotinas prontas e consolidadas que facilitam a implementação de algoritmos com métodos numéricos e uma excelente saída gráfica. A equação de controle é dada por:

$$u(t) = K_p \cdot e(t) - K_d \cdot dy(t)/dt \quad (1)$$

onde, $u(t)$ é o sinal da ação de controle; $e(t)$ o sinal de erro; $y(t)$ a saída do sistema; K_p e K_d , são constantes (OGATA, 2003).

O próximo passo consiste em embarcar o algoritmo de controle simulado no microcontrolador. Na programação do algoritmo foi utilizado o ambiente de desenvolvimento PIC C COMPILER que é integrado com o compilador CCS (*Custom Computer Services*) que possibilita programar o *firmware* do microcontrolador em linguagem C.

Antes de embarcar o *firmware*, foi utilizado o *software* PROTEUS, que consegue simular um microcontrolador PIC realizando diversas funções, inclusive a de comunicação em entre o componente e o computador (*host*), implementando os protocolos USB e RS232. Também foi desenvolvido um *software* que funciona-se como supervisor no computador, que foi programado em linguagem JAVA. A Figura 1 mostra uma tela do programa supervisor.

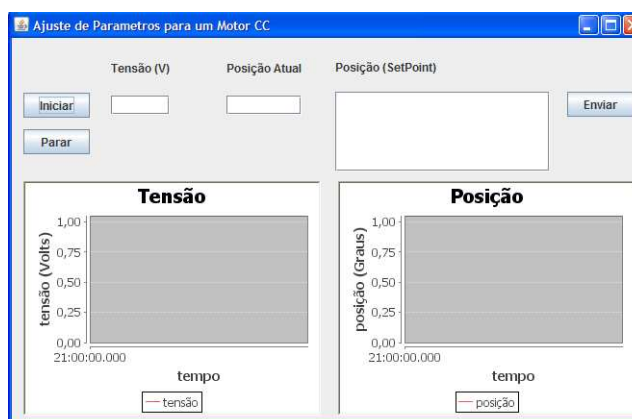


Figura 1: Software de supervisão.

Por fim, foi construído um protótipo para teste do algoritmo de controle. O protótipo consiste em uma placa de circuito impresso contendo um microcontrolador da família PIC18, um potenciômetro e um *driver* de potência conectado a um motor. O microcontrolador é responsável pela interface com o computador por meio da USB, e pela execução do algoritmo de controle. A Figura 2 ilustra a arquitetura do projeto.

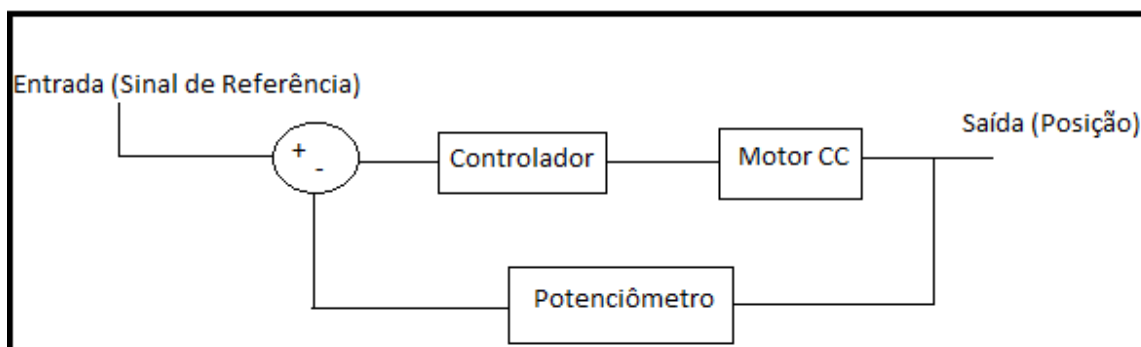


Figura 2: Arquitetura do projeto do microcontrolador.

A execução do teste começa com a escolha do valor de referência pelo usuário através do computador. Em seguida esse valor é transmitido ao protótipo por meio da USB, onde o algoritmo de controle calcula o valor de tensão a ser aplicado no motor. Então a tensão é aplicada e o ângulo de rotação do motor é medido pelo potenciômetro. Esse processo é repetido até que o valor de referência seja alcançado. O *software* supervisor apresenta as medidas obtidas com o potenciômetro durante a execução do controle. Na figura 3 pode-se visualizar o esquemático do projeto desenvolvido no PROTEUS.

