



## CONTROLADOR LQR APLICADO À PLANTA DIDÁTICA TWIN ROTOR MIMO SYSTEM UTILIZANDO MATLAB E ARDUINO

Laura Ribeiro\*<sup>1</sup>, Luís Ricardo Cândido Côrtes<sup>1</sup>, Kenji Fabiano Ávila Okada<sup>1</sup>, Aniel Silva de Moraes<sup>1</sup>

<sup>1</sup>FEELT – Universidade Federal de Uberlândia

**Resumo** - O objetivo deste artigo é demonstrar o design de um controlador projetado pelo método de otimização, utilizando o Regulador Linear Quadrático (LQR, do inglês *Linear Quadratic Regulator*), onde o controlador é aplicado à uma planta didática Twin Rotor MIMO System (TRMS). Nessa planta, há a comunicação entre o *software* MATLAB e o Arduino. O documento está dividido em diferentes tópicos que descrevem o desenvolvimento do design e o processo para a criação do controlador. Ademais, há a descrição de conceitos importantes e metodologias, para a compreensão dos parâmetros do controlador. O desempenho do controlador foi verificado através de simulações e ensaios com a planta física.

**Palavras-Chave** – Twin Rotor MIMO System, Controlador, LQR, Espaço de Estados, Controle.

### LQR CONTROLLER APPLIED TO TWIN ROTOR MIMO SYSTEM TEACHING PLANT USING MATLAB AND ARDUINO

**Abstract** - The objective of this paper is to demonstrate the design of a controller designed by the optimization method using the Linear Quadratic Regulator (LQR), and the controller is applied to a Twin Rotor MIMO System (TRMS) teaching plant. In this plant, there is communication between MATLAB software and Arduino. The document is divided into different topics that describe design development and the driver creation process. In addition, there is a description of important concepts and methodologies for understanding the parameters of control. The performance of controller was selected through simulations and tests with a physical plant.

**Keywords** – Twin Rotor MIMO System, Controller, LQR, State Space, Control.

#### NOMENCLATURA

TRMS *Twin Rotor MIMO System.*

LQR *Linear Quadratic Regulator.*

PWM *Pulse Width Modulation.*

### I. INTRODUÇÃO

A evolução das técnicas de abordagens para a realização do voo adequado de veículos aéreos é evidente na atualidade. A modelagem dinâmica de veículos aéreos é complexa, principalmente devido ao efeito de acoplamento significativo entre os rotores e à indisponibilidade de alguns estados do sistema [1]. Com semelhança ao voo de um helicóptero, temos o Twin Rotor MIMO System (TRMS), desenvolvido pela *Feedback Instruments*, um sistema elaborado para a simplicidade.

O TRMS se encontra fixado a uma torre, a sua base possui liberdade para girar e rotacionar livremente nos planos horizontal e vertical, e em ambas as extremidades do feixe, existem rotores acionados por motores de corrente contínua.

Por possuir duas entradas e quatro saídas em seus estados, ele é um sistema com múltiplas entradas e múltiplas saídas (*Multiple Input, Multiple Output* – MIMO). Analisando para o lado de controle, possui acoplamento cruzado significativo. Esse acoplamento cruzado pode ser analisado pelas ações dos rotores, onde as posições angulares são influenciadas por cada rotor [2].

Existem diversos equipamentos para o estudo de controle, sejam educacionais e para ramos industriais. O Twin Rotor MIMO System [3], por exemplo, é de suma importância, sendo possível a visualização na prática de sistemas complexos com acoplamentos através de sistemas embarcados, tendo a comunicação com outros equipamentos utilizando softwares. Consequentemente, dessa forma é possível realizar o monitoramento do sistema.

O Regulador Linear Quadrático (LQR, do inglês *Linear Quadratic Regulator*) vem sendo largamente utilizado em diversos tipos de aplicações onde requer-se um controle ideal. Técnicas baseadas em controle com observadores, representados por espaço de estados, são aplicáveis a sistemas multivariáveis. Porém, há uma exigência clássica na localização dos polos e a garantia da estabilidade, independente se todas as variáveis são controladas ou não. Uma importante vantagem do uso dos controladores LQR está nos índices de desempenho ao atender os critérios de projeto; uma estratégia que pode reduzir esforços de controle e/ou manter os estados sobre controle, diminuindo o uso de energia do sistema [4].

