

FILTRO DE KALMAN ESTENDIDO BASEADO EM QUATÉRNIONS PARA A FUSÃO DE SENSORES INERCIAIS E IMAGEM

RAFAEL S. CASTRO*, PAULO VANCIN*, VICTOR M. S. PEIXOTO*, THIAGO T. CAVALHEIRO*,
AURÉLIO TERGOLINA SALTON*

*PUCRS - Grupo de Automação e Controle de Sistemas
Av. Ipiranga, 6681, 90619-900
Porto Alegre (RS), Brasil

Emails: rafael.castro@pucrs.br, {paulo.vancin, victor.peixoto,
thiago.cavalheiro}@acad.pucrs.br, aurelio.salton@pucrs.br

Abstract— In this paper, a technique based on computer vision to determine position and orientation (pose) of an object is presented. A sensor fusion algorithm based on the Extended Kalman filter is proposed in order to treat the data collected from the visual sensor and the inertial sensor. This fusion produces a precise and robust estimate pose estimate. To facilitate the construction of the Kalman filter, a quaternion representation is used in the framework of the filter. The system is tested on a rig and the results produced show a satisfactory behavior of the project.

Keywords— Estimation, Computer vision, Quaternions.

Resumo— Neste artigo, uma forma de estimação de posição e orientação (pose) de um objeto utilizando visão computacional é apresentada. O algoritmo de fusão se baseia no filtro de Kalman Estendido para tratar os dados provenientes de sensores visuais e inerciais, produzindo, assim, uma estimação precisa e robusta da pose do sistema. Para facilitar a implementação do filtro de Kalman, a representação por quatérnios é utilizada para descrever o sistema. O algoritmo proposto é validado em uma bancada de testes apresentando resultados satisfatórios.

Palavras-chave— Estimação, Visão computacional, Quatérnios.

1 Introdução

A visão computacional possui uma grande gama de aplicações em sistemas dinâmicos, podendo variar de tarefas triviais, como o sensoriamento de braços robóticos (Rausch et al., 2013), até a exploração espacial por veículos autônomos (Goldberg et al., 2002). De fato, a estimação de orientação e localização de veículos autônomos é tema corrente no uso da visão computacional (Kornuta and Zieliński, 2013; Palomeras et al., 2013; Yang et al., 2013). A aplicação de odometria visual para a estimação de pose de quadrópteros é em encontrada em Altuğ et al. (2003), a tarefa de localização e mapeamento simultâneos é realizada em Silveira et al. (2008) e o uso do Filtro de Kalman é debatido em Janabi-Sharifi and Marey (2010). Uma alternativa à necessidade de utilização de múltiplas câmeras ou sensores visuais de alta qualidade é a utilização de sensores complementares na implementação de visão computacional. Esta é a abordagem a ser explorada neste trabalho.

Para a estimação da pose (localização e orientação) de sistemas autônomos geralmente são utilizados sensores inerciais do tipo acelerômetros e giroscópios. Em Rehbinder and Ghosh (2003), Chai et al. (2002), Nützi et al. (2011) e Engel et al. (2012) a fusão de dados provenientes de câmeras e sensores inerciais é realizada para estimar a localização e orientação dos sistemas. O trabalho de Steder et al. (2008) utiliza ainda outros sensores auxiliares como sonares e *lasers*. Já em

Martínez et al. (2011) além de sensores inerciais, sensores como GPS (*Global Positioning System*) e Magnetômetros são utilizados como estimadores de estado enquanto a câmera funciona como um controlador *low-level*.

Neste trabalho será proposto um estimador de estados capaz de realizar a fusão das informações coletadas por um sistema de visão computacional com as informações provenientes de sensores inerciais. Através da utilização do Filtro de Kalman Estendido, do inglês *Extended Kalman Filter* (EKF), baseado em quatérnios, a posição e orientação de uma câmera será obtida de maneira precisa a partir de marcadores (*landmarks*) conhecidos. O uso de Quatérnios para a descrição das equações dinâmicas simplifica o equacionamento do filtro e elimina as funções trigonométricas de seno e cosseno, que possuem alto custo computacional (Xu and Mandic, 2015). Resultados experimentais comparando o deslocamento de um eixo linear medido por um encoder com a estimativa fornecida pelo filtro serão apresentados a fim de validar o desempenho do algoritmo proposto.

