

# Identificação com uma Variação de Mínimos Quadrados Recursivo aplicado a um Manipulador Robótico

Darielson A. Souza\* Josias G. Batista\*  
Laurinda L. N dos Reis\* José N. N. Júnior\*  
José R. O. Júnior\* Kaio M. Ramos\* Antônio B. S. Júnior\*\*

\* Grupo de Pesquisa em Automação, Controle e Robótica (GPAR),  
Universidade Federal do Ceará, Fortaleza - CE, Brasil, (e-mail:  
darielson@dee.ufc.br).

\*\* Instituto Federal do Ceará-IFCE, Maracanaú - CE, Brasil.

**Abstract:** Robotics has been reaching a lot of space in recent years and is evolving every day. Today, robotics is seen as an important area in many applications. Dynamic control is an essential tool for robotics. To design the control, a well-modeled / identified plant is required to have a well-defined controller design. A misidentified system can compromise the project, causing it to have a control effort and a very long tuning time, which is not good at all. The present work aims to show some solutions for the identification of a robotic manipulator with a hybrid system of a recursive least squares technique and a metaheuristic called differential evolution (DE). The results are compared with other methods and can surpass them in order to validate the study and showing its potential.

**Resumo:** A robótica vem alcançando muito espaço nos últimos anos e está evoluindo a cada dia. Atualmente, a robótica é vista como uma área importante em muitas aplicações. O controle dinâmico é uma ferramenta essencial para a robótica. Para projetar o controle, é necessária uma planta bem modelada/identificada para ter um projeto de um controlador bem definido. Um sistema mal identificado pode comprometer o projeto, fazendo com que ele tenha um esforço de controle e um tempo de sintonia muito longo, o que não é nada bom. O presente trabalho tem como objetivo mostrar algumas soluções para uma identificação de um manipulador robótico com um sistema híbrido de uma técnica dos mínimos quadrados recursivos (MQR) e uma metaheurística denominada evolução diferencial (DE). Os resultados são comparados com outros métodos e conseguem superá-los, a fim de validar o estudo e mostrando o seu potencial.

**Keywords:** Optimal identification; Mechanical system; Computational intelligence; Optimization; Cylindrical Manipulator.

**Palavras-chaves:** Identificação ótima; Sistema mecânico; Inteligência computacional, Otimização; Manipulador Cilíndrico.

## 1. INTRODUÇÃO

A área da robótica é muito promissora, tem grande relevância na maioria das aplicações recentes, mas as técnicas de controle têm sido constantemente testadas (Pinto et al., 2014). Os fundamentos do controle na robótica são de suma importância para que as tarefas sejam realizadas com sucesso. Os manipuladores robóticos industriais realizam muitas trajetórias repetitivas. As principais características dos robôs manipuladores nas indústrias são: velocidade, torque e precisão, esses atributos são essenciais para um bom funcionamento (Alenany and Shang, 2013).

Uma das características de um manipular robótico em geral é a presença de uma garra no final do seu elo, e é projetada para os componentes serem transportados levando em conta as características do material, como peso e formato, e deve ser ajustado para manipular esses objetos. Um dos problemas do controle dinâmico é a má

identificação/modelagem, causando um grande esforço de controle, podendo causar danos à planta. Para isso, este trabalho tem a contribuição de apresentar um sistema híbrido utilizando o método de mínimos quadrados recursivo com uma evolução diferencial (DE) (Storn and Price, 1997) para ponderar a matriz  $P$  dos mínimos quadrados recursivo.

O método proposto será comparado com outros de identificação, como mínimo quadrado recursivo clássico e um Filtro de Kalman. Os métodos são comparados e avaliados pelo coeficiente de correlação múltipla, onde as funções de transferência geradas por cada método serão testadas.

Este trabalho é estruturado por seções, a primeira é a introdução que apresenta uma visão geral sobre o trabalho uma contextualização. A seção seguinte mostrará os detalhes técnicos do manipulador robótico, que foi utilizado. Na Seção 3 são apresentadas as identificações do sistema. A Seção 4 mostra os resultados e discussões, contendo

gráficos e tabelas para ser avaliado e comparado, e finalmente a seção seguinte demonstra as conclusões acerca dos resultados e pode ser uma contribuição para outras obras que possam surgir.