\*lauraribeiroeng@gmail.com

## II. CONCEITOS E METODOLOGIA

Para a criação e desenvolvimento do controlador é necessário compreender o processo e as etapas que devem ser seguidas e consideradas, além dos conceitos sobre os procedimentos executados.

### A. Planta TRMS

O Twin Rotor MIMO System é uma planta que pode ser dividida em diferentes sistemas. Na Figura 1, temos a representação simbólica do modelo real e físico do TRMS utilizado no projeto do controlador [3].

O sistema físico é composto por um eixo fixo em uma torre vertical, com duas hélices acopladas a dois motores posicionadas nas extremidades do eixo. Um dos motores é responsável pelo deslocamento vertical em relação ao eixo transversal da torre, e outra pelo deslocamento horizontal em relação a torre. Para questões de equilíbrio do sistema, é utilizando um contrapeso perpendicular ao eixo.

Figura 1: Representação Simbólica do TRMS.



O sistema de sensoriamento é composto por dois *encoders* - dispositivos e/ou sensores que possuem a capacidade de transformar a posição em sinal elétrico digital [5] - localizados na torre. Um deles é responsável por mensurar a posição angular vertical chamado de *Psi* e outro pela posição angular horizontal chamado de *Phi*. Esses sensores são de natureza incremental, logo é preciso estabelecer um ponto de referência do sistema. Essa referência foi estabelecida posicionando o eixo no seu limite máximo horizontal e vertical. Além disso, existem dois tacômetros acoplados nos rotores, com a função de medir as velocidades angulares

### B. Espaço de Estados

O uso da representação de espaço de estados no sistema faz-se necessário devido ao tipo do mesmo. Para a descrição de um sistema dinâmico complexo, de característica MIMO, é exigido, de uma maneira geral, equações diferenciais não-lineares de ordem elevada [6]. As soluções reais dessas equações podem ser raras e complexas, levando a consideração de equações e representações das equações em que possam simplificar o problema.

A representação por Espaço de Estados pode ser definida como um modelo matemático de um sistema físico composto

de um conjunto de variáveis de entrada, de saída e de estado relacionadas entre si por meio de equações diferenciais de primeira ordem. Essas equações podem ser identificadas pelas Equações 1 e 2, de forma genérica. Na Figura 2, há a representação esquemática.

A Equação 1 é a Equação dos Estados.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1)$$

Sendo:

- $x(t)$  - Estados.
- $u(t)$  - Entradas
- $A$  - Matriz de Estados
- $B$  - Matriz de Entrada

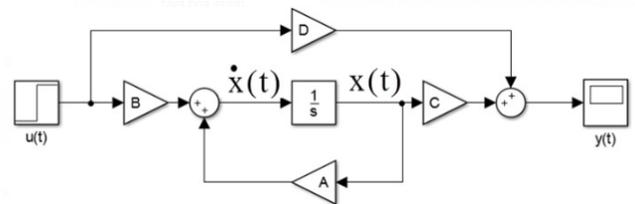
A Equação 2 é a Equação da Saída.

$$y(t) = Cx(t) + Du(t) \quad (2)$$

Onde:

- $y(t)$  - Saídas
- $u(t)$  - Entradas
- $C$  - Matriz de Saída
- $D$  - Matriz de Transmissão Direta

Figura 2: Representação Esquemática do Espaço de Estados



### C. Realimentação por Estados

O controle por realimentação dos estados é definido por uma variável  $K$ , sendo ela uma matriz de ganhos com o número de linhas iguais o número de entradas do sistema e o número de colunas iguais a quantidade de estados do sistema.

No controlador por realimentação dos estados a entrada do sistema é calculado em função dos estados do sistema multiplicados por uma matriz de ganhos constante, utilizaremos funções do *software* MATLAB que através do modelo do sistema em espaço de estados é possível a geração dos ganhos para serem empregados no controle.

### D. Observador de Estados

Nem sempre em todos os casos será possível mensurar certos parâmetros de plantas físicas, como por exemplo, a posição, velocidade, temperatura, dentre outros. E isso pode ser explicado e provocado por falhas de sensores e/ou componentes danificados pela ação do tempo ou de usuários, por exemplo. Para o controle, essa falta de informações é um grande ponto negativo, nesses casos não há como realizar de forma satisfatória o controle destas plantas.