O artigo está organizado da seguinte forma: na Seção 2 serão expostos os conhecimentos preliminares para a realização do trabalho e a definição do problema. Na Seção 3 o modelo do sistema será derivado. O algoritmo responsável pela fusão dos sensores está exposto na Seção 4. O experimento realizado para validar o sistema proposto, as ferramentas utilizadas para a coleta de dados visuais e inerciais e os resultados destes testes serão

expostos e analisados na Seção 5. As conclusões derivadas deste trabalho serão discutidas na Seção 6.

2 Preliminares

2.1 Representação por Quatérnios

No que diz respeito à modelagem de sistemas dinâmicos, quatérnios são interessantes por representarem matrizes de rotação em três dimensões. Como descrito em (Kristiansen et al., 2005), uma rotação de θ radianos sobre um vetor unitário arbitrário $\mathbf{k} \in \mathbb{R}^3$ pode ser parametrizada como¹

$$R(\mathbf{q}) = I_3 + 2\eta S(\boldsymbol{\varepsilon}) + 2S(\boldsymbol{\varepsilon})^2, \quad R(\mathbf{q}) \in \mathbb{R}^{3 \times 3} \quad (2)$$

sendo I_3 a matriz identidade de ordem três, $\eta = \cos(\theta/2)$, $\boldsymbol{\varepsilon} = \mathbf{k} \cdot \sin(\theta/2) \in \mathbb{R}^3$, e $\mathbf{q} \in \mathbb{R}^4$ definido como,

$$\mathbf{q} := \begin{bmatrix} \eta \\ \boldsymbol{\varepsilon} \end{bmatrix}. \quad (3)$$

Desta forma o vetor $\mathbf{q}$ passa a representar a orientação de um corpo em relação ao sistema de coordenadas globais. Além de serem computacionalmente mais eficientes do que os ângulos de Euler, quatérnios também evitam singularidades cinemáticas e descontinuidades (Szeliski, 2010).

Derivado a matriz de rotação $R(\mathbf{q})$ encontra-se as equações dinâmicas que descrevem a rotação do sistema em função de quatérnios (Xu and Mandic, 2015),

$$\dot{\eta} = -\frac{1}{2}\boldsymbol{\varepsilon}^\top \boldsymbol{\omega}, \quad (4)$$

$$\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}[\eta I_3 + S(\boldsymbol{\varepsilon})]\boldsymbol{\omega}, \quad (5)$$

sendo $\boldsymbol{\omega} \in \mathbb{R}^3$ o vetor de velocidades angulares em coordenadas locais. Este vetor é diretamente medido por um sensor do tipo giroscópio acoplado ao corpo.

2.2 Modelo de Perspectiva da Câmera

O modelo de perspectiva de uma câmera ideal (*pinhole*) descreve a relação entre um ponto tridimensional $\boldsymbol{\xi}_i \in \mathbb{R}^3$ e sua projeção $\mathbf{m}_i \in \mathbb{R}^2$ no plano da imagem (medida em *pixels*). De acordo com Mariottini and Prattichizzo (2005), uma câmera de orientação $\mathbf{q}$ e posição $\mathbf{p} \in \mathbb{R}^3$, percebe a projeção do ponto $\boldsymbol{\xi}_i$ de acordo com a equação²

$$\lambda \mathbf{m}_i = K(R(\mathbf{q}) \boldsymbol{\xi}_i + \mathbf{p}) \quad (6)$$

¹O operador $S(\boldsymbol{\varepsilon})$ é o produto vetorial dado por,

$$S(\boldsymbol{\varepsilon}) = \begin{bmatrix} 0 & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & 0 & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & 0 \end{bmatrix} \quad \therefore \quad S(\boldsymbol{\varepsilon})\boldsymbol{\varepsilon} = 0. \quad (1)$$

²Considera-se a notação $\mathbf{m}_i = \begin{bmatrix} m_i \\ 1 \end{bmatrix}$.

sendo o parâmetro λ um escalar desconhecido. A matriz $K \in \mathbb{R}^{3 \times 3}$ – denominada “matriz de parâmetros intrínsecos” da câmera – descreve os parâmetros de distância focal, coeficientes de distorção e resolução (número de *pixels* por imagem).

É possível eliminar a dependência do parâmetro desconhecido λ no modelo da câmera ao multiplicar ambos os lados da equação (6) por $S(\mathbf{m}_i)$. Desta forma, obtém-se a equação

$$0 = S(\mathbf{m}_i)K(R(\mathbf{q}) \boldsymbol{\xi}_i + \mathbf{p}) \quad (7)$$

dado que $S(\mathbf{m}_i) \mathbf{m}_i = 0$.

