

# SISTEMA DE NAVEGAÇÃO VISUAL BASEADO EM MPC E FILTROS PREDITORES APLICADO A UM ROBÔ OMNIDIRECIONAL

RAMON O. FERNANDES\*, ANDRÉ G. S. CONCEIÇÃO\*, JESSIVALDO SANTOS\*, TITO L. M. SANTOS\*

\* *Universidade Federal da Bahia*  
*Departamento de Engenharia Elétrica*  
*Salvador, Bahia, Brasil*

Emails: ramon.elt.ba@gmail.com, andre.gustavo@ufba.br, jessivaldojr@hotmail.com, tlsantos@ufba.br

**Abstract**— This paper deals with the application of a synthesis methodology based on Model Predictive Control (MPC), associated to Smith Predictor (FSP) and a Extended Kalman Filter (EKF), applied to a three-wheeled omnidirectional mobile robot for trajectory tracking. The approach makes use of an objective function for finite horizon and is based on Linear Matrix Inequality (LMI) framework. The actuator constraints can also be taken in account by this controller. The Filtered Smith predictor approach is used in order to compensate time delay and to attenuate high frequency measurement noise. The EKF is used to fuse data acquired by odometric sensors with those obtained by a visual perception system based on fiducial landmarks that estimates robot's pose. A practical study is presented in order to evaluate the proposed strategy with respect to the advantages associated to the reduction of path following errors related to the pure odometric pose estimation.

**Keywords**— Smith Predictor, Model Predictive Control (MPC), Omnidirectional Robot, Trajectory Control, Extended Kalman Filter (EKF), AprilTags, Visual Servoing.

**Resumo**— Este trabalho apresenta uma aplicação da estratégia de Controle Preditivo baseado em Modelo (MPC) juntamente com o Preditore de Smith (FSP) e o Filtro de Kalman Estendido (EKF) no controle de trajetória de um robô móvel omnidirecional de três rodas. O MPC com horizonte de previsão finito é desenvolvido no contexto das desigualdades matriciais lineares (LMIs), e pode tratar de restrições associadas ao sinal de controle e sua variação. O Preditore de Smith filtrado é utilizado no tratamento do tempo de atraso do sistema e na atenuação de ruídos de alta frequência. O EKF é usado para fundir os dados provenientes dos sensores odométricos do robô com a informação obtida a partir de um sistema de percepção visual baseada em marcos fiduciais que estima a pose do robô. São realizados testes práticos para avaliar as vantagens da técnica proposta no que diz respeito à redução dos desvios de seguimento associados à estimação de pose do robô baseada puramente em odometria.

**Palavras-chave**— Preditore de Smith, Controle Preditivo Baseado em Modelo (MPC), Robô Omnidirecional, Controle de Trajetória, Filtro de Kalman Estendido(EKF), AprilTags, Controle Servo-visual.

## 1 Introdução

A robótica atualmente é uma área bastante difundida e de grande interesse para comunidade em geral. Em especial, é notável o crescimento do uso de robôs móveis em vários contextos: fábricas, carros autônomos, agricultura de precisão, entre outros. Assim, temas relacionados com a mobilidade dos robôs têm despertado grande interesse da comunidade científica.

Para que possa ser possível uma boa navegação e execução satisfatória de qualquer tarefa proposta, é necessário que haja um bom sistema de localização associado ao robô. Este, por sua vez, demanda uma boa percepção do entorno através do uso de diversos sensores. Nesse contexto, o sensor visual que tem se mostrado particularmente eficiente. Abordagens como (Burri, 2015), que se valem da técnicas localização e mapeamento simultâneo (SLAM), aplicando fusão sensorial para unir leituras de scanners laser, Sistema de Posicionamento Global (GPS), câmeras de vídeo estéreo para localizar o veículo em ambientes internos. Existem também abordagens para locais externos, como visto em (Santana, 2008) onde é proposto um EKF que utiliza informações de um sensor GPS

em conjunto com as de sensores inerciais para obter estimativas de posição que servem para o controle de um quadrotor. Observa-se que os sistemas de estimação de posição em ambientes externos podem se beneficiar significativamente do uso de dados visuais para evitar o desvio da informação, uma vez que haja oclusões ou falhas de comunicações com os demais sensores. Vale observar que tais falhas no sensor, ou mesmo uma alta latência são usuais em sensores como por exemplo em GPS, daí a necessidade da filtragem.

Outra tarefa associada à navegação de robôs é o controle de trajetória. Atualmente existem várias abordagens para realizá-la, boa parte delas resultam da verificação de diversos *checkpoints* fixados sobre um caminho caminho. O problema então é formulado em termos da distância entre a posição do robô e cada um dos *checkpoints*. Para regular essa distância a zero são propostos várias estratégias. Em (Calzolari et al., 2017) é apresentada uma breve explicação e comparação de diversas técnicas adotadas atualmente, como: LQR, controle por modos deslizantes, linearização por realimentação de estados, entre outras. Em (Abid et al., 2017) é projetado um controlador PI para um robô diferencial com duas rodas.