## 2. DETALHES TÉCNICOS

Existem vários tipos de manipuladores usados em diversas aplicações de acordo com sua geometria, porém o que foi utilizado na proposta foi um manipulador robótico do tipo cilíndrico. O manipulador cilíndrico tem 3 graus de liberdade, o primeiro grau é a base que movimenta rotativos, o 2º grau é linear que é o tronco que faz os movimentos na vertical, o terceiro grau é o que faz os movimentos na horizontal, e uma garra que também pode ser chamada de efeito final ou *end-effector* (Rebouças et al., 2017).

A Figura 1 apresenta o manipulador utilizado no trabalho, o mesmo tem as juntas acionadas por motores de indução trifásicos. Para o acionamento do manipulador, foi utilizado um microcontrolador Processador Digital de Sinais (DSP) da Texas Instruments®. A principal vantagem na utilização do DSP, é que, além do alto desempenho de processamento, com capacidade de realizar 150 MPIS (milhões de instruções por segundo), pode ser implementado uma modulação espacial vetorial, SVPWM. A interface de programação é o CCS, que será implementado em C.

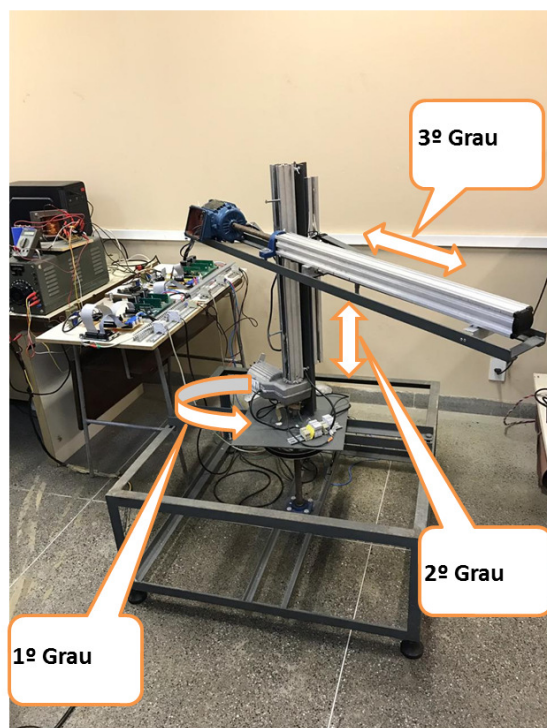


Figura 1. Manipulador robótico cilíndrico.

O sistema robótico possui 3 graus de liberdade, porém foi preparada apenas a aquisição de dado da 1ª junta (rotativa). Foi realizada também a aquisição dos comportamentos mecânicos e elétricos para fazer a identificação. Especificando a coleta, será colhido a corrente e a velocidade

durante o acionamento da 1ª articulação. O objetivo é criar um modelo que satisfaça o comportamento mecânicos da mesma.

A Figura 2 apresenta o comportamento a ser trabalhado da planta, onde tem-se a corrente como entrada e como saída a velocidade. O tempo total foi de 1000 s, logo o conjunto de dados possui 1000 rótulos (entrada/saída).

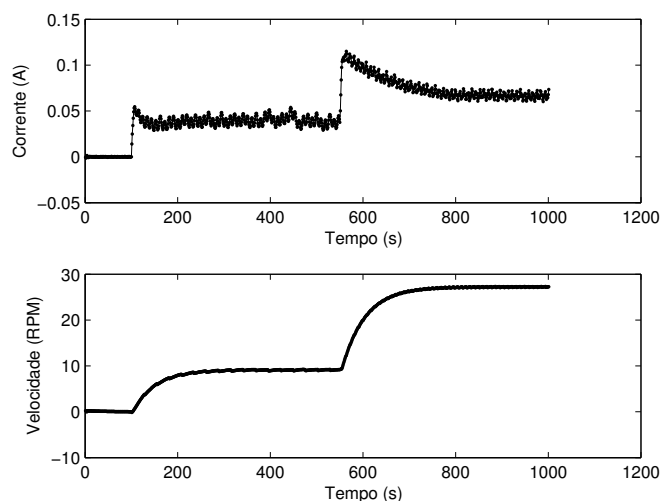


Figura 2. Entrada e saída real da planta.