2.3 Definição do Problema

O problema sendo abordado no presente artigo consiste em estimar a posição $\mathbf{p}$ e a orientação $\mathbf{q}$ de uma câmera a partir de suas imagens e de sensores inerciais acoplados à mesma. Em particular, busca-se um algoritmo capaz de fundir medições provenientes de um acelerômetro $\mathbf{a}_{acc} \in \mathbb{R}^3$ e um giroscópio $\boldsymbol{\omega}_{gyr}$, com projeções $\boldsymbol{\xi}_i$ de *landmarks* conhecidos, a fim de encontrar a estimativa da posição e orientação da câmera. Vale ressaltar que o processamento da imagem responsável pela aquisição das projeções $\boldsymbol{\xi}_i$ dos *landmarks* está fora do escopo do presente trabalho.

3 Modelagem Matemática

Para o desenvolvimento do algoritmo de estimação, apenas as relações entre os sensores e os graus de liberdade de um corpo rígido são descritos. Desta forma, desenvolve-se um modelo de sistemas autônomos com ampla aplicação, podendo ser utilizado na descrição de diversos veículos autônomos.

3.1 Descrição do Processo

O modelo matemático do sistema parte da suposição que as derivadas das acelerações lineares $\dot{\mathbf{a}} \in \mathbb{R}^3$ e angulares $\dot{\boldsymbol{\alpha}} \in \mathbb{R}^3$ são desconhecidas. Explorando a forma como o Filtro de Kalman Estendido parametriza incertezas no modelo, consideramos que estes sinais são “passeios aleatórios” descritos por $\dot{\mathbf{a}} = \boldsymbol{\delta}_a$ e $\dot{\boldsymbol{\alpha}} = \boldsymbol{\delta}_\alpha$, sendo $\boldsymbol{\delta}_i \in \mathbb{R}^3$ uma variável aleatória Gaussiana de média zero e covariância $Q_i \in \mathbb{R}^{3 \times 3}$, isto é, $\boldsymbol{\delta}_i \sim \mathcal{N}(0, Q_i)$.

O restante do modelo é equacionado de forma determinística,

$$\begin{aligned} \dot{\mathbf{q}} &= G(\mathbf{q})\boldsymbol{\omega} = F(\boldsymbol{\omega})\mathbf{q} \\ \dot{\boldsymbol{\omega}} &= \boldsymbol{\alpha} \\ \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a} \end{aligned} \quad (8)$$

sendo que $G(\mathbf{q})$, $F(\boldsymbol{\omega}) \in \mathbb{R}^{4 \times 3}$ são derivados de

(4) e (5):

$$F(\omega) = \frac{1}{2} \begin{bmatrix} 0 & -\omega^\top \\ \omega & -S(\omega) \end{bmatrix}, \quad G(\mathbf{q}) = \frac{1}{2} \begin{bmatrix} -\varepsilon^\top \\ \eta I_3 + S(\varepsilon) \end{bmatrix}. \quad (9)$$

3.2 Descrição dos Sensores Inerciais

As medidas inerciais provenientes do giroscópio $\omega_{gyr} \in \mathbb{R}^3$ e do acelerômetro $\mathbf{a}_{gyr} \in \mathbb{R}^3$ são corrompidas por sinais de ruído branco δ_i e por um erro de escorregamento $\mathbf{b}_i$ (*bias*) desconhecido. Assim, o giroscópio é descrito por,

$$\omega_{gyr} = \omega + \mathbf{b}_{gyr} + \delta_{gyr}, \quad (10)$$

e, de forma similar, o equacionamento do acelerômetro é dado por,

$$\mathbf{a}_{acc} = R^\top(\mathbf{q})(\mathbf{a} - \mathbf{g}) + \mathbf{b}_{acc} + \delta_{acc}. \quad (11)$$

onde $\mathbf{g} \in \mathbb{R}^3$ é o vetor de aceleração da gravidade (por exemplo $\mathbf{g} = [0 \ 0 \ -9.81]^\top \text{ m/s}^2$).