Em (Montesdeoca et al., 2017) o controle de trajetória é tratado para uma configuração diferencial com duas rodas, se valendo de um modelo cinemático estendido que permite tratar as restrições não-holonômicas inerentes. Por outro lado, em (Ma et al., 2017) observa-se um controle de trajetória baseado em redes neurais junto com um sistema de SLAM para um robô omnidirecional. Em (Abdelaal and Hahn, 2016) é abordado um sistema de controle preditivo baseado em modelo não-linear (NMPC) com desvio de obstáculos implementado diretamente no controle de um veículo marinho. Também em (Guerreiro et al., 2014) é apresentada uma estratégia NMPC, mas dessa vez tratando da saturação dos atuadores. Já em (Hwang and Huang, 2017) é apresentada uma estratégia baseada no controle por modos deslizantes junto com um desvio de obstáculos baseados no sensor RGB-D para robôs com configuração similar a um carro.

O trabalho apresentado propõe uma extensão da abordagem feita em (Santos et al., 2017), para isso o sistema de estimação de posição baseado no EKF é inserido diretamente no controlador existente. Este artigo se trata de uma melhoria na estratégia para controle de trajetória baseado em MPC, e tem como principal contribuição a redução dos desvios de seguimento associados à estimação de pose do robô baseada puramente em odometria. Melhoria que se deve à introdução de uma estratégia de estimação para a postura do robô baseada em informação visual adquirida por um sistema de marcos fiduciais e de um processamento estatístico dos dados sensoriais. Os resultados são comprovados a partir de experimentos práticos com uma plataforma móvel experimental omnidirecional com três rodas.

O restante desse artigo está organizado da seguinte forma: a segunda seção descreve brevemente a modelagem do robô adotada; a terceira trata do Filtro de Kalman; já na quarta seção o sistema proposto é descrito; a quinta apresenta os resultados dos testes bem como uma breve análise; e, por fim, são feitas as conclusões.

## 2 Modelo do Robô

Considere o sistema apresentado na Figura 1, uma representação do robô com a definição dos sistemas de coordenadas adotados. Note que o sistema de coordenadas denotado por  $x_r$  e  $y_r$  está fixado no centro de massa do robô, e assim as velocidades  $v$  e  $v_n$  representam a taxa de deslocamento linear nas direções de  $x_r$  e  $y_r$ , respectivamente, e  $\omega$  é a velocidade angular de giro do robô sobre o próprio eixo.

O modelo do robô, apresentado na Equação 1, é obtido a partir da análise da Figura 1. Para isso, são feitas as seguintes considerações: todas as rodas tem o mesmo raio  $r$ ; o raio do robô é dado

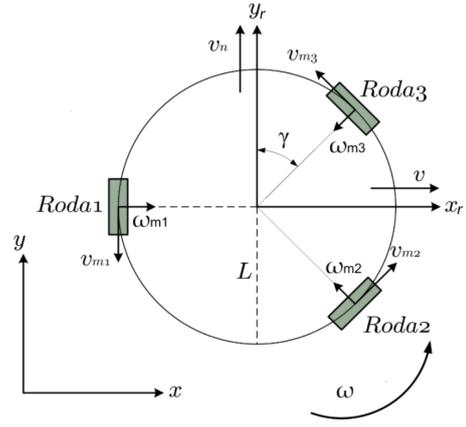


Figura 1: Representação do robô omnidirecional.

por  $L$ ; e  $\gamma = 30^\circ$ . Note ainda que os termos  $vm1$ ,  $vm2$ , e  $vm3$  representam as velocidades lineares de cada roda.

$$\begin{cases} vm1 = -v_n + \omega L \\ vm2 = v \cos(\gamma) + v_n \sin(\gamma) + \omega L \\ vm3 = -v \cos(\gamma) + v_n \sin(\gamma) + \omega L \end{cases} \quad (1)$$

A Equação 2 representa o modelo cinemático do robô, e é obtida a partir da manipulação da Equação 1, como pode ser verificado em (Thrun, 2005). As grandezas são dadas por: as velocidades do robô em relação ao sistema de coordenadas global,  $[\dot{x} \ \dot{y} \ \dot{\theta}]^T$ ; orientação do robô em relação ao sistema de coordenadas global,  $\theta$ ; a velocidade angular da cada um dos motores,  $\omega_{mi}$ ; a relação de redução das engrenagens das rodas,  $n$ .

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & L \\ \cos\gamma & \sin\gamma & L \\ -\cos\gamma & \sin\gamma & L \end{bmatrix}^{-1} \begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \end{bmatrix} \quad (2)$$

## 3 Filtro de Kalman

O Filtro de Kalman é uma técnica recursiva que trabalha em um ciclo de predição e atualização. Ele se propõe a estimar de forma ótima os estados de um sistema, assumindo que os dados obtidos estejam corrompidos por um ruído gaussiano branco. Para isso, é adotada uma descrição probabilística do processo, onde o vetor de estados passa a ser representado por meio de duas variáveis: o vetor da média dos estados,  $\mu_k$ , e a matriz de covariância dos estados,  $\Sigma_k$ .