## 3. IDENTIFICAÇÃO DE SISTEMAS

A maioria dos processos são não-lineares e variam no tempo, os modelos lineares são frequentemente empregados para modelar e controlar tais processos. À medida que as condições operacionais mudam, é importante que o modelo e suas características sejam usados para benefícios do controle baseado em modelos avançados (Coelho, 2004). Logo é interessante fazer uma formulação adequada dos algoritmos para proporcionar procedimentos eficientes.

Além do método proposto será apresentado outros como uma forma de uma validação do mesmo. Porém a maneira de preparar os dados para a identificação não mudam. Os dados a serem identificados são dados reais coletados: corrente na entrada e velocidade na saída. A posição do rótulo (entrada e saída) será embaralhada, para evitar *overfitting*. Após o conjunto de dados será dividido em 80% para treinamento e 20% para testes.

### 3.1 Mínimos Quadrados Recursivo

O algoritmo dos mínimos quadrados recursivo (MQR) tem uma grande vantagem em relação aos demais devido a sua velocidade pela maneira de trabalhar em lote. Uma das desvantagens é ponderar a matriz de covariância  $P$  durante o processo de identificação, Coelho (2004), (Haykin, 2002). A Figura 3 apresenta como a identificação é desenvolvida, ou seja, sempre o resultado algoritmo é comparado com os parâmetros reais coletados da planta de forma paralela.

Para cada período de amostragem, novas medidas tornam-se disponíveis e são utilizadas com o modelo atual para gerar

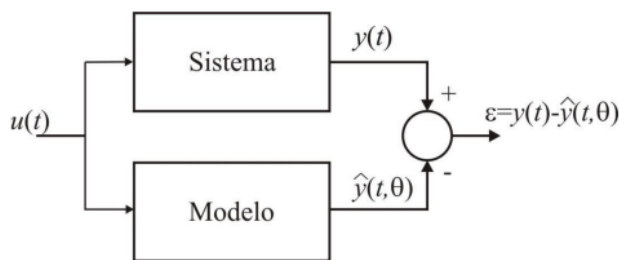


Figura 3. Identificação por MQR.

um novo erro de modelagem. Os passos do algoritmo do MQR pode ser demonstrado no Algoritmo 1.

**Algoritmo 1 MQR**

```

Inicializar valores de:  $\theta, \varphi$  and  $P$ ;
for i = 1 até tamanho (y) do
    e = y(t) -  $\varphi_{(t)}^T \theta_{(t-1)}$ 
     $K_{(t)} = P_{(t-1)} \varphi_{(t)} [I + \varphi_{(t)}^T P_{(t-1)} \varphi_{(t)}]^{-1}$ 
     $P_{(t)} = [P_{(t-1)} - K_{(t)} \varphi_{(t)}^T P_{(t-1)}]$ 
     $\theta_{(t)} = \theta_{(t-1)} + K_{(t)} e$ 
end for
    
```

onde  $e$  corresponde o erro;  $K$  é o ganho;  $P$  a matriz de covariância; e  $\theta$  é a equação para calcular do vetor de paramentos.

**3.2 Variação dos Mínimos Quadrados Recursivo com DE**

Como mencionado anteriormente a ponderação da matriz  $P$  dos mínimos quadrados recursivo pode ser um problema ao ser inicializada no momento da identificação. A metaheurística DE vai sanar esse problema ao tentar buscar a matriz  $P$  que satisfaça a uma função objetiva que seria minimização da raiz do erro quadrático médio (RMSE) do MQR, conforme mostra (1).

$$\min_{RMSE} = \sqrt{\frac{\sum_{i=1}^T (\hat{y}_i - y_i)^2}{T}}, \quad (1)$$

Onde  $\hat{y}$  é a previsão,  $y$  é a variável observada e  $T$  é o número de iterações.

