

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica
ELE00070 Tópicos Especiais em Controle e Automação I

Estimação de Posição e Orientação

Prof. Walter Fetter Lages

11 de agosto de 2008

Considerando-se que o objetivo principal em geral é desenvolver técnicas para controle de robôs móveis, a estimação de posição e orientação é abordada aqui como forma de permitir a implementação em tempo real e simulação realística das estratégias de controle propostas. Várias técnicas estão disponíveis para a obtenção da posição e da orientação de robôs móveis, utilizando-se encoders, visão computacional [31] [27] [29], sonar [4] [13] [19] e scanner a laser [26]. Uma descrição bastante detalhada dos métodos para estimação de posição e orientação de robôs móveis pode ser encontrada em [2].

O método de *dead-reckoning* consiste em calcular-se o deslocamento linear e angular do robô a partir dos deslocamentos das rodas obtidos por encoders incrementais. É conhecido [9] que para trajetórias longas, o método de *dead-reckoning* [2] não é adequado para estimação da posição de robôs móveis, pois pela própria natureza do método, os erros vão sendo acumulados ao integrar-se os deslocamentos para obtenção da posição e orientação. Os erros presentes no método de *dead-reckoning* podem ser classificados em erros sistemáticos e erros não sistemáticos.

Os erros sistemáticos são causados basicamente por incertezas nos parâmetros geométricos utilizados para calcular a posição e orientação do robô. Para robôs atuados diferencialmente, as principais causas de erros sistemáticos são a) a diferença entre o diâmetro nominal médio e o diâmetro real das rodas; b) a diferença entre os diâmetros médios das duas rodas de tração e c) a diferença entre a distância entre rodas nominal e real.

Erros não-sistemáticos ocorrem devido a escorregamento das rodas, imperfeições no solo e fenômenos similares. Obviamente, por estarem constantemente presentes, os erros sistemáticos são os que mais influenciam a estimativa final de posição e orientação. No entanto, através de procedimento de calibração adequados [3] é possível reduzir-se estes erros sistemáticos em até duas ordens de

grandeza.

Os sensores baseados em visão computacional, por outro lado, não sofrem estes problemas. Em geral, sistemas baseados em visão operam através do reconhecimento de *landmarks* existentes no ambiente. Como *landmarks* podem ser utilizados padrões especiais como códigos de barras distribuídos pelo ambiente [16] ou características existentes naturalmente no ambiente [20].

Para robôs operando em ambientes externos, é comum a utilização de *digital elevation maps* (DEM) [25] [24]. Um DEM é basicamente um perfil da linha do horizonte que é armazenado em um banco de dados. A localização do robô é feita comparando-se o perfil de horizonte visto pelo robô com os perfis armazenados no banco de dados.

Pode-se também utilizar visão computacional sem *landmarks*. Neste enfoque procura-se estimar o movimento do robô a partir do movimento das imagens captadas pela câmera [18] [17] [23]. No entanto, estes métodos apresentam desvantagens semelhantes as apresentadas pelo *dead-reckoning*, pois o movimento detectado pelo sistema de visão será integrado para obtenção da posição e orientação, com possível acúmulo de erros.

Outra desvantagem dos sistemas baseados em visão computacional é o tempo requerido para o processamento da imagem, o que dificulta a sua implementação em tempo real.

Dados provenientes de sonares de ultra-som podem ser processados em um tempo menor, mas a precisão de orientação não é boa [10]. Os problemas encontrados para se obter uma estimativa da posição e orientação do robô através de medidas de distância por ultra-som devem-se basicamente aos efeitos de reflexão, difração e ângulo de abertura a que estão sujeitas as ondas sonoras [14]. Um método para estimação de posição e orientação de robôs móveis é detalhado em [10]. Em geral, os métodos que utilizam sensores de ultra-som baseiam-se no casamento de um mapa de ultra-som armazenado no robô com a resposta obtida pelos sensores.

As dificuldades inerentes ao uso de sensores de ultra-som podem ser evitadas utilizando-se medidas de distância obtidas a partir de um *scanner* a laser, como em [26]. Basicamente, os mesmos métodos utilizados com sonares podem ser utilizados com *scanner* a laser, tendo-se a vantagem de que os efeitos de reflexão, difração e ângulo de abertura do feixe são minimizados. No entanto, o custo de um *scanner* a laser é bem superior ao de um sistema de sonares. Além disso, apresenta inconvenientes para a montagem sobre o robô, pois não pode ser montado distribuído em torno do mesmo, como ocorre com um *array* de sonares.