Para solucionar essa adversidade, aplica-se o conceito do observador.

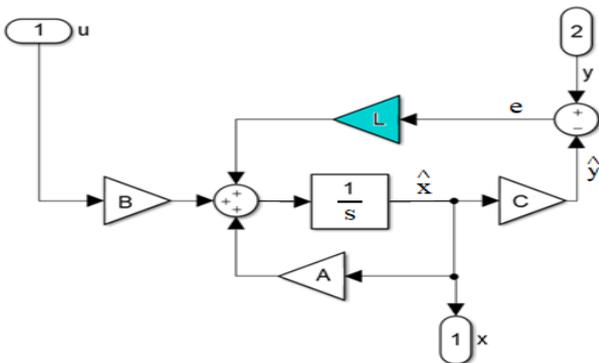
O observador de estados tem como função realizar o ajuste das informações nos estados do sistema. Estas informações podem estar ligadas diretamente aos próprios estados do sistema.

O observador é capaz de realizar a comparação do modelo teórico do sistema com o modelo físico, onde há a análise do erro por meio de uma matriz de ganhos, denominada L, que multiplica a diferença entre  $y$  e  $\hat{y}$ .

A matriz L então é inserida ao modelo teórico e tende a ser ajustada até que o erro se torne praticamente nulo, consequentemente, teremos como saída do observador, estados estimados que podem ser utilizados na realimentação do sistema.

Na Figura 3, visualiza-se a montagem de um observador de estados no *Simulink*.

Figura 3: Observador de Estados.



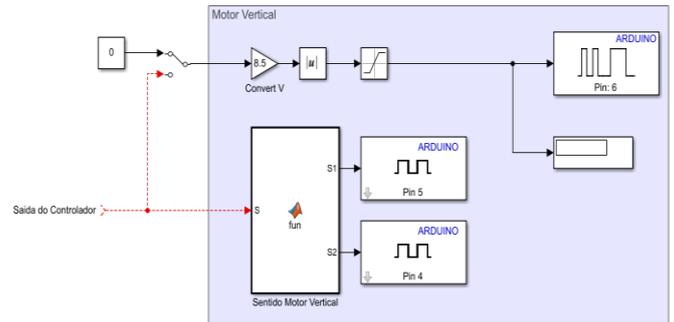
### E. União Arduino e MATLAB

A corporação americana *MathWorks*, que desenvolveu o *software* MATLAB, também desenvolveu um pacote capaz de interagir com a maior parte das placas Arduino conhecidas, possibilitando estabelecer uma comunicação USB com as mesmas.

O pacote desenvolvido nos permite controlar na plataforma Arduino: DI/DO, AI/AO, PWMs entre outras portas de entrada e saída digital, entradas do conversor analógico-digital e saídas de sinal modulado por largura de pulso. Além disso, permite usufruir dos demais blocos e funções, tanto do MATLAB quanto do *Simulink*.

Na Figura 4, há a representação esquemática da ferramenta, com as funções necessárias para a comunicação entre as plataformas e interfaces distintas. Todos os recursos abordados foram utilizados no projeto do controlador.

Figura 4: Pacote MATLAB/Arduino implementado para o controle do motor vertical do TRMS.



## III. DESENVOLVIMENTO

Nesta sessão, é descrito com detalhes o processo de elaboração e implementação do controlador LQR junto a planta didática TRMS, além de sua atribuição. O processo foi executado utilizando a ferramenta do *Simulink* e os procedimentos detalhados são referentes a mesma.

### A. Modelo em Espaço de Estados

Em posse dos parâmetros físicos da planta didática TRMS, 6 estados foram definidos para o sistema, sendo eles o  $\psi$  e o  $\phi$ , as suas derivadas  $d\psi$  e  $d\phi$ , e os torques como  $T1$  e  $T2$ .

Esses parâmetros podem ser definidos por equações de estado, onde para cada parâmetro há uma equação.

Na Figura 5, as definições das equações de estado para o processo do projeto do controlador podem ser visualizadas. Essas definições foram programadas na ferramenta comentada.