O algoritmo evolucionário DE, proposto por Storn e Price em 1997 (Storn and Price, 1997), é um método heurístico para otimizar funções espaciais contínuas não diferenciáveis. Na computação evolucionária, o DE é um método que otimiza um problema melhorando iterativamente uma solução candidata com base em uma determinada medida de qualidade. Esses métodos são comumente conhecidos como metaheurísticas, pois fazem pouca ou nenhuma suposição sobre a função que está sendo otimizada e podem procurar um espaço muito amplo de soluções candidatas.

O Evolução diferencial (DE) é usado para funções multi-dimensionais de valor real. Essa heurística, ao contrário de alguns métodos clássicos de otimização, como gradiente e quase-newton, não exige que o problema a ser otimizado seja diferenciável. O requisito é descartado devido ao fato de que o DE não usa o gradiente de função para otimizá-lo. Portanto, o DE também pode ser usado para otimizar

funções contínuas, ruidosas ou até mesmo variantes no tempo. A Figura 4 demonstra como o sistema fica com a combinação dos métodos.

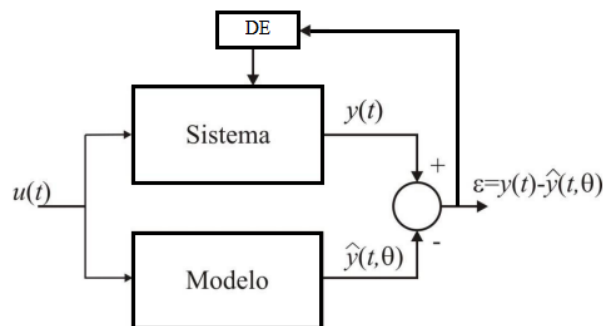


Figura 4. Sistema com MQR+DE.

Pode-se dizer que a otimização da matriz  $P$  do MQR pelo DE é um subsistema onde a busca é feito ao inicializar a matriz  $P$ . O Algoritmo 2 apresenta o pseudo-código do DE para se fazer a ponderação da matriz de covariância  $P$ .

**Algoritmo 2 MQR+DE**

```

Função x = DE(NP, CR, F, range, f)
x ← Aleatorio(range, NP)
fit_x ← 1mmf(x)
while até o critério de parada do
    for i=1 até NP do
        v_i, G + 1 ← mutao (x_i, G, F)
        u_j, G + 1 ← crossover(x_i, G, v_i, G + 1, CR)
    end for
    fit_u ← f(u)
    for i=1 até NP do
        if fit_u(i) > fit_x(i) then
            x_i, G + 1 ← u_i, G + 1
        else
            x_i, G + 1 ← x_i, G
        end if
    end for
end while
    
```

O DE utiliza NP vetores de parâmetros D-dimensionais  $x_i, G, i = 1, \dots, NP$ , como população em cada geração  $G$ . A inicialização do conjunto de vetores é gerado de forma aleatória e deve cobrir todo o espaço de busca. Os novos vetores são gerados através da adição da diferença ponderada entre dois vetores de parâmetros a um terceiro indivíduo, essa mutação pode ser denominada como uma mutação.

A mutação dos vetores é então combinada com outros vetores pré-determinados, denominados vetores alvos, afim de gerar vetores de testes. A presente combinação de parâmetros é referida como *crossover* no DE. Caso os vetores de testes forneça um valor de *fitness* maior que o associado ao respectivo vetor alvo, este último dará lugar ao primeiro na próxima geração, essa etapa pode ser chamada de seleção do DE. Para cada vetor alvo  $x_i, G, i = 1, \dots, NP$ , um novo vetor é gerado da relação de (2):

$$v_i, G + 1 = x_{r_1}, G + F \cdot (x_{r_3}, G - x_{r_2}, G) \quad (2)$$

De acordo com (2)  $r_1, r_2, r_3 \in 1, 2, \dots, NP$  são índices distintos entre si e de  $i$ . Já o  $F$  é uma constante real  $\in [0, 2]$  que determina o tamanho do coeficiente a ser dado na direção definida pelo vetor diferença  $x_{r_3}, G - x_{r_2}, G$ .