Com o desenvolvimento de plataformas inerciais de estado sólido de baixo custo, este tipo de sensor também foi proposto para robôs móveis [6][15]. Como este tipo de sensor fornece informação de aceleração linear e angular, torna-se necessário uma dupla integração para obter-se posição e orientação do robô. Por-

tanto, este também é um método de *dead-reckoning*. No entanto, ao contrário dos métodos utilizando encoders, o conhecimento dos parâmetros geométricos do robô não é necessário, pois o sensor fornece diretamente a aceleração linear e angular. Por outro lado, devido à necessidade do sinal de saída do sensor ser integrado duas vezes, o efeito do acúmulo dos erros torna-se mais significativo. Conseqüentemente, o uso deste tipo de sensor requer uma modelagem bastante cuidadosa dos erros.

Embora sistemas GPS (*Global Positioning System*) não sejam adequados para robôs móveis que devam operar em ambientes internos, estes sistemas quando operando de forma diferencial podem fornecer a posição do robô com uma precisão razoável, o que o torna uma boa alternativa para uso em ambientes externos. Recentemente tem sido realizados esforços no sentido de desenvolver sistemas LPS (*Local Positioning System*) [28]. Contrariamente ao GPS, onde cada receptor determina sua posição com relação a um referencial fixo, nos sistemas LPS sendo desenvolvidos a posição de cada *tag* (dispositivos sendo rastreados) é determinada por um agente central. Portanto, para utilização deste sistema em robôs móveis seria necessário a existência de comunicação entre este agente central e o robô. Isto representa sérias limitações à autonomia do robô. Tanto devido à possibilidade de perda da comunicação quanto à complexidade introduzida caso hajam vários robôs operando na mesma área. No entanto, é razoável imaginar-se que dentro de algum tempo surgirão sistemas LPS operando de forma similar ao GPS, ou seja, de forma que os receptores possam determinar a sua própria posição.

Sabe-se que uma parcela significativa dos erros em sistemas de *dead-reckoning* deve-se ao erro de orientação [2]. Para reduzir os efeitos do erro de orientação na estimativa final de posição obtida por *dead-reckoning*, alguns trabalhos [15] [22] utilizam uma bússola digital. A idéia é obter-se uma estimativa mais confiável para a orientação através da fusão de dados dos encoders e da bússola. Esta fusão de dados é realizada utilizando-se um filtro de Kalman estendido, devido as não linearidades presentes no modelo cinemático do robô.

Na realidade, a fusão de dados de sensores operando sob princípios diferentes utilizando o filtro de Kalman estendido é um procedimento usual para a estimação de posição e orientação de robôs móveis [10] [12]. Como é percebido nos parágrafos anteriores, os sensores de posição e orientação podem ser classificados em dois tipos: sensores absolutos e sensores incrementais.

Nos sensores absolutos os erros não se propagam ao longo do tempo, mas estes sensores requerem um processamento sofisticado para obter-se uma estimativa. Esta complexidade de processamento causa dois problemas principais: a) como o robô está em movimento, ao término do processamento a posição e orientação do robô sofreram alterações, portanto a estimativa obtida está defasada e b) normalmente o tempo necessário para realizar o processamento é bem maior do que o período de amostragem do sistema de controle. Sensores incrementais, por ou-

tro lado, em geral são processados rapidamente, mas apresentam erros crescentes pois a posição e orientação são obtidas por integração.

A fusão de dados de sensores absolutos e de sensores incrementais permite se obter estimativas de posição e orientação a intervalos de tempos compatíveis com as necessidades do sistema de controle, sem que no entanto os erros cresçam de forma ilimitada. Basicamente, o efeito resultante é de que os erros dos sensores incrementais são zerados cada vez que uma leitura dos sensores absolutos está disponível.

Considerando-se que este trabalho enfoca o controle de robôs móveis com objetivo de convergência para um ponto, as trajetórias percorridas pelo robô serão tipicamente curtas. Se os parâmetros geométricos do robô forem calibrados convenientemente, os erros de *dead-reckoning* acumulados durante as trajetórias de teste serão desprezíveis. Além disso, para avaliação do desempenho do controlador, o que importa são a posição e orientação medidas pelo sensor, e que deverão convergir para os valores desejados. Se os valores medidos pelo sensor correspondem ou não aos valores reais, é uma questão que está além do escopo do problema de controle. Portanto, o método utilizado para determinação da posição e orientação do robô utilizado é basicamente *dead-reckoning*. No entanto, visando avaliar os problemas de integração das técnicas baseadas na fusão de dados utilizando o filtro de Kalman com os métodos de controle propostos, foram também implementados métodos de estimação baseados na fusão dos dados dos encoders com um sistema de visão computacional e com uma bússola digital.