Uma vez que os valores dos *bias* são desconhecidos e variantes no tempo, eles serão considerados estados do EKF e estimados *online*. Uma vez que são desconhecidos, considera-se que estes sinais também são gerados por processos estocásticos do tipo “passeio aleatório” descritos por $\dot{\mathbf{b}}_{acc} = \delta_{\mathbf{b}_{acc}}$ e $\dot{\mathbf{b}}_{gyr} = \delta_{\mathbf{b}_{gyr}}$.

4 Aplicação ao Filtro de Kalman Estendido

Para utilizar o filtro de Kalman é necessário colocar as equações do processo (8) na forma de espaço de estados no domínio discreto,

$$\begin{aligned} \mathbf{x}_{k+1} &= f_k(\mathbf{x}_k) + \delta_{\mathbf{x}_k} \\ \mathbf{z}_k &= h_k(\mathbf{x}_k) + \delta_{\mathbf{z}_k}. \end{aligned} \quad (12)$$

sendo $\mathbf{x}_k$ o vetor de estados do processo e $\mathbf{z}_k$ e vetor de sensores ou medidas de correções na k -ésima amostra. Os modelos $f_k(\mathbf{x}_k)$ e $h_k(\mathbf{x}_k)$ descrevem a porção determinística da dinâmica do processo e da leitura dos sensores, enquanto que as entradas $\delta_{\mathbf{x}_k}$ e $\delta_{\mathbf{z}_k}$ representam a porção estocástica.

O Filtro de Kalman permite obter a estimativa de uma densidade de probabilidade gaussiana para o vetor de estados $\mathbf{x}_k$, ou seja:

$$\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_k, P_k) \quad (13)$$

onde $\hat{\mathbf{x}}_k$ representa a média estimada do estados e P_k representa matriz de a covariância desta estimativa.

Para realizar a implementação do filtro de Kalman Estendido é necessário definir as funções $f_k(\mathbf{x}_k)$ e $h_k(\mathbf{x}_k)$, bem como suas respectivas linearizações:

$$A_k(\mathbf{x}_k) = \frac{\partial f_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}, \quad C_k(\mathbf{x}_k) = \frac{\partial h_k(\mathbf{x}_k)}{\partial \mathbf{x}_k}. \quad (14)$$

Além disso, deve estabelecer as matrizes de covariância $Q_{\mathbf{x}_k}$ e $Q_{\mathbf{z}_k}$ das entradas estocásticas do sistema. Com base nestas informações, pode-se implementar o algoritmo descrito na Tabela 1.³

Tabela 1: Algoritmo do Filtro de Kalman Estendido

Etapa de predição

1. $\hat{\mathbf{x}}_k^* = f_k(\hat{\mathbf{x}}_{k-1})$
2. $P_k^* = A_k(\hat{\mathbf{x}}_{k-1}) P_{k-1} A_k^\top(\hat{\mathbf{x}}_{k-1}) + Q_{\mathbf{x}_{k-1}}$

Etapa de Correção

1. $S_k = C_k(\hat{\mathbf{x}}_k^*) P_k^* C_k^\top(\hat{\mathbf{x}}_k^*) + Q_{\mathbf{z}_k}$
2. $L_k = P_k^* C_k^\top(\hat{\mathbf{x}}_k^*) S_k^{-1}$
3. $W_k = I - L_k C_k(\hat{\mathbf{x}}_k^*)$
4. $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^* + L_k(\mathbf{z}_k - h_k(\hat{\mathbf{x}}_k^*))$
5. $P_k = W_k P_k^*$

4.1 Modelo do processo

Os estados do sistema são definidos a partir de (8) e dos *bias* dos sensores. Desta forma, $\mathbf{x}_k \in \mathbb{R}^{25}$,

$$\mathbf{x}_k := [\mathbf{q}_k^\top \ \omega_k^\top \ \alpha_k^\top \ \mathbf{p}_k^\top \ \mathbf{v}_k^\top \ \mathbf{a}_k^\top \ \mathbf{b}_{gyr_k}^\top \ \mathbf{b}_{acc_k}^\top]^\top, \quad (15)$$