Uma versão desse filtro é definida na Equação 3. Nela o sistema é descrito pela função  $g(\mu_{k-1}, u_k)$ , que tem como parâmetros o vetor da média dos estados,  $\mu_k$ , e o vetor de sinais de controle,  $u_k$ . São definidos: a matriz jacobiana do sistema em relação ao vetor da média dos estados,  $G_k$ ; a matriz de covariância das medições,  $Q_k$ ; a matriz de covariância dos ruídos do processo,  $R_k$ ; o ganho ótimo computado pelo filtro,  $K_k$ ; o modelo do sensor,  $h(\mu_{k-1}, m, j)$ ; a matriz jacobiana desse modelo,  $H_k$ ; e  $z_k$  que é a informação sensorial. O primeiro par de equações representa a predição dos estados, enquanto as demais representam os passos de correção e atualização.

$$\begin{cases} \overline{\mu}_k = g(\mu_{k-1}, u_k) \\ \overline{\Sigma}_k = G_k \Sigma_k G_k^T + R_k \\ \left\{ \begin{array}{l} K_k = \overline{\Sigma}_k H_k^T (H_k \overline{\Sigma}_k H_k^T + Q_k)^{-1} \\ \mu_k = \overline{\mu}_k + K_k (z_k - h(\mu_{k-1}, m, j)) \\ \Sigma_k = (I - K_k H_k) \overline{\Sigma}_k \end{array} \right. \end{cases} \quad (3)$$

Primeiramente, é definida a matriz jacobiana do modelo em relação à média dos estados,  $G_k$ . Ela é dada pela expressão abaixo, sendo que os elementos  $g_{13}$  e  $g_{23}$  são obtidos pela Equação 5. Observe que  $\theta_k$  representa a orientação do robô em relação ao sistema de coordenadas global no instante  $k$ .

$$G_k = \frac{\partial g(\mu_{k-1}, u_k)}{\partial \mu_{k-1}} = \begin{bmatrix} 1 & 0 & g_{13} \\ 0 & 1 & g_{23} \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$\begin{cases} g_{13} = \frac{r}{3} (2 \cos(\theta_{k-1}) \omega_{m1} + (-\sqrt{3} \text{sen}(\theta_{k-1}) \\ \quad - \cos(\theta_{k-1}) \omega_{m2} + (\sqrt{3} \text{sen}(\theta_{k-1}) \\ \quad - \cos(\theta_{k-1}) \omega_{m3}) \\ g_{23} = \frac{r}{3} (2 \text{sen}(\theta_{k-1}) \omega_{m1} + (\sqrt{3} \cos(\theta_{k-1}) \\ \quad - \text{sen}(\theta_{k-1}) \omega_{m2} + (-\sqrt{3} \cos(\theta_{k-1}) \\ \quad - \text{sen}(\theta_{k-1}) \omega_{m3}) \end{cases} \quad (5)$$

O cálculo da matriz  $R_k$  é feito a partir da equação apresentada a seguir. É importante notar que, como mostrado em (Santana, 2008), essa expressão é derivada do modelo odométrico proposto por (Thrun, 2005), e incorpora os *encoders* como parte do sistema e não como sensores. Note também que a matriz  $M_k$  representa a covariância do ruído dos *encoders*, e é calculada a partir da Equação 7.

$$R_k = V_k M_k V_k^T \quad (6)$$

$$M = \begin{bmatrix} (\alpha_1 |\Delta \omega_{m1}|)^2 & 0 & 0 \\ 0 & (\alpha_2 |\Delta \omega_{m2}|)^2 & 0 \\ 0 & 0 & (\alpha_3 |\Delta \omega_{m3}|)^2 \end{bmatrix} \quad (7)$$

É importante salientar que: na Equação 7 o erro associado a cada *encoders* é considerado proporcional ao módulo do deslocamento angular,  $\omega_{mi}$ ; e que as constantes de proporcionalidade,  $\alpha_i$ , foram estabelecidas a partir da experimentação.

Por fim, a matriz jacobiana que mapeia os ruídos dos *encoders* para o espaço de estados,  $V_k$ , é definida a seguir, junto com os seus elementos que são calculados de acordo com a Equação 9.

$$V_k = \frac{\partial g(\mu_{k-1}, u_k)}{\partial u_k} = \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ \frac{r}{3l} & \frac{r}{3l} & \frac{r}{3l} \end{bmatrix} \quad (8)$$

$$\begin{cases} v_{11} = 2 \frac{r}{3} \text{sen}(\theta_{k-1}) \\ v_{12} = \frac{r}{3} (\sqrt{3} \cos(\theta_{k-1}) - \text{sen}(\theta_{k-1})) \\ v_{13} = \frac{r}{3} (-\sqrt{3} \cos(\theta_{k-1}) - \text{sen}(\theta_{k-1})) \\ v_{21} = -2 \frac{r}{3} \cos(\theta_{k-1}) \\ v_{22} = \frac{r}{3} (\sqrt{3} \text{sen}(\theta_{k-1}) + \cos(\theta_{k-1})) \\ v_{23} = \frac{r}{3} (-\sqrt{3} \text{sen}(\theta_{k-1}) + \cos(\theta_{k-1})) \end{cases} \quad (9)$$

### 3.1 Modelo do Sensor Visual

Para finalizar, resta definir as equações que modelam o sensor visual. Para isso, não serão utilizadas diretamente as estimações de posição geradas pelo sistema de marcos artificiais adotado, AprilTag, descrito em (Olson, 2011). Ao invés, a distância entre o alvo e a câmera,  $q$ , e a orientação entre o alvo e a câmera,  $\phi$ , serão consideradas como informações sensoriais. Essa abordagem foi adotada com base em (Neunert et al., 2015), e se deve à dificuldade de definir um modelo adequado para o ruído das estimações visuais.