Neste tipo de problema é comum considerar-se que todos os sensores envolvidos possuem a mesma taxa de amostragem. Isto é, assume-se que todos os sensores são capazes de fornecer medidas periodicamente e que este período é constante e igual para todos os sensores. No entanto, quando são utilizados sensores com diferentes princípios físicos, é comum a situação onde cada sensor exige uma taxa de amostragem diferente. Nestes casos, a simulação do sistema, ou uma implementação da fusão de dados *off-line* não é muito diferente do caso com uma única taxa de amostragem. Porém, uma implementação para fusão dos dados em tempo real apresenta uma série de problemas relacionados ao processamento para aquisição de dados de cada sensor, com o sincronismo entre os sensores e destes com o filtro de Kalman.

1 Método de *Dead-Reckoning*

Uma das técnicas mais simples para estimação de posição (e orientação) consiste em contar-se a quantidade de pulsos obtidos a partir dos encoders de cada roda em um certo intervalo de tempo. Assumindo-se que não há escorregamento e conhecendo-se o raio das rodas (r) e o número de pulsos por volta dos encoders

(II), pode-se calcular o deslocamento linear das rodas no intervalo de tempo

$$\Delta D_r(k) = \frac{2\pi r N P_r(k)}{\Pi}$$

$$\Delta D_l(k) = \frac{2\pi r N P_l(k)}{\Pi}$$

onde

$\Delta D_r(k)$, $\Delta D_l(k)$ = deslocamento linear das rodas direita e esquerda no intervalo de tempo kT entre e $(k+1)T$;

$N P_r(k)$, $N P_l(k)$ = número de pulsos obtidos nos encoders das rodas direita e esquerda no intervalo de tempo entre kT entre e $(k+1)T$.

Os deslocamentos linear e do robô considerando-se como referência o ponto no centro axial das rodas podem ser calculados por

$$\Delta D(k) = \frac{\Delta D_r(k) + \Delta D_l(k)}{2}$$

$$\Delta \theta(k) = \frac{\Delta D_r(k) - \Delta D_l(k)}{2b}$$

onde $2b$ é a distancia axial entre as rodas.

Assumindo-se que a trajetória percorrida pelo veículo entre os instantes de tempo k e $k+1$ é uma reta tem-se

$$x_c(k+1) = x_c(k) + \Delta D(k) \cos \theta(k) \quad (1)$$

$$y_c(k+1) = y_c(k) + \Delta D(k) \sin \theta(k) \quad (2)$$

$$\theta(k+1) = \theta(k) + \Delta \theta(k) \quad (3)$$

As expressões (1) a (3) foram desenvolvidas considerando-se uma trajetória circular, mas são também exatas quando a trajetória é uma reta. Em [16] é feita uma análise semelhante considerando que a trajetória é um arco de circunferência. Na prática, verificou-se que os benefícios desta modelagem mais detalhada são mascarados pelos demais erros do sistema. Existem no entanto, erros de modelagem devido ao escorregamento das rodas, às irregularidades do solo, à não exatidão no conhecimento do raio das rodas e da distância entre elas, etc., que são estudados com maior detalhe em [3].

Na Fig. 1 apresenta-se uma simulação da trajetória realizada pelo robô IEEE-Rover e a trajetória estimada por *dead-reckoning*.

Observa-se que a trajetória estimada por *dead-reckoning* coincide com a trajetória realizada apenas nos instantes iniciais. Este comportamento deve-se ao efeito integral intrínseco à esta técnica de estimação. Como as estimativas são obtidas

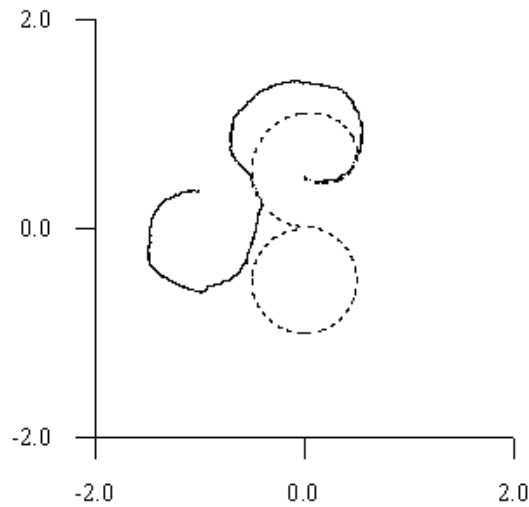


Figura 1: Trajetória do robô (pontilhada) e trajetória estimada por *dead-reckoning* (sólida).

através de somatórios, pode-se perceber que estes erros vão se acumulando. Isto faz com que esta estratégia de estimação do vetor de estado seja adequada apenas para trajetos pequenos, quando o erro acumulado pode ser desprezado.