Seja  $x_i, G$  o vetor alvo sob análise e  $v_i, G+1$  o respectivo vetor mutado obtido por meio da relação de (2). O vetor  $u_i, G+1 = (u_i, G + 1u_{2i}, G + 1 \dots u_{Di}, G + 1)$ , denominado vetor de teste, é obtido de acordo com (3).

$$u_{j,v}, G + 1 = \begin{cases} v_{j,i}, G + 1, & \text{se } r_j \leq CR \text{ ou } j = 1 \\ G, & \text{se } r_j \geq CR \text{ e } j \neq I_i \end{cases} \quad (3)$$

onde  $j = 1, \dots, D, r_j \sim U(0, 1)$ ,  $CR \geq [0, 1]$  é uma constante definida pelo usuário e  $I_i$  é um índice aleatoriamente escolhido  $\geq 1, \dots, D$ , o que garante que  $u_i, G + 1$  receba pelo menos uma componente de  $v_i, G + 1$ . Assim, seja  $v_i, G + 1$  o respectivo vetor mutado obtido por meio de (2).

Após os passos de mutação e *crossover*, onde os vetores totais serviram com vetores alvos, a seleção dos melhores vetores que passarão para a próxima geração é feita usando busca gulosa. Onde  $x_i, G$  é o vetor alvo analisado e  $u_i, G + 1$  o seu vetor de testes, pode-se deduzir algumas condições gulosas como:

1. Se  $f(u_i, G + 1) > f(x_i, G)$ , então  $x_i, G + 1 = u_i, G + 1$ .
2. Caso contrário,  $x_i, G + 1 = x_i, G$ .

### 3.3 Filtro de Kalman

Também foi utilizado o Filtro de Kalman para se fazer um análise comparativa com o método proposto. O Filtro de Kalman é um conjunto de equações matemáticas que constitui um processo de estimativa recursiva eficiente, uma vez que o erro quadrático é minimizado. Ao observar a variável chamada "variável de observação", e outra variável não observável, chamada de "variável de estado" pode ser estimada de maneira eficiente. Estados passados, estado atual e até mesmo estados futuros previstos podem ser estimados (Aguirre, 2007).

O algoritmo é recursivo, ou seja, ele pode ser executado em tempo real, usando apenas as medições de entrada atuais e o estado calculado anteriormente e sua matriz de incerteza; nenhuma informação passada adicional é necessária. Usando um filtro de Kalman não assume que os erros são gaussianos (Aguirre, 2007). No entanto, o filtro produz a estimativa de probabilidade condicional exata no caso especial de que todos os erros são gaussianos Harvey (1990). O modelo para o filtro de Kalman assume que o estado real no tempo  $k$  é obtido através do estado no tempo  $(k-1)$  de acordo com (4).

$$x_k = F_k x_{k-1} + B_k u_k + w_k \quad (4)$$

onde:  $F_k$  : é o modelo de transição de estado, aplicado na previsão do estado.

$B_k$  : É o modelo de entradas de controle, aplicado ao vetor de controle  $x_{ji}$  de entradas.

$w_k$  : É o ruído do processo, assumido para ser amostrado

a partir de uma distribuição de média zero multivariada normal e covariância.

No momento  $k$ , uma observação (ou medição)  $z_k$  do estado real  $x_k$  é feita de acordo com (5).

$$z_k = H_k x_k + v_k \quad (5)$$

onde  $H_k$  é o modelo de observação, que mapeia o espaço de estado real no espaço de estados observado, e  $V_k$  é o ruído de observação, assumido como sendo um ruído branco gaussiano de média zero e covariância  $R_k$ . O Algoritmo 3 apresenta os passos do Filtro de Kalman.

#### Algoritmo 3 Filtro de Kalman

---

Predição  
 Início do KF  
 Início do KF  
 $x_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$   
 $P_{k|k-1} = F_k P_{k-1|k-1} + F_k^T + Q_k$   
 Atualização  
 $\hat{Y}_k = z_k - H_k \hat{x}_{k|k-1}$   
 $S_k = H_k P_{k|k-1} H_k^T + R_k$   
 $K_k = P_{k|k-1} H_k^T S_k^{-1}$   
 $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{Y}_k$   
 $P_{k|k} = (I - K_k H_k) P_{k|k-1}$

---

sendo:

Y: Resíduo da medição;

S: Resíduo da covariância;

K: Ganho Kalman;

P: matriz de covariância do erro.