Assim, utilizando o exposto em (Thrun, 2005), o sensor visual é modelada como descrito na Equação 10. Também é importante mencionar que a função  $\text{atan2}()$  utilizada na formulação representa uma variação da função trigonométrica  $\text{atan}()$ , onde o resultado obtido é um ângulo restrito ao intervalo  $(-\pi, \pi]$ .

$$\begin{bmatrix} q \\ \phi \end{bmatrix} = h(\mu_{k-1}, m, j) = \begin{bmatrix} \sqrt{(m_{x,j} - \bar{x})^2 + (m_{y,j} - \bar{y})^2} \\ \text{atan2}((m_{y,j} - \bar{y}), (m_{x,j} - \bar{x})) - \bar{\theta} \end{bmatrix} \quad (10)$$

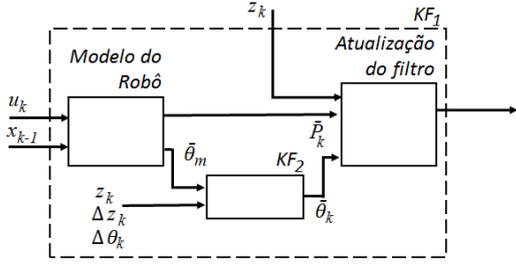


Figura 2: Representação do sistema de filtros proposto, onde  $\bar{P}_k$  denota a posição do robô, e as demais variáveis são como definidas no texto.

Dessa forma, pode-se definir a matriz  $H_k$ , que representa o jacobiano do modelo do sensor e é definida a seguir, onde:  $\bar{x}$ ,  $\bar{y}$  e  $\bar{\theta}$  são a estimativa dos estados feitas na etapa de predição; as grandezas  $m_{x,j}$  e  $m_{y,j}$  são as posições do alvo que são predefinidas, e  $j$  representa a identificação, *id*, de cada alvo.

$$H_k = \frac{\partial(h(\mu_{k-1}, m, j))}{\partial \mu_{k-1}} = \begin{bmatrix} -\frac{(m_{x,j} - \bar{x})}{q} & -\frac{(m_{y,j} - \bar{y})}{q} & 0 \\ \frac{(m_{y,j} - \bar{y})}{q} & -\frac{(m_{x,j} - \bar{x})}{q} & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad (11)$$

### 3.2 Filtro da Posição Angular

Durante o desenvolvimento do sistema percebeu-se que a estimativa da posição angular do robô tem um papel crucial para o bom desempenho do controle. Porém, a estimativa feita pelo sistema de visão para essa grandeza é bastante oscilante, e afeta a qualidade de todo o sistema.

Para resolver este problema foi proposto utilizar um segundo EKF, responsável por filtra os valores da estimativa de orientação a serem passados para o primeiro filtro, que roda numa camada de controle superior. Esse filtro considera que a dinâmica do sensor não varia subitamente, uma vez que as velocidades das rodas do robô são limitadas. Assim, são consideradas como informações sensoriais: os valores da estimativa a partir dos sensores odométricos,  $\theta_{odom_k}$ ; as variações dessa estimativa,  $\Delta\theta_{odom_k}$ ; o valor da estimativa do sensor visual,  $\theta_{vis_k}$ ; e a sua variação,  $\Delta\theta_{vis_k}$ . O segundo EKF tem como modelo do sistema a equação usado para estimar a orientação do robô durante a fase de predição do primeiro filtro, Equação 2, representado aqui como  $\theta_{pred_k}$ . Uma representação da disposição dos filtros é vista na Figura 2

O segundo filtro pode ser descrito pela Equação 3, porém nesse caso a formulação é dada pelo

seguinte conjunto de expressões, onde  $Ts$  representa o período de amostragem do sistema completo, e  $q$ ,  $\sigma_\theta$ , e  $\sigma_z$  são tratados como parâmetros de ajuste:

$$H_k = \begin{bmatrix} 1 & Ts \\ Ts & 1 \end{bmatrix}$$

$$G_k = 1$$

$$R_k = \begin{bmatrix} (\sigma_\theta)^2 & 0 & 0 & 0 \\ 0 & (\sigma_\theta)^2 & 0 & 0 \\ 0 & 0 & (\sigma_z)^2 & 0 \\ 0 & 0 & 0 & (\sigma_\theta)^2 \end{bmatrix}$$

$$Q_k = q$$

$$h(\mu_{k-1}) = \mu_{k-1} * H_k \quad (12)$$

## 4 Sistema Proposto

Um esquema da abordagem de controle pode ser visto na Figura 3, onde a *loop* externo é responsável pelo seguimento de caminho. Ele usa uma estimativa da postura atual do robô, fornecida pelo sistema de estimativa baseado em visão, para calcular o erro de seguimento,  $e(t)$ . O erro então serve para obter as referências de velocidade a serem passadas ao segundo controlador, o controlador de velocidade. Este controlador é o mesmo proposto em (Santos et al., 2017), e opera segundo uma lei de controle MPC. Também foi incluído o Preditor de Smith para lidar com as implicações do atraso variável da comunicação entre robô e o computador base.