Figura 5: Equações de Estado

```

%% Definindo as equações de cada estado
% Estado x1 = psi
f_x1 = dpsi;

% Estado x2 = dpsi
f_x2 = M_vt/I_v;

% Estado x3 = phi
f_x3 = dphi;

% Estado x4 = dphi
f_x4 = M_ht/I_h;

% Estado x5 = T1
f_x5 = (k_1*v1-T_10*T1)/T_11;

% Estado x6 = T2
f_x6 = (k_2*v2-T_20*T2)/T_21;

```

Utilizando o *software* MATLAB e definindo os pontos de operação para os parâmetros iniciais do sistema, torna-se viável a montagem de quatro matrizes que representam o TRMS em espaço de estados no tempo contínuo. Essas quatro matrizes são as matrizes citadas nas Equações 1 e 2: A, B, C e D - Matriz de Estados, Matriz de Entrada, Matriz de Saída e Matriz de Transmissão Direta, respectivamente.

Figura 6: Código para a Obtenção das Matrizes em Espaço de Estados

```

%%
% Pontos de operação do sistema
psi = 0;
dpsi = 0;
phi = 0;
dphi = 0;
T1 = 0;
T2 = 0;

% Construindo as matrizes do espaço de estados do modelo

A = [a11 a12 a13 a14 a15 a16;
     a21 a22 a23 a24 a25 a26;
     a31 a32 a33 a34 a35 a36;
     a41 a42 a43 a44 a45 a46;
     a51 a52 a53 a54 a55 a56;
     a61 a62 a63 a64 a65 a66];

A = subs(A);
A = vpa(A,5);
A = double(A);

%%
B = [zeros(1,size(A,1)-2) b51 b61;
     zeros(1,size(A,1)-2) b52 b62]';

B = subs(B);
B = vpa(B,5);
B = double(B);

%%
C = [1 0 0 zeros(1,size(A,1)-3) ;
     0 0 1 zeros(1,size(A,1)-3)];

%%
D = zeros(size(C,1),size(B,2));

```

Figura 7: Modelo em Espaço de Estados Discreto.

```

A =
    x1      x2      x3      x4      x5      x6
x1  0.9999  0.009994  0  1.054e-05  -7.887e-05  5.385e-08
x2 -0.01965  0.9989  0  0.002109  -0.01574  1.614e-05
x3  5.59e-07  -8.53e-05  1  0.01  2.796e-05  7.654e-05
x4  0.0001677  -0.01706  0  1  0.005628  0.01528
x5  0  0  0  0  0.991  0
x6  0  0  0  0  0  0.99

B =
    u1      u2
x1 -2.631e-07  1.078e-10
x2 -7.887e-05  4.308e-08
x3  9.29e-08  2.043e-07
x4  2.796e-05  6.123e-05
x5  0.009955  0
x6  0  0.00796

C =
    x1  x2  x3  x4  x5  x6
y1  1  0  0  0  0  0
y2  0  0  1  0  0  0

D =
    u1  u2
y1  0  0
y2  0  0

Sample time: 0.01 seconds
Discrete-time state-space model.

```

Adquirindo as matrizes que representam o modelo do TRMS em espaço de estados contínuo, esse modelo é realizável para o modelo de espaço de estados em tempo discreto, onde é utilizado um tempo de “discretização” de 0.01 segundos. Passando o modelo para o tempo discreto, podemos manter a constante o tempo de amostragem do sistema, permitindo a observação da resposta e o comportamento que este controle fornece. O controlador, dessa maneira, pode ser ajustado conforme as especificações.

## B. Design do LQR

Referindo-se ao modelo do sistema em espaço de estados no tempo discreto, a função *DQLR* é aplicada no *software*. Essa função é capaz de selecionar o modelo discreto de um sistema e aplicar o *LQR* para gerar os ganhos do controlador. Essa função requer um argumento inicial, sendo ele as matrizes A e B do modelo da planta didática e outras duas matrizes diagonais. Essas matrizes (que recebem os nomes de Q e R) são compostas por pesos e são responsáveis por refinar o controle.