## 4. RESULTADOS E DISCUSSÕES

A fim de fazer uma análise bem apurada, a abordagem proposta no trabalho será comparada com outros *benchmarks* clássicos comumente usados em identificação de sistemas, como: Mínimos Quadrados Recursivo e Filtro de Kalman. Ambos serão avaliados o custo computacional e o RMSE. É importante comparar com métodos clássicos, pois a validação dos mesmos vai comprovar a eficácia da pesquisa.

Como foi mencionado anteriormente, o objetivo do DE em relação ao MQR é encontrar uma matriz P que satisfaça um RMSE bem próximo de "0". Assim os mínimos quadrados e o DE foram projetados com as seguintes configurações:

#### Mínimos quadrados Recursivo (MQR):

P = 4x4, matriz de covariância a ser buscada pelo DE;

$\theta = [0; 0; 0; 0]$  é o estado inicial;

#### Evolução diferencial (DE):

NP=50 Tamanho da população ;

F=0.5 fator de mutação;

C=0.9 Passo do crossover;

$I_{max}$  =200 tempo máximo de iterações;

$X_{max}$  = 50 intervalo máximo de busca

$X_{min}$  =0 intervalo mínimo



A Figura 5 apresenta a curva de convergência do DE aplicado ao MQR. Por meio de (6) é apresentada a função de transferência gerada com o MQR+DE.

$$TF = \frac{2.712^{05} - 2.734^{z-1}}{1 - 1.004^{z-2}} \quad (6)$$

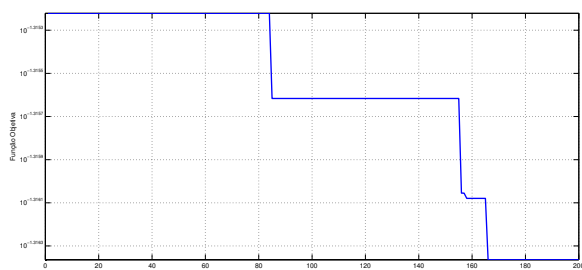


Figura 5. Convergência do erro do MQR

As Figuras 6, 7 e 8 apresentam as identificações dos testes dos Mínimos quadrados Recursivo, Filtro de Kalman (FK), e MQR+DE.

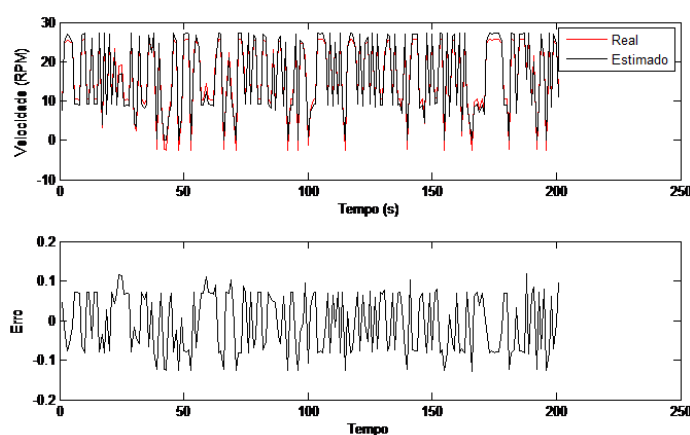


Figura 6. Identificação com MQR.

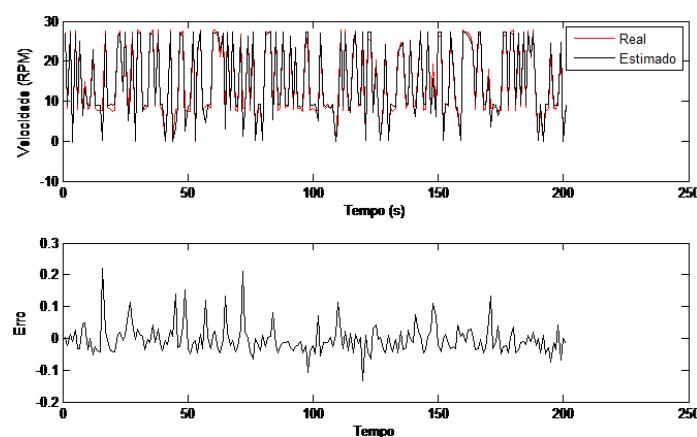


Figura 7. Identificação com FK.