é o vetor de estados estimados. O modelo $f(\mathbf{x}_k)$ pode ser obtido a partir da discretização *Euler Forward* das equações em (8):

$$f(\mathbf{x}_k) = \mathbf{x}_k + T_s \begin{bmatrix} F(\omega_k) \mathbf{q}_k \\ \alpha_k \\ 0_{3 \times 1} \\ \mathbf{v}_k \\ \mathbf{a}_k \\ 0_{9 \times 1} \end{bmatrix}, \quad (16)$$

onde T_s representa o período de amostragem do processamento do filtro. Desta forma, a linearização de $f(\mathbf{x}_k)$ é dada por,

$$A(\mathbf{x}_k) = I_{25} + T_s \text{diag}\{\Phi(\mathbf{x}_k), \Psi, \mathbf{0}_{6 \times 6}\}, \quad (17)$$

$$\Phi(\mathbf{x}_k) = \begin{bmatrix} F(\omega_k) & G(\mathbf{q}_k) & 0_{4 \times 3} \\ 0_{3 \times 4} & 0_{3 \times 3} & I_3 \\ 0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad (18)$$

$$\Psi = \begin{bmatrix} 0_{3 \times 3} & I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}. \quad (19)$$

4.2 Modelo dos sensores

Considera-se que o sistema realiza leituras ω_{gyr_k} e $\mathbf{a}_{gyr_k}$ de um sensor inercial conforme os respectivos modelos (10) e (11). Além disto, também

³As variáveis $\mathbf{x}_k^*$ e P_k^* denotam as estimativas *a priori* (antes da correção) da médias e covariância dos estados.

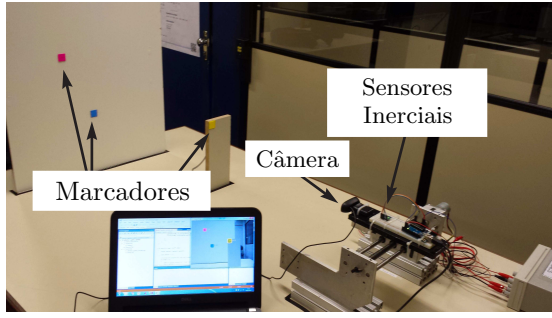


Figura 1: Ambiente para validação do algoritmo.

supõe-se que o sistema possui n *landmarks* de referência $\xi_1, \xi_2, \dots, \xi_n$ e as suas respectivas medições na imagem $\mathbf{m}_{1,k}, \mathbf{m}_{2,k}, \dots, \mathbf{m}_{n,k}$, com base na equação da câmera (7). Então pode-se organizar $\mathbf{z}_k$ e $h_k(\mathbf{x}_k)$ conforme

$$\mathbf{z}_k = \begin{bmatrix} \omega_{gyrk} \\ \mathbf{a}_{gyrk} \\ 0_{3n \times 1} \end{bmatrix} \quad (20)$$

$$h_k(\mathbf{x}_k) = \begin{bmatrix} \omega_k + \mathbf{b}_{gyrk} \\ R^T(\mathbf{q}_k)(\mathbf{a}_k - \mathbf{g}) + \mathbf{b}_{ackk} \\ \Upsilon_{1,k}(R(\mathbf{q}_k)\xi_1 + \mathbf{p}_k) \\ \vdots \\ \Upsilon_{n,k}(R(\mathbf{q}_k)\xi_n + \mathbf{p}_k) \end{bmatrix} \quad (21)$$

onde matriz auxiliar $\Upsilon_{i,k}$ é definida conforme:

$$\Upsilon_{i,k} = S(\tilde{\mathbf{m}}_{i,k}) K(R(\mathbf{q}_k)\xi_i + \mathbf{p}_k). \quad (22)$$

A respectiva linearização da função não-linear $h_k(\mathbf{x}_k)$ é obtida por

$$C_k(\mathbf{x}_k) = [\Gamma_k(\mathbf{x}_k) \quad \Pi_k(\mathbf{x}_k) \quad \Omega] \quad (23)$$

utilizando as seguintes matrizes auxiliares:

$$\Gamma_k(\mathbf{x}_k) = \begin{bmatrix} 0_{3 \times 4} & I_3 & 0_{3 \times 3} \\ J'(\mathbf{q}_k, \mathbf{a}_k - \mathbf{g}) & 0_{3 \times 3} & 0_{3 \times 3} \\ \Upsilon_{i,k} J(\mathbf{q}_k, \xi_1) & 0_{3 \times 3} & 0_{3 \times 3} \\ \vdots & \vdots & \vdots \\ \Upsilon_{n,k} J(\mathbf{q}_k, \xi_n) & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad (24)$$

$$\Pi_k(\mathbf{x}_k) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & R^T(\mathbf{q}_k) \\ \Upsilon_{i,k} & 0_{3 \times 3} & 0_{3 \times 3} \\ \vdots & \vdots & \vdots \\ \Upsilon_{n,k} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad (25)$$

$$\Omega = \begin{bmatrix} I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & I_3 \\ 0_{3n \times 3} & 0_{3n \times 3} \end{bmatrix}. \quad (26)$$