Figura 8: Processo para a Obtenção do Controlador.

```

sys_i = ss(A,B,C,D);
sys_id = c2d(sys_i,Ts);

A_i = [ eye(size(sys_id.C,1)) sys_id.C*sys_id.A;
        zeros(size(sys_id.A,1),size(sys_id.C,1)) sys_id.A ];

B_i = [sys_id.C*sys_id.B ; sys_id.B];

C_i = [zeros(size(sys_id.C,1),size(sys_id.C,1)) sys_id.C ];

Q_i = [1 0 0 0 0 0 0 0;
       0 1 0 0 0 0 0 0;
       0 0 0.00001 0 0 0 0 0;
       0 0 0 1 0 0 0 0;
       0 0 0 0 1 0 0 0;
       0 0 0 0 0 1 0 0;
       0 0 0 0 0 0 1 0;
       0 0 0 0 0 0 0 1];

R_i = [1000 0;
       0 1000];

[K_i,S_i,E_i] = dlqr(A_i,B_i,Q_i,R_i);

```

A Figura 8 detalha o código para obtenção do controlador utilizando o método LQR. A matriz Q está relacionada aos estados do sistema, ela é ajustada modificando os valores significativos dos pesos, conforme a necessidade observada durante o processo. A matriz R está conectada com as entradas do modelo, onde é notável que a matriz está com os mesmos pesos em sua diagonal – justificado, pois não houve a necessidade de se dar mais peso, ou modificar o seu valor, para relacionar com a entrada do sistema.

As matrizes utilizadas como argumento de entrada para a função *DQLR* estão na forma de matrizes aumentadas. As dimensões foram alteradas por meio do método de estados aumentados, recebendo assim dois novos estados no sistema, onde há um estado para cada integrador presente nas entradas do sistema.

Nas Figuras 9 e 10, estão visíveis as respostas retornadas pelo uso da função *DQLR*.

Figura 9: Ganhos do Controle

```

K_i =
    -0.0223  0.0218  0.0680  -3.8250  3.6348  2.3123  2.9103  1.1687
    0.0221  0.0221  3.8384  -0.9600  3.9959  3.9682  0.9397  2.6730

```

A variável *K\_i* é um vetor de duas linhas e oito colunas que indica os ganhos utilizados para o controle. Cada linha se refere à uma entrada do sistema e cada coluna à um estado do modelo, onde as duas primeiras colunas são para os estados

aumentados e as outras seis colunas se referem aos estados iniciais do modelo.

Figura 10: Polos do Sistema

$$E_i = \begin{matrix} 0.9955 + 0.0182i \\ 0.9955 - 0.0182i \\ 0.9866 + 0.0037i \\ 0.9866 - 0.0037i \\ 0.9884 + 0.0071i \\ 0.9884 - 0.0071i \\ 0.9939 + 0.0106i \\ 0.9939 - 0.0106i \end{matrix}$$

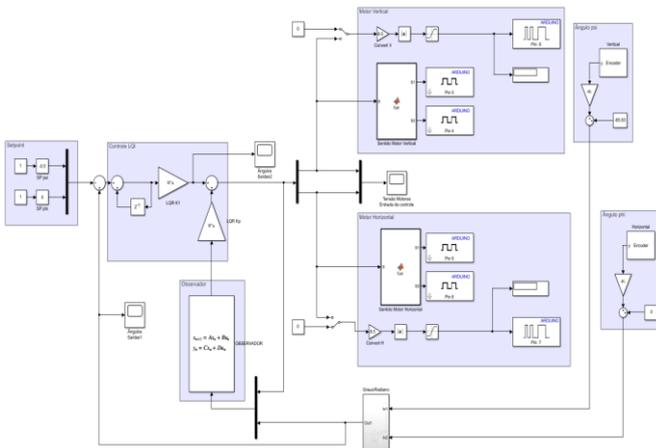
A variável  $E_i$  representa um ponto interessante do método LQR. Esse ponto é formado pelos polos do sistema em malha fechada, garantindo que o sistema será estável.

### C. Interface do Controle

A planta TRMS foi readaptada para funcionar com diversas técnicas de controle utilizando apenas uma interface criada no *Simulink* e a plataforma Arduino. Essas adaptações se dão para adquirir dados de sensores, provenientes dos *encoders* vertical e horizontal, e enviar dados como o sinal PWM para os motores presentes nas hélices do TRMS.

A Figura 11 exibe a interface de controle implementada no *Simulink*.

Figura 11: Representação da Interface do Controle



Nessa interface, o usuário fornece inicialmente os valores de pontos de operação (*setpoint*) no sistema, sendo eles: os ângulos  $Psi$  (referente ao movimento vertical do TRMS) e  $Phi$  (referente ao movimento horizontal). Sequencialmente, o controle do sistema é acionado, coletando essas informações e jogando-as para os motores do TRMS, através de um sinal PWM que controla a velocidade e o sentido de giro de cada motor.