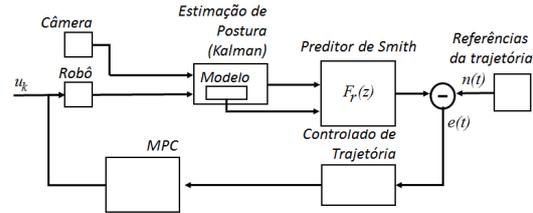


Figura 3: Representação do sistema de controle proposto.

## 5 Resultados Experimentais

Para avaliar os benefícios do sistema proposto e fazer uma comparação com a abordagem baseada puramente em informação odométrica foram realizados experimentos práticos. Neles o robô deveria dar três voltas consecutivas seguindo um caminho circular com meio metro de raio e mantendo a orientação. Nesse teste os marcos artificiais, AprilTags, foram dispostos sobre a plataforma robótica,



Figura 4: Montagem experimental.

e uma câmera foi ajustada de forma que eles esteja sempre visível durante o trajeto.

Todos os controladores foram desenvolvidos no *framework* ROS. A tarefa de controle é executada em um computador rodando em LINUX OS. Essa tarefa é periódica e tem como tempo de amostragem 60ms. A comunicação entre o robô e os controladores é feita por meio de uma rede Zigbee (Baronti et al., 2007). O sensor visual utilizado foi o Kinect, sendo que o sistema de estimação proposto funciona a 30ms. A velocidade de navegação adotada foi de 0.25m/s e os horizontes de predição do MPC foram fixados em 6.

Para que seja possível ter uma base de comparação foram realizados dois tipos de testes, no primeiro a informação visual foi repassada ao controlador por meio do filtro proposto. No segundo o sistema de estimação baseada em visão foi substituído por um modelo puramente odométrico. Em ambos testes as estimações obtidas pelo AprilTags são salvas e serão usadas como referência para avaliar a qualidade do controle. Note que, com essa abordagem pequenos desvios na posição inicial do robô, devidos ao posicionamento impreciso dele em relação à câmera, influenciam no desempenho final.

Ao observar as Figuras 6 e 7 é possível notar que o robô realiza uma translação indesejada no trajeto. Esse fato se deve ao sistema de estimação que se baseia apenas nos dados odométrico e, portanto, acumula erros durante a execução. Observa-se que a medida em que o tempo progride, o erro de posição aumenta. É enfatizado que a forma circular consistente e com poucos desvios, mostrada em vermelho, representa a percepção do robô. Assim, o fato da odometria estar seguindo o círculo em preto evidencia a ineficiência da abordagem, uma vez que do ponto de vista do controlador não há um erro a ser corrigido. Por outro lado, a trajetória no mundo real, representada em azul e oriunda do sistema AprilTags, se mostra inconsistente. Isso evidencia que a estratégia de controle é adequada, porém, a percepção e estimação de posição não são satisfatórias, e fornecem pouca informação para a boa execução da tarefa.

Assim, é introduzido o sistema de estimação

baseado em visão. Como visto na Figura 8 o robô realiza descreve uma trajetória mais coerente com o sistema de visão. Observa-se que nesse caso a medida em que o tempo progride, o erro de posição se reduz. Comportamento que pode ser explicado por dois fatores: o primeiro é a convergência dos filtros de Kalman, pois é sabido que essa ferramenta necessita de algumas iterações para tornar-se efetiva; o segundo é uma oclusão momentânea, vista na Figura 5. Essa privação da informação durante os ciclos iniciais, onde a convergência do filtro ainda não foi alcançada, acarreta um maior tempo de transitório para o filtro, além da própria degradação na estimação. Também é enfatizado que a forma circular em vermelho, que anteriormente era consistente, agora apresenta um deslocamento, denotando a ação benéfica da inclusão do sistema baseado em visão.

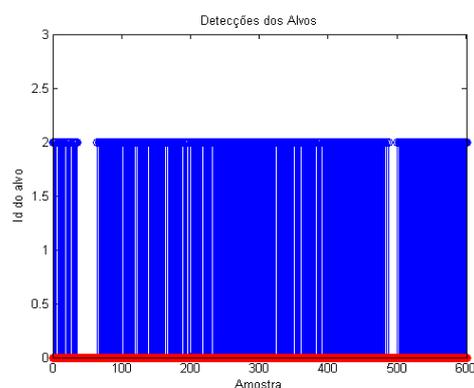


Figura 5: Detecção do marco artificial no tempo.

Também, a partir das Figuras 6, 8 e 14, observa-se que a trajetória obtida pela abordagem proposta é consistente, e a informação visual aponta um desvio menor do que o apresentado anteriormente. Além disso, nota-se que a orientação do robô se beneficiou da estratégia adotada, já que é possível observar variações menos abruptas no sinal obtido, mesmo que em ambos experimentos elas sejam maiores, e mais impulsivas do que o desejado. Também deve-se ter em conta que a estimação de angulação fornecida por sistemas visuais, bem como pelo AprilTags, é inerentemente oscilante. Dessa forma parte do comportamento obtido pode ser entendido pela escolha do sensor.