As seguintes identidades de derivação parcial foram aqui utilizadas:

$$J(\mathbf{q}, \mathbf{u}) = \frac{\partial R(\mathbf{q})\mathbf{u}}{\partial \mathbf{q}} = \quad (27)$$

$$2 [S(\epsilon)\mathbf{u} \quad -S(\mathbf{u})\boldsymbol{\eta} + W(\epsilon, \mathbf{u})],$$

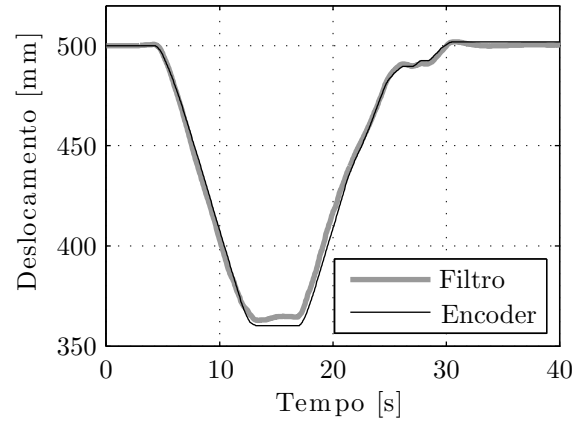


Figura 2: Comparação entre valor estimado e valor medido pelo Encoder.

$$J'(\mathbf{q}, \mathbf{u}) = \frac{\partial R^T(\mathbf{q})\mathbf{u}}{\partial \mathbf{q}} = \quad (28)$$

$$2 [-S(\epsilon)\mathbf{u} \quad S(\mathbf{u})\boldsymbol{\eta} + W(\epsilon, \mathbf{u})],$$

$$W(\epsilon, \mathbf{u}) := \frac{\partial S^2(\epsilon)\mathbf{u}}{\partial \epsilon} = \epsilon \mathbf{u}^T - 2 \mathbf{u} \epsilon^T + \epsilon^T \mathbf{u} I_3. \quad (29)$$

Estas relações são válidas para qualquer quaternion unitário $\mathbf{q}$ e qualquer vetor $\mathbf{u} \in \mathbb{R}^3$.

5 Resultados Experimentais

Para realizar a validação do filtro proposto foi montado um experimento que consiste na fusão dos dados provenientes de uma câmera Logitech C270 HD 720p e de um sensor MPU-6050 contendo três acelerômetros e três giroscópios. A leitura do sensor inercial é realizada por uma placa de desenvolvimento contemplando um microcontrolador ATmega328P, de 16 MHz. Estes dados são enviados via porta serial para o computador responsável pela implementação do filtro de Kalman Estendido e pelo processamento da imagem, que é realizado com o auxílio de uma biblioteca de visão computacional. Em particular, a detecção dos marcadores de posições ξ_i , e a informação de sua posição no plano da imagem $\mathbf{m}_i$ (medida em *pixels*) é realizada por algoritmos disponíveis na *Open Source Computer Vision Library* (OpenCV) (Bradski, 2002).

Primeiramente o sistema foi testado sobre uma base móvel limitando seu deslocamento ao eixo y , como pode ser visto na Fig. 1. Desta forma é possível obter dados precisos sobre o deslocamento real do sistema, a fim de validar a estimação obtida pelo filtro proposto. Os resultados deste teste podem ser vistos nas Figuras 2 e 3, que apresentam respectivamente uma comparação entre os dados provenientes do encoder e do estimador, e o erro resultante desta comparação. Neste teste a câmera começou na posição $\mathbf{p}_0 = [10^3 \ 500 \ 200]$

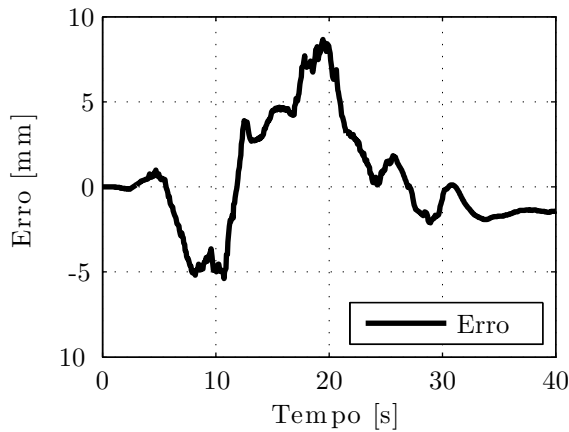


Figura 3: O erro m ximo na trajet ria estimada fica abaixo de 10 mm.