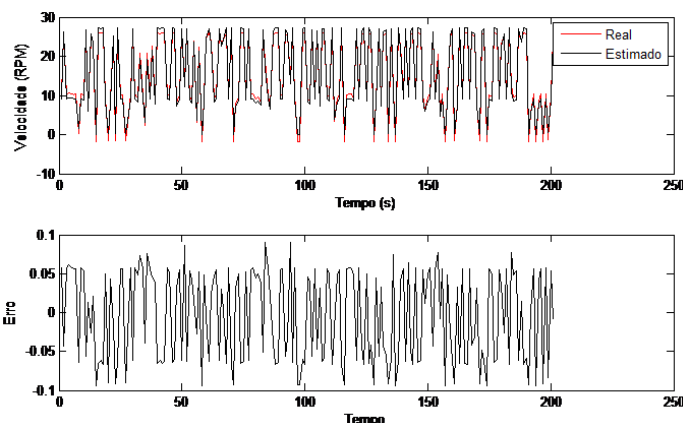


Figura 8. Identificação com MQR+DE.

Em vista das identificações apresentadas nas figuras 6 a 8 pode-se perceber a que obteve o desempenho melhor foi a MQR+DE, isso pode-se validar pelo erro. Também será mostrado uma tabela que será analisado o RMSE e o custo computacional de cada método conforme a Tabela 1.

Tabela 1. Comparação dos métodos

Método	RMSE	Custo computacional (s)
MQR	0.02322612	
FK	0.0220975	3
MQR + DE	0.0219373	3

Com base na Tabela 1 o MQR+DE conseguiu melhor resultado do que os métodos clássicos, assim validando o modelo com o RMSE igual **0.0219373**. Isso se deve a busca global dos parâmetros da matriz P.

## 5. CONCLUSÃO

O trabalho apresentou métodos de identificação de parâmetros de uma das juntas de manipulador robótico. O método proposto ao ser comparado com os clássicos conseguiu um melhor desempenho, o método proposto foi um mínimos quadrados combinado com o evolução diferencial para ponderar a matriz de covariância. Os métodos clássicos que foram usados: mínimos quadrados recursivo e um Filtro de Kalman. O método proposto conseguiu um melhor desempenho ao ser avaliado pelo RMSE. Como proposta de trabalho futuro pretende-se fazer um projeto de controle com o modelo gerado para ter uma validação melhor aceita.

## AGRADECIMENTOS

Os autores agradecem a Capes pela bolsa de auxílio.

## REFERÊNCIAS

- Aguirre, L.A. (2007). Introdução à identificação de sistemas. Editora UFMG, Belo Horizonte, 3.
- Alenany, A. and Shang, H. (2013). Recursive subspace identification with prior information using the constrained least squares approach. *Computers & Chemical Engineering*, 54, 174–180.

- Coelho, Antonio A. R. dos Santos Coelho, L. (2004). *Identificação de sistemas dinâmicos lineares*.
- Harvey, A.C. (1990). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press. doi:10.1017/CBO9781107049994.
- Haykin, S. (2002). *Adaptive filter theory*. Prentice Hall, Upper Saddle River, NJ, 4th edition.
- Pinto, M.F., Mendonça, T.R., Olivi, L.R., Costa, E.B., and Marcato, A.L. (2014). Modified approach using variable charges to solve inherent limitations of potential fields method. In *2014 11th IEEE/IAS International Conference on Industry Applications*, 1–6. IEEE.
- Rebouças, L.R., Cláudio Filho, G., Júnior, A.B., and Laurinda, L.d. (2017). Predictive control applied in 3-phase squirrel cage induction motor for zero speed.
- Storn, R. and Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341–359.