2 Visão Computacional

A estimação de posição e orientação através de visão computacional consiste em obter-se uma ou mais imagens que contenham pontos conhecidos (posição e orientação conhecidas) do ambiente. Estes pontos são identificados na imagem e conhecendo-se os parâmetros do sistema de aquisição de imagens, a posição do robô em relação a eles pode ser computada. Como a posição e orientação dos pontos de referência em relação ao sistema de coordenadas global são conhecidas, calcula-se por transformações de coordenadas a posição e orientação do robô.

O sistema de processamento de imagens utilizado neste trabalho é baseado em *landmarks* como os apresentados na Fig. 2, que são fixados nas paredes do ambiente de trabalho da plataforma. A escolha deste tipo de landmark foi motivada por [5].

Ao serem vistos em perspectiva pela plataforma, estes *landmarks* sofrem distorções nos eixos vertical e horizontal. Como os *landmarks* são fixados à altura da câmera, a distorção no eixo vertical é devida apenas à distância da câmera ao landmark, enquanto a distorção no eixo horizontal se dá devido à distância e à orientação da câmera com relação ao *landmark*. No caso geral, o landmark aparecerá na imagem como uma elipse. A posição e os eixos desta elipse são detectados na

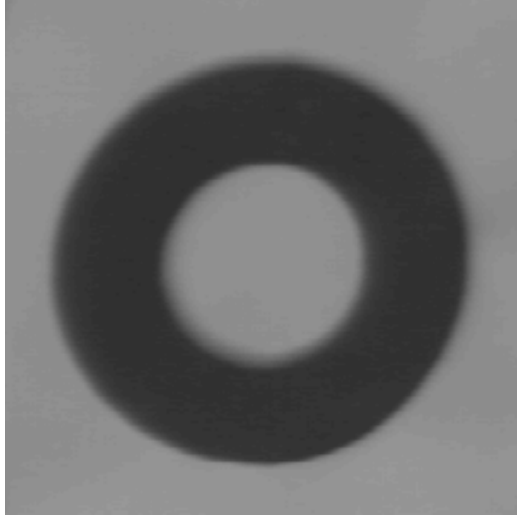


Figura 2: Landmark utilizado pelo sistema de visão.

imagem através da transformada de Hough generalizada [1] [30].

A posição e orientação do landmark em relação à câmera então será dada pela transformação de perspectiva [11]

$$Z = \lambda \left(1 + \frac{r_l}{a_y} \right)$$

$$\Theta = \pm \arccos(a_x, a_y) \quad (4)$$

onde Z é distância da câmera ao *landmark*, λ é a distância focal da câmera, r_l é raio do landmark, a_x e a_y são respectivamente os semi-eixos horizontal e vertical da elipse e Θ é a orientação da câmera com relação ao *landmark*.

A seguir, determina-se a matriz de transformação homogênea [7] entre o *landmark* e a câmera

$${}^vT_l = \begin{bmatrix} -\sin \Theta & \cos \Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \cos \Theta & \sin \Theta & 0 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz de transformação homogênea entre o sistema de coordenadas global e o sistema de coordenadas centrado em $x - c$ e y_c será dada por

$${}^gT_c = {}^gT_l {}^lT_v {}^vT_c$$

onde gT_l é a matriz de transformação homogênea entre o sistema de coordenadas localizado no centro do *landmark* e o sistema de coordenadas global, lT_v é a matriz

de transformação homogênea entre a câmera e o landmark e vT_c é a matriz de transformação homogênea entre sistema de coordenadas centrado em x_c e y_c e a câmera.