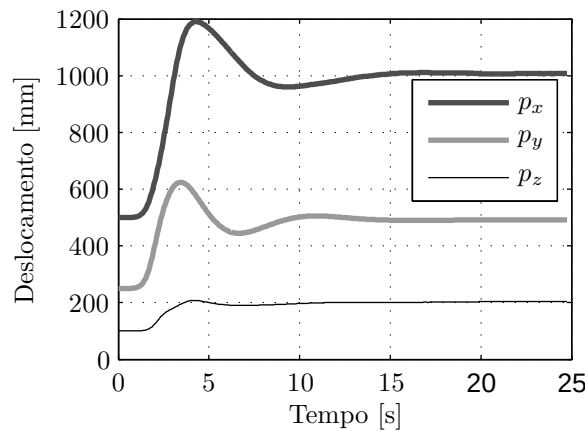


Figura 4: Evolu o da posi o estimada da c mera $\mathbf{p} = [p_x \ p_y \ p_z]^T$ para condi es iniciais desconhecidas.

mm, se deslocou at  $\mathbf{p} = [10^3 \ 360 \ 200]$ mm e retornou   $\mathbf{p}_0$. Como pode ser observado na Fig. 3, durante toda a trajet ria da c mera o erro de estima o se manteve abaixo de 1 cm.

O segundo teste consistiu em alterar os par metros da posi o inicial do sistema para determinar a capacidade do filtro de convergir o erro de estima o a zero. Neste teste a c mera se manteve fixa na posi o $\mathbf{p} = [10^3 \ 500 \ 200]$ mm com orienta o $\mathbf{q} = [0.5 \ -0.5 \ -0.5 \ 0.5]$. J  o filtro foi inicializado na posi o $\hat{\mathbf{p}}_0 = [500 \ 250 \ 100]$ mm e orienta o $\hat{\mathbf{q}}_0 = [0.5773 \ -0.4083 \ -0.4083 \ 0.5773]$. Os resultados obtidos de converg ncia do filtro s o mostrados nos gr ficos das Figuras 4 e 5, respectivamente para a estima o da posi o e orienta o do sistema. Estas figuras demonstram claramente que o filtro   capaz de convergir   estima o correta dos estados, apesar de ser inicializado em condi es iniciais consideravelmente diferentes das do sistema.

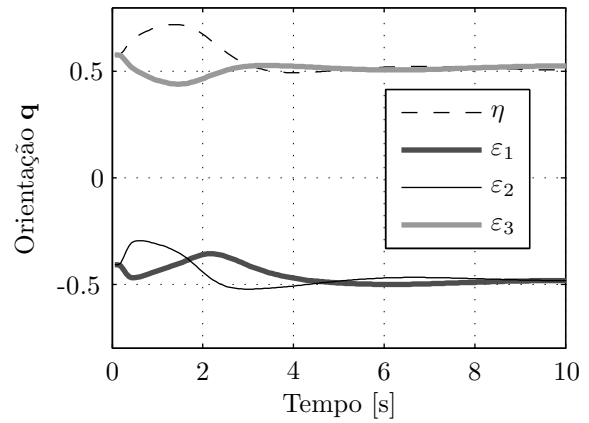


Figura 5: Evolu o da estima o dos quatern os da c mera $\mathbf{q}$ para condi es iniciais desconhecidas.

6 Conclus o

O presente artigo prop s um algoritmo para a fus o de sensores inerciais e vis o computacional na tarefa de estima o da posi o e orienta o (pose) de um corpo r gido. Um modelo baseado na orienta o por quatern os foi utilizado, facilitando o desenvolvimento das matrizes Jacobianas utilizadas pelo Filtro de Kalman Estendido. A modelagem da proje o dos *landmarks* no plano da c mera gerou um conjunto de sensores virtuais prontamente integrados ao algoritmo de fus o de sensores. Os resultados experimentais permitem a conclus o de que o algoritmo proposto   capaz de estimar a pose de um corpo r gido de forma precisa, apesar de lidar com sensores de baixo custo.