Ao analisar as Figuras 10 e 11 não é possível visualizar grandes impactos da abordagem adotada sobre o sistema de controle. É importante ressaltar que a velocidade angular, nos dois casos, constitui um subsistema de dinâmica mais rápida na tarefa proposta, e, por isso, tem um papel maior na qualidade do resultado obtido. Também, a essa dinâmica se atribui o pior desempenho em termos do seguimento de referência do sinal. Além disso, observando as Figuras 12 e 13 é possível concluir que ambas propostas apresentam esforços de controle semelhante, sempre respeitando as limi-

tações impostas pelo MPC. Daí, é possível observar que a introdução do sistema visual tem pouco impacto no esforço de controle, e ganhos no seguimento de trajetória.

## 6 Conclusões

Neste artigo é apresentado um estudo prático sobre um sistema de estimação de postura baseado em informação visual, a ser utilizado em conjunto com uma abordagem de controle MPC mais FSP, para o seguimento de caminhos. A partir da comparação dos resultados obtidos durante os testes realizados, ficou clara a importância do sistema proposto na redução do desvio no seguimento de caminho, sem afetar significativamente os esforços de controle. Assim, constitui uma vantagem real para o controle em geral. Também é possível observar que a estimação de orientação, e o tempo de convergência dos filtros de Kalman adotados constituem dois gargalos no desempenho do sistema. Dessa forma, pode ser vantajoso a adição de sensores mais confiáveis para a estimação, como por exemplo um sensor de bússola, e também, a formulação de uma variedade mais rápida e com maior liberdade no modelo de ruído para o filtro de Kalman, como o *Unscented Kalman Filter*.

## Agradecimentos

Os autores agradecem ao CNPq pelo fomento. Este trabalho recebeu financiamento do programa de investigação e inovação Horizonte 2020 da União Europeia, contrato 777096 e da SEPIN / MCTI 4º chamada coordenada BR-UE no CIT.

## Referências

Abdelaal, M. and Hahn, A. (2016). Nmpc-based trajectory tracking and collision avoidance of unmanned surface vessels with rule-based colregs confinement, *2016 IEEE Conference on Systems, Process and Control (ICSPC)* pp. 23–28.

Abid, D. B. H., Allagui, N. Y. and Derbel, N. (2017). Navigation and trajectory tracking of mobile robot based on kinematic pi controller, *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)* pp. 252–256.

Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A. and Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards, *Computer Communications* **30**(7): 1655–1695.

Burri, Michael; Oleynikova, H. A. M. W. S. R. (2015). Real-time visual-inertial mapping,

re-localization and planning onboard mavs in unknown environments, *Intelligent Robots and Systems (IROS)* p. 1872–1878.

- Calzolari, D., Schurmann, B. and Althoff, M. (2017). Comparison of trajectory tracking controllers for autonomous vehicles, *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* pp. 1–8.
- Guerreiro, B. J., Silvestre, C., Cunha, R. and Pascoal, A. (2014). Trajectory tracking nonlinear model predictive control for autonomous surface craft, *IEEE Transactions on Control Systems Technology* **22**(6): 2160–2175.
- Hwang, C. L. and Huang, H. H. (2017). Experimental validation of a car-like automated guided vehicle with trajectory tracking, obstacle avoidance, and target approach, *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society* pp. 2858–2863.
- Ma, J., Liu, H., Wang, S. and Huang, Q. (2017). Adaptive neural network controller of an intelligent commode robot for trajectory tracking, *2017 IEEE International Conference on Cyborg and Bionic Systems (CBS)* pp. 304–309.
- Montesdeoca, J. C., Santos, M. C. P., Monllor, M. and Herrera, D. (2017). Trajectory tracking controller for differential-drive mobile robots, *2017 XVII Workshop on Information Processing and Control (RPIC)*.
- Neunert, M., Bloesch, M. and Buchli, J. (2015). An open source, fiducial based, visual-inertial state estimation system.
- Olson, E. (2011). A robust and flexible visual fiducial system, *IEEE International Conference on Robotics and Automation (ICRA-11)* pp. 3400–3407.
- Santana, A. M.; Sousa, A. A. S. B. R. S. A. P. J. M. A. A. D. (2008). Localization of a mobile robot based in odometry and natural landmarks using extended kalman filter, *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.
- Santos, J., G.S. Conceicao, A. and Santos, T. (2017). Trajectory tracking of omnidirectional mobile robots via predictive control plus a filtered smith predictor, *IFAC-PapersOnLine* **50**: 10250–10255.
- Thrun, S.; Burgard, W. F. D. (2005). *Probabilistic Robotics*, MIT Press.

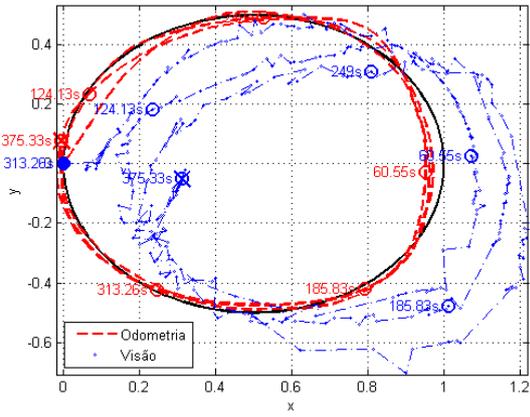


Figura 6: Resultados obtidos sem a estimação baseada em visão. Os pontos iniciais estão representados pelo círculos preenchidos, e os finais por um X. O círculo ideal é representado em preto, o *ground truth* em azul, e a posição estimada em vermelho.