Como gT_c é constante pois depende somente da posição do *landmark* sendo observado, e vT_c depende apenas da geometria da plataforma e ${}^lT_v = ({}^vT_l)^{-1}$ tem-se que

$${}^gT_c = {}^gT_l ({}^vT_l)^{-1} {}^vT_c$$

Por outro lado, considerando a posição e orientação do robô, pode-se facilmente obter

$${}^gT_c = \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 & x_c \\ \sin \Theta & \cos \Theta & 0 & y_c \\ 0 & 0 & 1 & d_l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

onde d_l é a altura do ponto x_c, y_c . Portanto, tem-se que

$$x_c = ({}^gT_c)_{14} \quad (5)$$

$$y_c = ({}^gT_c)_{24} \quad (6)$$

e

$$\theta = \arctan 2(({}^gT_c)_{11}, ({}^gT_c)_{21}) \quad (7)$$

É importante observar, que devido à ambigüidade do sinal de Θ em (4), serão obtidos dois conjuntos de valores x_c, y_c e θ para um mesmo landmark. Esta ambigüidade pode ser removida observando-se dois ou mais *landmarks*, veja-se [5] [16] para maiores detalhes.

Utilizando-se as expressões (5)-(7), pode-se reconstituir a posição e orientação do robô. O erro de estimação de x_c, y_c e θ neste caso não é cumulativo como no caso anterior, mas o tempo necessário para estimação via processamento de imagens é maior, forçando o sistema de controle a usar uma estimativa não muito atual do estado ou uma taxa de amostragem muito baixa.

2.1 Fusão de Dados

Como o método de *dead-reckoning* e o processamento de imagens possuem características diferentes parece razoável supor-se que a fusão dos dados obtidos pelos dois sistemas produzirá uma estimativa de posição e orientação de melhor

qualidade. Esta fusão de dados pode ser feita utilizando-se o filtro de Kalman estendido.

Definindo-se

$$U(k) = \begin{bmatrix} \Delta D_r(k) \\ \Delta D_l(k) \end{bmatrix}$$

e

$$X(k) = \begin{bmatrix} x_c(k) \\ y_c(k) \\ \theta(k) \end{bmatrix}$$

pode-se escrever (1) - (3) na forma

$$x(k+1) = f(X(k), U(k)) + \omega(k) \quad (8)$$

onde $w(k)$ representa perturbações como escorregamento das rodas, incertezas nos parâmetros do modelo e erros nos encoders.

A expressão (8), representa a posição e orientação reais do robô, pois as perturbações foram levadas em consideração através de $w(k)$ enquanto as expressões (1)-(3) representam a posição do veículo sem considerar as perturbações, portanto incluem erro.

A saída do sistema de processamento de imagens, dada pelas expressões (5) - (7) é o vetor

$$y(k) = \begin{bmatrix} x_c(k) \\ y_c(k) \\ \theta(k) \end{bmatrix} + v(k) \quad (9)$$

considerado como uma observação do processo (8) que contém um erro de observação $v(k)$ denominado ruído de observação.

Fazendo-se as suposições usuais, isto é, $\omega(k)$ e $v(k)$ são ruídos gaussianos com média zero e $E \begin{bmatrix} \omega(k) \\ v(k) \end{bmatrix} \begin{bmatrix} \omega^T(n) & v^T(m) \end{bmatrix} = \begin{bmatrix} Q & S^T \\ S^T & r \end{bmatrix} \delta(k-n)$, com $Q \geq 0$ e $R > 0$, $E [\omega(k) \ X^T(n)] = 0$, para $k > n$, e $E [v(k) \ X^T(n)] = 0$, para $k > n$, o filtro de Kalman estendido pode ser utilizado para combinar as estimativas de posição obtidas pelos sistemas de *dead-reckoning* e processamento de imagens.

Expandindo-se $f(\cdot)$ em (8) em série de Taylor em torno do ponto $\hat{X}(k)$, $U(k)$ onde $\hat{X}(k)$ é a estimativa de $X(k)$ no instante k e desprezando os termos de ordem superior, resulta

$$f(X(k), U(k) + \omega(k)) = f(\hat{X}(k), U(k)) + F_k (X(k) - \hat{X}(k)) + G_k \omega(k) \quad (10)$$

onde

$$F_k = \left. \frac{\partial f(X(k), U(k))}{\partial X(k)} \right|_{X(k)=\hat{X}(k)}$$

e

$$G_k = \left. \frac{\partial f(X(k), U(k))}{\partial U(k)} \right|_{X(k)=\hat{X}(k)}$$

substituindo-se (10) em (8) resulta

$$X(k+1) = F_k X(k) + G_k \omega(k) + f\left(\hat{X}(k), U(k)\right) - F_k \hat{X}(k)$$