Trabalhos futuros investigar o a perda de informa o proveniente da c mera devido   oclus es e instabilidade dos algoritmos de tratamento de imagens. Imagina-se que a estima o da pose do sistema se mantenha adequada nestas situa es, desde que a estima o n o fique exclusivamente dependente dos sensores inerciais por um tempo muito alto.

Agradecimentos

Este trabalho foi parcialmente apoiado pela Capes e CNPq (309272/2015-7) e em coopera o com a Hewlett-Packard Brasil Ltda. e com recursos provenientes da Lei de Inform tica (Lei n mero 8.248, de 1991).

Refer ncias

- Altug , E., Ostrowski, J. P. and Taylor, C. J. (2003). Quadrotor control using dual camera visual feedback, *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, Vol. 3, IEEE, pp. 4294–4299.

- Bradski, G. (2002). Opencv: Examples of use and new applications in stereo, recognition and tracking, *Proc. Intern. Conf. on Vision Interface (VI?2002)*, p. 347.
- Chai, L., Hoff, W. A. and Vincent, T. (2002). Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality, *Presence: Teleoperators and Virtual Environments* **11**(5): 474–492.
- Engel, J., Sturm, J. and Cremers, D. (2012). Accurate figure flying with a quadcopter using onboard visual and inertial sensing, *IMU* **320**: 240.
- Goldberg, S. B., Maimone, M. W. and Matthies, L. (2002). Stereo vision and rover navigation software for planetary exploration, *Aerospace Conference Proceedings, 2002. IEEE*, Vol. 5, IEEE, pp. 5–2025.
- Janabi-Sharifi, F. and Marey, M. (2010). A kalman-filter-based method for pose estimation in visual servoing, *Robotics, IEEE Transactions on* **26**(5): 939–947.
- Kornuta, T. and Zieliński, C. (2013). Robot control system design exemplified by multi-camera visual servoing, *Journal of Intelligent & Robotic Systems* **77**(3): 499–523.
- Kristiansen, R., Nicklasson, P. J. and Gravdahl, J. T. (2005). Satellite attitude tracking by quaternion-based backstepping, *Proceedings of the 16 th IFAC World Congress*, Citeseer.
- Mariottini, G. L. and Prattichizzo, D. (2005). Egt for multiple view geometry and visual servoing: robotics vision with pinhole and panoramic cameras, *Robotics & Automation Magazine, IEEE* **12**(4): 26–39.
- Mart nez, C., Mondrag n, I. F., Olivares-M ndez, M. A. and Campoy, P. (2011). On-board and ground visual pose estimation techniques for uav control, *Journal of Intelligent & Robotic Systems* **61**(1-4): 301–320.
- N tzi, G., Weiss, S., Scaramuzza, D. and Siegwart, R. (2011). Fusion of imu and vision for absolute scale estimation in monocular slam, *Journal of intelligent & robotic systems* **61**(1-4): 287–299.
- Palomeras, N., Nagappa, S., Ribas, D., Grac as, N. and Carreras, M. (2013). Vision-based localization and mapping system for auv intervention, *OCEANS-Bergen, 2013 MTS/IEEE*, IEEE, pp. 1–7.
- Raush, G., Freire, F., Khamashta, M., Codina, E., Set , J. and Cangas, J. (2013). *Precision agriculture '13*, Wageningen Academic Publishers, Wageningen, chapter Hydraulic robot arm controlled by visual servoing, pp. 365–371.
- Rehbinder, H. and Ghosh, B. K. (2003). Pose estimation using line-based dynamic vision and inertial sensors, *Automatic Control, IEEE Transactions on* **48**(2): 186–199.
- Silveira, G., Malis, E. and Rives, P. (2008). An efficient direct approach to visual slam, *Robotics, IEEE Transactions on* **24**(5): 969–979.
- Steder, B., Grisetti, G., Stachniss, C. and Burgard, W. (2008). Visual slam for flying vehicles, *Robotics, IEEE Transactions on* **24**(5): 1088–1093.
- Szeliski, R. (2010). *Computer vision: algorithms and applications*, Springer Science & Business Media.
- Xu, D. and Mandic, D. P. (2015). The theory of quaternion matrix derivatives, *Signal Processing, IEEE Transactions on* **63**(6): 1543–1556.
- Yang, J., Chung, S.-J., Hutchinson, S., Johnson, D. and Kise, M. (2013). Vision-based localization and mapping for an autonomous mower, *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, pp. 3655–3662.