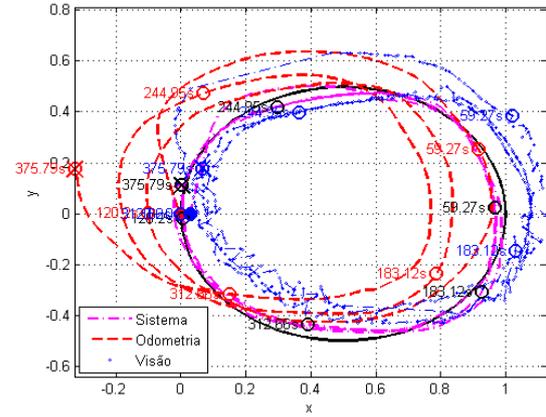


Figura 8: Resultados com a estimação visual. Os pontos iniciais estão representados pelo círculos preenchidos, e os finais por um X. O círculo ideal é representado em preto o *ground truth* em azul, a posição estimada por odometria em vermelho, e a estimação do sistema proposto em magenta.

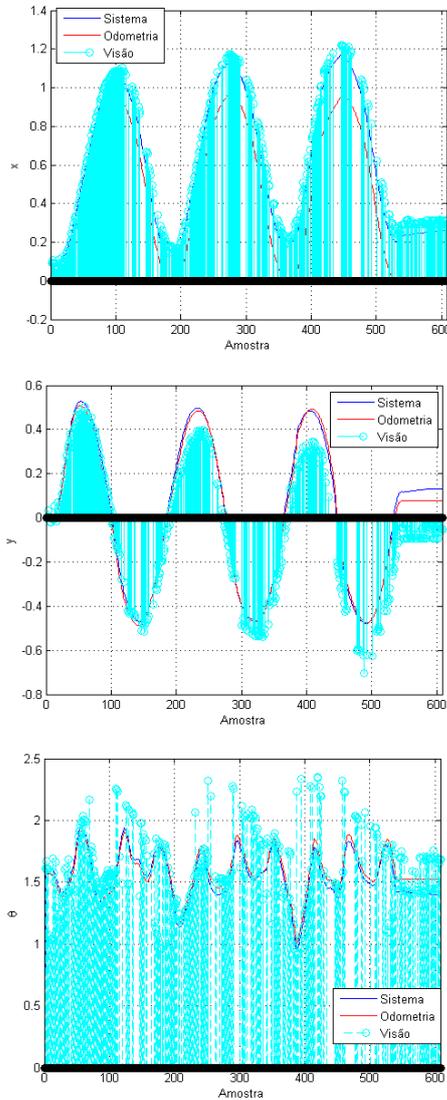


Figura 7: Resultados obtidos sem a estimação baseada em visão.

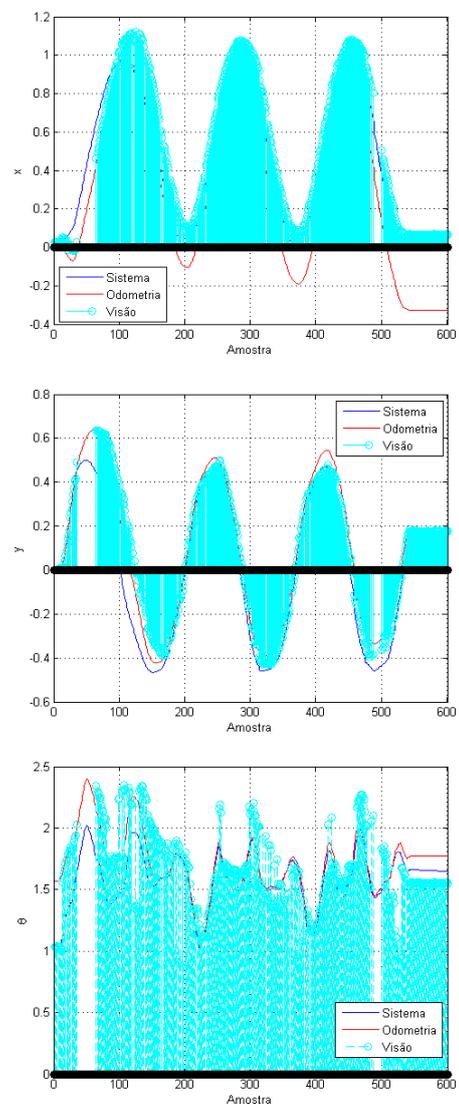


Figura 9: Resultados obtidos com a estimação baseada em visão.

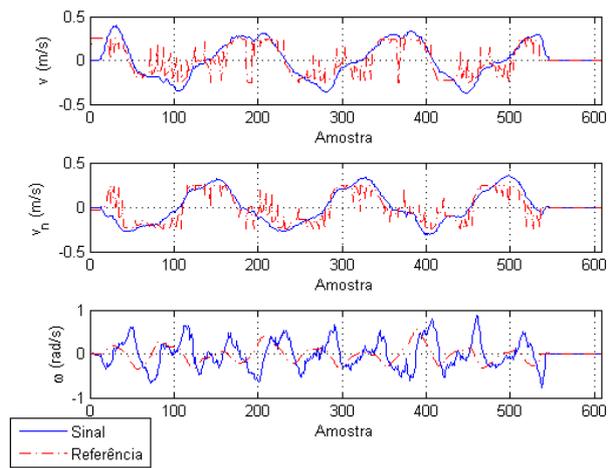


Figura 10: Velocidades do robô durante o primeiro experimento.