que juntamente com (9) forma um sistema linear variante no tempo com entrada $f\left(\hat{X}(k), U(k)\right) - F_k \hat{X}(k)$, que pode ser perfeitamente determinada no instante k e portanto é uma entrada determinística. Assim, pode-se utilizar o seguinte algoritmo para o filtro de Kalman estendido [8]

$$K(k) = (F_k P(k) F_k^T + G_k S) (P(k) + R)^{-1}$$

$$\hat{X}(k+1) = f\left(\hat{X}(k), U(k)\right) + K(k) (y(k+1) - \hat{X}(k))$$

$$P(k+1) = F_k P(k) F_k^T + G_k Q G_k - K(k) (P(k) + R) K^T(k)$$

com condições iniciais $P(0)$ e $\hat{X}(0)$

Na Fig. 3 pode-se ver um diagrama de blocos do método de estimação de posição e orientação utilizando fusão de dados através do filtro de Kalman estendido.

2.2 Resultados de Simulação

A Fig. 4 mostra uma trajetória descrita pelo robô IEEE-Rover com a posição e orientação estimadas através de dados obtidos por visão computacional e *dead-reckoning*, fundidos através do filtro de Kalman estendido.

3 Bússola Digital

O cálculo de orientação baseado na leitura de uma bússola digital $\theta_b(k)$ apresenta erros, ou seja, o valor verdadeiro $\theta_t(k)$ é corrompido por o ruído de medida $v(k)$, isto é,