Os *encoders* presentes nas torres do TRMS enviam a posição em que cada hélice se encontra, e a informação coletada é convertida para radianos - devido aos ângulos do

modelo original do TRMS estarem nessa unidade de medida - e enviada ao observador de estados.

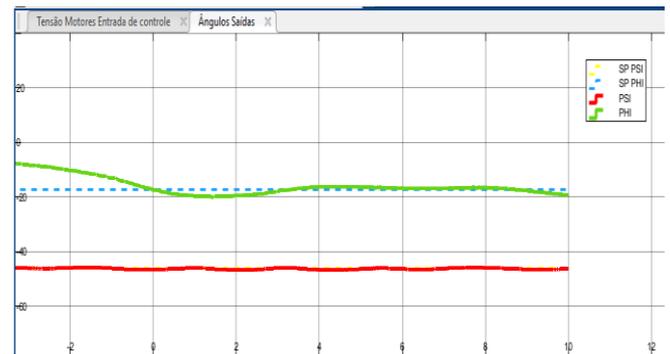
O observador de estados fará a comparação entre a saída do modelo teórico e do modelo físico do TRMS, gerando um sinal de erro entre os modelos. O sinal gerado ajusta os estados do sistema para que os mesmos sejam o mais próximo possível do teórico, reduzindo o sinal de erro ao máximo. Os estados estimados (estados resultantes) são enviados para a realimentação que contém os ganhos obtidos pelo método LQR.

Consequentemente, essa informação é comparada com o *setpoint*, gerando assim o erro do sistema. O processo se repete ininterruptamente até que o erro do sistema tenda a ser nulo ou o mais próximo de zero.

## IV. RESULTADOS E DISCUSSÕES

Para a validação do sistema, foram executados testes na planta com os ângulos de saída do sensoriamento pelos *encoders*, além de observar se a mesma seguiria o *setpoint* proposto. A Figura 12 expressa os sinais obtidos com a implementação do controlador pelo método LQR à planta didática TRMS.

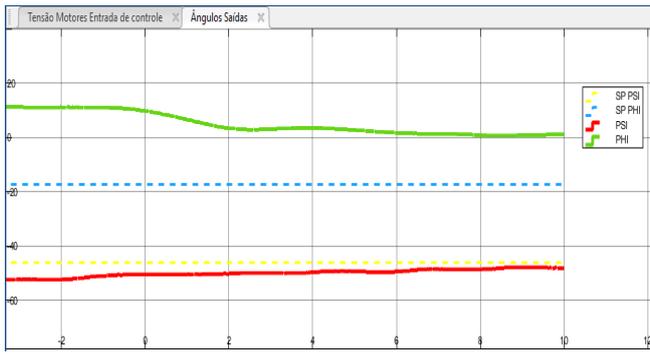
Figura 12: *SetPoint* e Ângulos de Saída do TRMS



Uma consideração a ser feita a partir dos resultados obtidos, é sobre o acoplamento presente entre os ângulos dos eixos vertical e horizontal. Através do controlador projetado, no momento em que há a mudança de *setpoint* de uma das variáveis, ambas as saídas reagirão à variação gerada no erro de realimentação. Para esse sistema de controle, a interferência do ângulo  $psi$  no ângulo  $phi$  é consideravelmente maior que o inverso. Neste caso, o comportamento observado nos sinais de saída, no momento da mudança de *setpoint* da variável  $psi$ , é a variação do ângulo  $phi$  à uma região fora do seu *setpoint* até o momento em que o ângulo  $psi$  atinge o regime permanente. Neste instante, o controlador força o ângulo  $phi$  a retornar ao seu estado anterior em regime permanente.

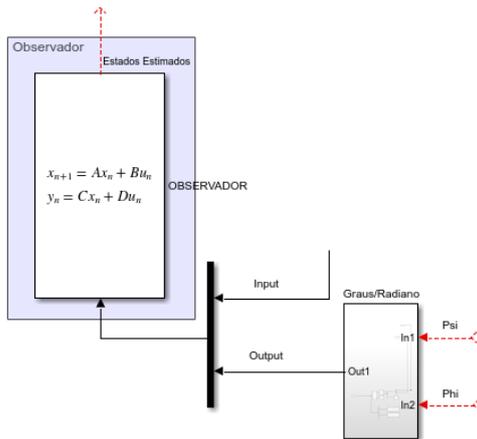
Esse efeito descrito pode ser observado na Figura 13.