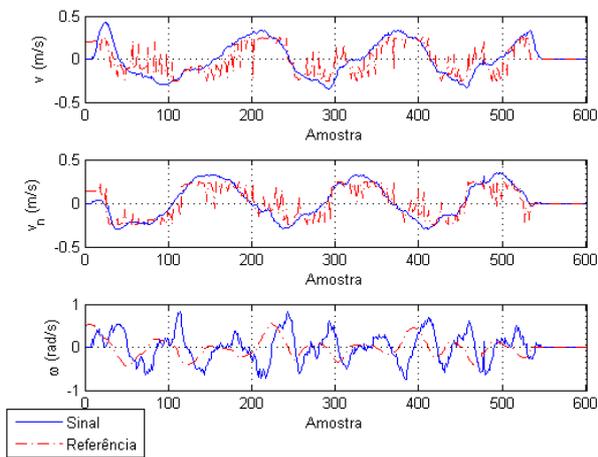


Figura 11: Velocidades do robô durante o segundo experimento.

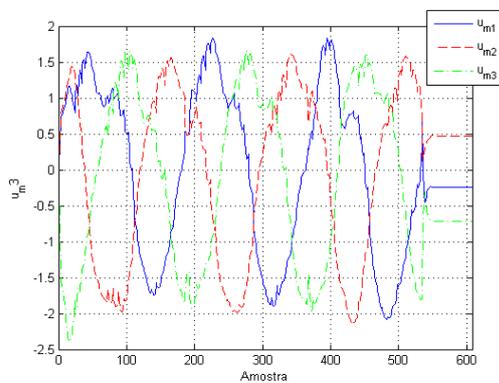


Figura 12: Sinais de controle dos motores do robô durante o primeiro experimento.

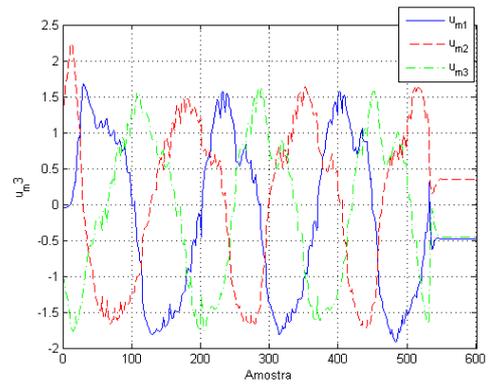


Figura 13: Sinais de controle dos motores do robô durante o segundo experimento.

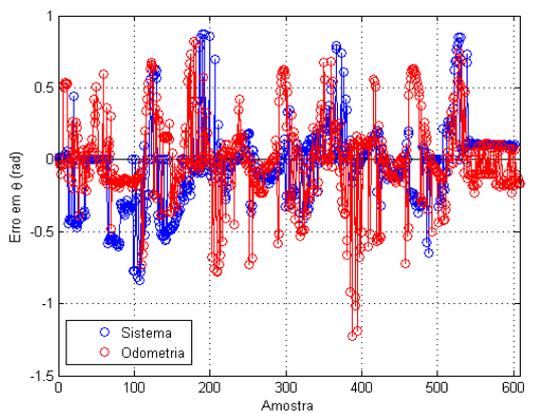
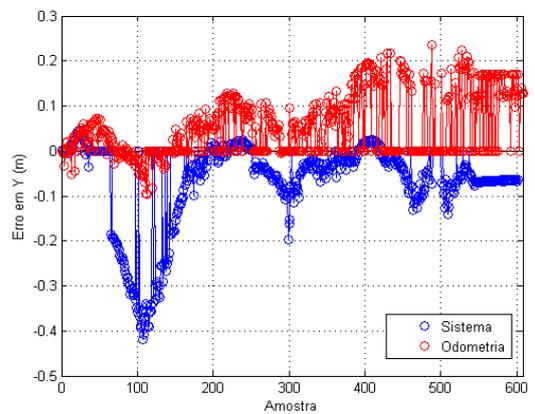
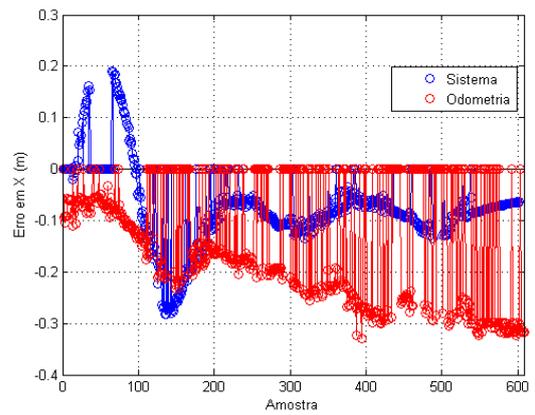


Figura 14: Comparação dos erros do controle de trajetória durante os experimentos.