$$\theta_b(k) = \theta_t(k) + v(k) \quad (11)$$

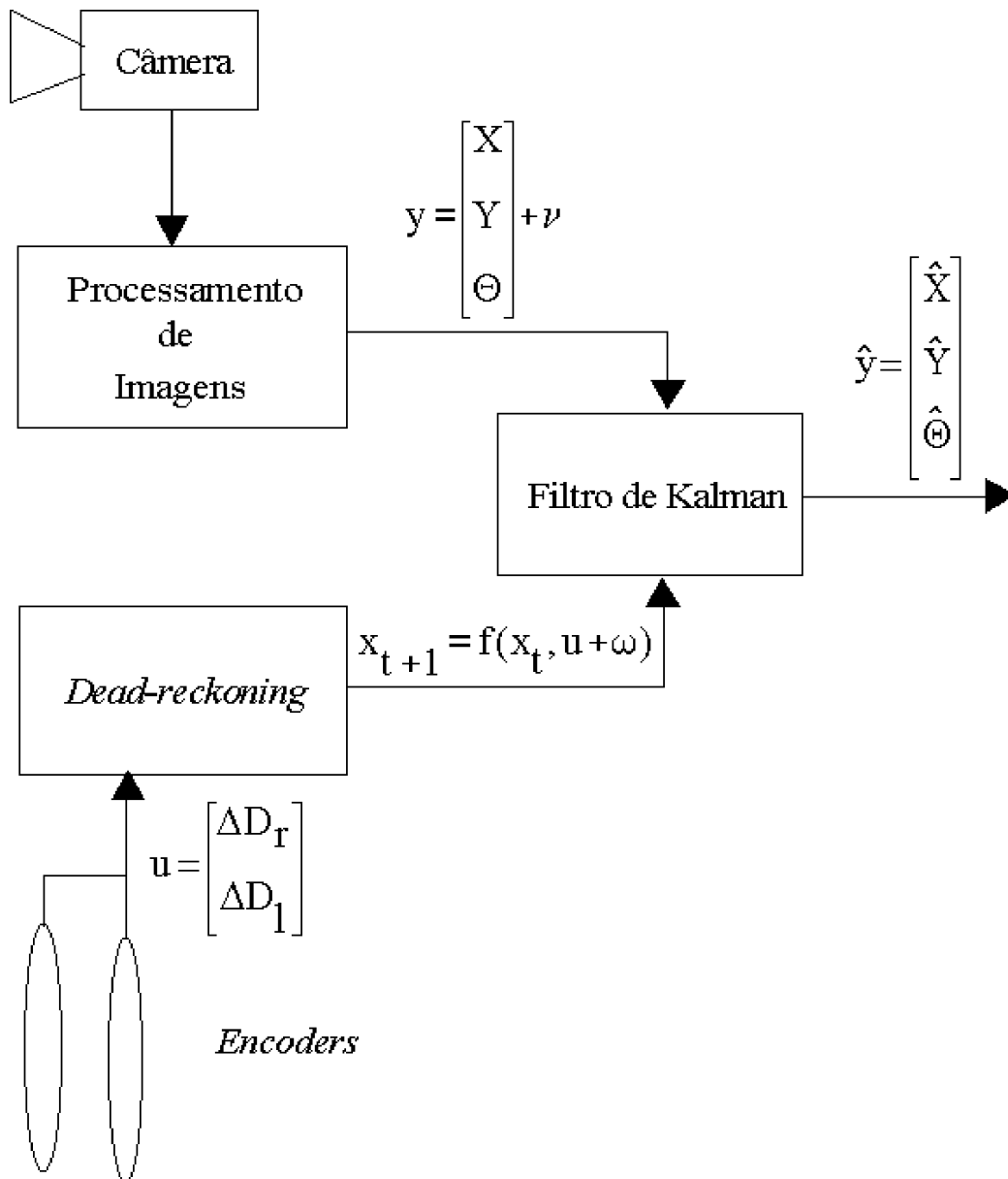


Figura 3: Diagrama de blocos da fusão de dados obtidos por visão computacional e encoders.

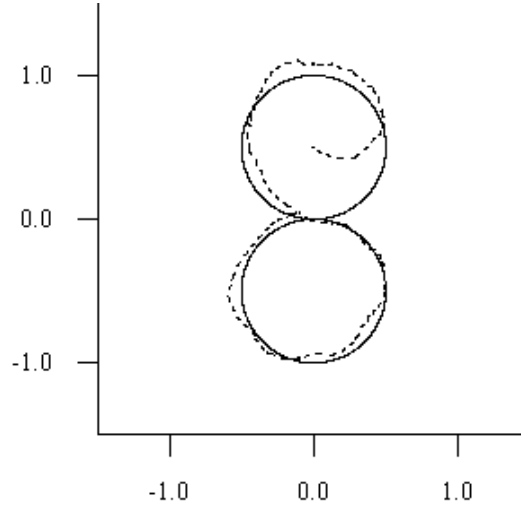


Figura 4: Trajetória de referência (sólida) e trajetória estimada por fusão de dados (tracejada).

Neste experimento foi utilizada a bússola digital Vector 2X Compass Module fabricada pela Precision Navigation, Inc., cujas especificações estão em [21].

A integração de ambos os sistemas, odômetro e bússola, foi realizada mediante um filtro de Kalman linear. A orientação calculada com base nos encoders, $\theta(k)$, pode ser modelada como o valor verdadeiro $\theta_t(k)$ mais um erro de orientação $\delta\theta(k)$, isto é,

$$\theta(k) = \theta_t(k) + \delta\theta(k) \quad (12)$$

O processo de observação $y(k)$, é a diferença de orientação entre ambos os sistemas, (12) e (11), portanto,

$$y(k) = \delta\theta(k) + v(k)$$

Supondo-se que um ruído branco $\omega(k)$ é adicionado ao incremento de orientação a cada instante de tempo, a orientação calculada pelos encoders no instante $k + 1$ será dada por

$$\theta(k + 1) = \theta(k) + \Delta\theta_t(k) + \omega(k) \quad (13)$$

onde $\Delta\theta_t(k)$ é o valor verdadeiro do incremento de orientação.

De (3), assumindo-se que as magnitudes são verdadeiras, tem-se

$$\theta_t(k + 1) = \theta_t(k) + \Delta\theta_t(k) \quad (14)$$

subtraindo-se (13) de (14) e utilizando-se (12) obtém-se

$$\delta\theta(k+1) = \delta\theta(k) + \omega(k) \quad (15)$$

que constitui a equação de estado para o filtro de Kalman.

Assim, o filtro de Kalman linear é escalar com os ruídos de medida, $v(k)$, e de estado, $\omega(k)$, não-correlacionados, permitindo que se estime o erro de orientação, para a seguir subtraí-lo do valor calculado a partir dos encoders, como mostrado na Fig. 5. A orientação corrigida é realimentada para calcular a posição, de modo a se reduzir o acúmulo de erro.