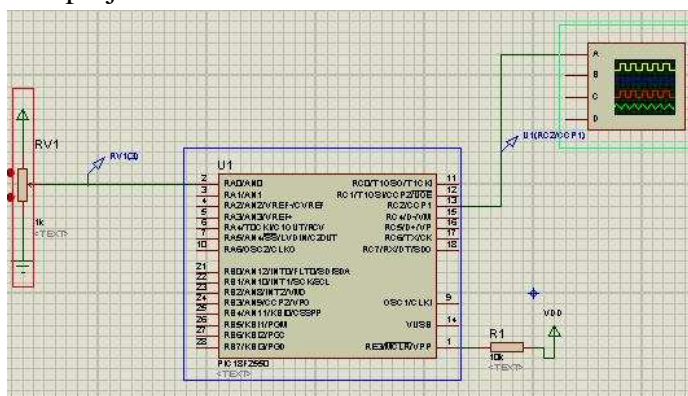


Figura 3: Esquemático do projeto de controle no PROTEUS.

O objeto destacado em vermelho na figura 3 é referente ao potenciômetro, em azul o microcontrolador da família PIC18 onde se embarca o algoritmo de controle e em verde o osciloscópio para observar a saída dada pelo microcontrolador a fim de averiguar seu estado.

RESULTADOS E DISCUSSÃO

Com base nas especificações de desempenho em termos do coeficiente de amortecimento (ζ) e da frequência angular (ω_n) do sistema, foi estabelecido que a primeira especificação tivesse como valor igual a 1 e o segundo como 91 rad/s. Utilizou-se ζ igual a 1, pois o sistema robótico não pode ter oscilação, portanto a melhor situação é que o sistema seja criticamente amortecido. Com relação a ω_n foi escolhido o valor 91 rad/s, para que a resposta do sistema seja bastante rápida, na faixa de 7 milissegundos. A Figura 4 representa o sistema controlado utilizando um controlador PD por realimentação de velocidade simulado no *software* MATLAB.

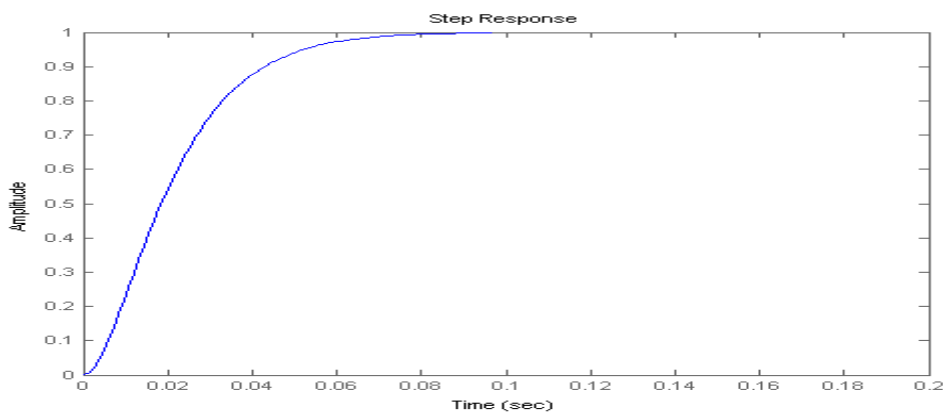


Figura 4: Gráfico da resposta do sistema mecânico com o controlador PD por realimentação de velocidade.

Pode-se verificar através da figura 4 que a resposta do sistema é satisfatória, visto que a resposta não apresenta oscilações e a mesma se estabiliza rapidamente.

CONCLUSÕES

O objetivo do projeto foi controlar um sistema robótico (Maxon, 2011). Partindo desse pressuposto foi feito um levantamento técnico de como se projetava um controlador de acordo com as especificações de desempenho do sistema. Em relação ao tipo de controlador, foram feitos estudos de acordo com as especificações, e foi escolhido um controlador do tipo proporcional-derivativo (PD) por realimentação de velocidade ao invés do só proporcional-derivativo, pois o segundo é suscetível a geração de instabilidade da resposta de desempenho, por este motivo optou-se pelo primeiro que se adequa melhor as especificações de projeto. Vale ressaltar que não foi utilizada a configuração PID, pois a planta a ser controlada já possuía um integrador na função de transferência.

Já foi desenvolvido o algoritmo de controle e o mesmo já está em fase de teste no microcontrolador. Portanto o próximo passo é finalizar o teste no *software* e realizar os últimos testes no microcontrolador.

REFERÊNCIAS BIBLIOGRÁFICAS

CRAIG, J. J. 1989. Introduction Robotics Mechanics of Control, Addison Wesley.

OGATA, K. 2003. Engenharia de Controle Moderno, Prentice Hall.

DORF, R. C.; BISHOP, R. H. 2001. Sistemas de Controle Modernos, LTC.

FERREIRA, P. A. V. 2006. Notas de Aulas de EA721 Princípios de Controle & Servomecanismo. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica & Computação.

MAXON 2011. Disponível em: <http://www.maxonmotor.com/>, Acessado em 3 de Maio de 2011.