Figura 13: Interferência nos Motores



Em relação ao controle, nota-se necessário o uso de um observador de estados para obter uma resposta com um comportamento semelhante e mais fiel do sistema. Originalmente, o TRMS possuía tacômetros em seus motores que fazem o sensoriamento da velocidade, porém na planta real que foi desenvolvido esse projeto, os mesmos se encontram queimados.

Figura 14: Observador de Estados Implementado



Quando o sistema físico possui estados não mensuráveis (por um motivo indefinido) e tem-se um modelo teórico da planta, o observador faz a comparação entre esses modelos e compensa as informações que não estão presentes no modelo físico. Sendo assim, o observador de estados se comporta de maneira bastante eficaz, justificando seu uso.

## V. CONCLUSÕES

O objetivo do artigo foi atingido com sucesso, já que o design do controlador projetado pelo método de otimização, utilizando um controlador LQR aplicado à uma planta didática Twin Rotor MIMO System (TRMS) foi demonstrado e comprovado o seu funcionamento, executando uma boa performance.

Conclui-se que controlador LQR em relação à atividade que lhe é imposta, é capaz de garantir um bom desempenho e manter o sistema estável em malha fechada. O método de otimização, através do ajuste das matrizes Q e R, procura e/ou busca o fornecimento de uma solução ótima às equações diferenciais do sistema.

O sistema, por fim, apresentou-se controlável e com variações não comprometedoras, gerando possibilidades para trabalhos futuros melhoria e/ou aprimoramento do controle obtido, de modo a deixar mais imperceptíveis as variações existentes, ou ainda, há aplicações de outros possíveis métodos e projetos de controladores.

## AGRADECIMENTOS

Este projeto foi financiado pelo CNPq (processo 131505/2019-0). Os autores agradecem pelo apoio e pelo empréstimo do espaço físico do Laboratório Aterramento Elétrico, Máquinas e Processamento de Energia (LAMEP), da Universidade Federal de Uberlândia (UFU), para a realização do trabalho.

## REFERÊNCIAS

- [1] M. Ilyas, "Control Law Design for Twin Rotor MIMO System with Nonlinear Control Strategy," *Discrete Dynamics in Nature and Society*, vol. 2016, p. 10, 2016.
- [2] F. I. Ltd., "Twin Rotor MIMO System Control Experiments," [Online]. Disponível: <http://www.cpdee.ufmg.br/~palhares/33-942rotor.pdf>. [Acesso em Julho 2019].
- [3] F. LEYBOLD, "Set Twin Rotor MIMO System," [Online]. Disponível: <https://www.leybold-shop.com/set-twin-rotor-mimo-system-33-007-pci.html>. [Acesso em Julho 2019].
- [4] KANIESKI, J. M., "Modelagem E Controle Lqr Aplicado A Um Condicionador De Energia", 2010, [Online]. Disponível: [http://repositorio.utfpr.edu.br:8080/jspui/bitstream/1/224/1/PB\\_PPGEE\\_M\\_Kanieski%2C%20Jo%C3%A3o%20Marcos\\_2010.pdf](http://repositorio.utfpr.edu.br:8080/jspui/bitstream/1/224/1/PB_PPGEE_M_Kanieski%2C%20Jo%C3%A3o%20Marcos_2010.pdf). [Acesso em Agosto 2019].
- [5] F. Almeida, "O que é Encoder? Para que serve? Como escolher? Como interfacear?," *HI Tecnologia*, 2017. [Online]. Disponível: <https://www.hitecnologia.com.br/blog/o-que-%C3%A9-encoder-para-que-serve-como-escolher-como-interfacear/>. [Acesso em Agosto 2019].
- [6] A. D. T. FLEURY e D. C. DONHA, "PME 3481 - CONTROLE E APLICAÇÕES," *HI Tecnologia*, 2017. [Online]. Disponível: <http://sites.poli.usp.br/d/PME2472/aulatres.pdf>. [Acesso em Agosto 2019].