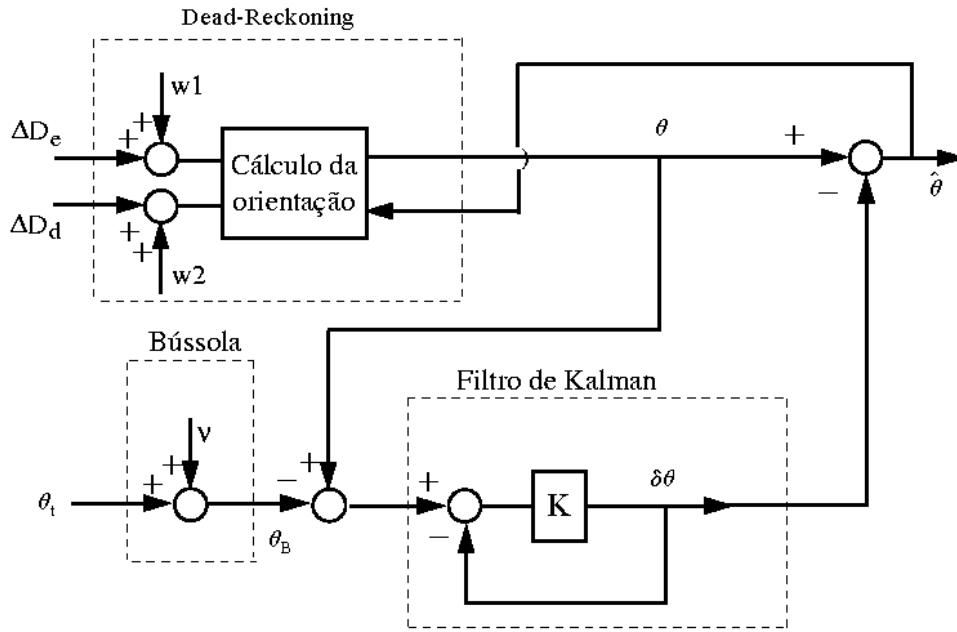


Figura 5: Diagrama de Blocos da Fusão de Dados da Bússola Digital e Encoders.

3.1 Resultados de Simulação

A estimação de posição e orientação através da fusão de dados de *dead-reckoning* e de uma bússola digital via filtro de Kalman foi simulada para uma trajetória circular com 1.37 m de raio. A Fig. 6 mostra a trajetória determinada por *dead-reckoning* (odômetro) e a trajetória estimada utilizando-se o filtro de Kalman, para uma velocidade constante do veículo de 0.14 m/s. A Fig. 7 mostra as diferentes orientações durante o movimento. Utilizou-se um período de amostragem de 50ms para os encoders e de 200 ms para a bússola. Um ruído de medida com desvio padrão de 4°/s foi utilizado, e o desvio padrão do ruído em cada encoder é de 10% do valor máximo medido.

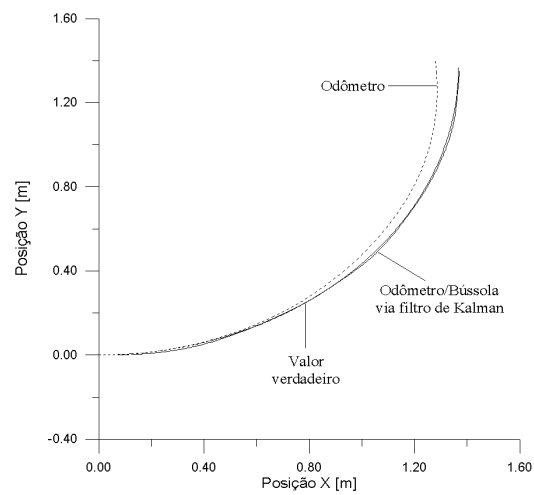


Figura 6: Simulação do odômetro e da fusão de dados odômetro/bússola.

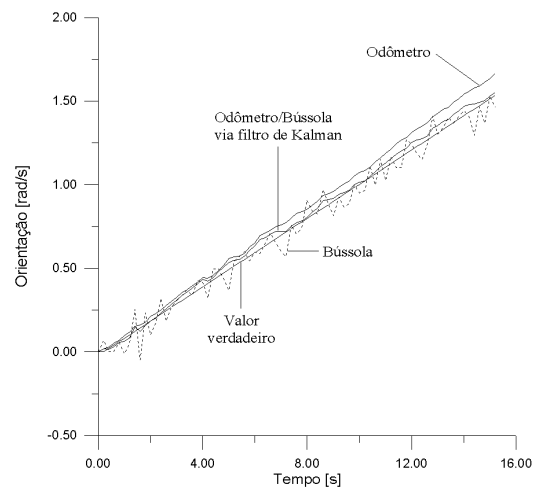


Figura 7: Simulação da orientação.

3.2 Resultados Experimentais

Navegou-se em uma trajetória circular, com raio mantido constante via elo rígido, de um ângulo de 90° , com os mesmos parâmetros utilizados na simulação. A trajetória nominal percorrida pelo veículo foi então calculada. A Fig. 8 mostra a trajetória nominal e as posições calculadas com base na integração odômetro/bússola e com o odômetro apenas.

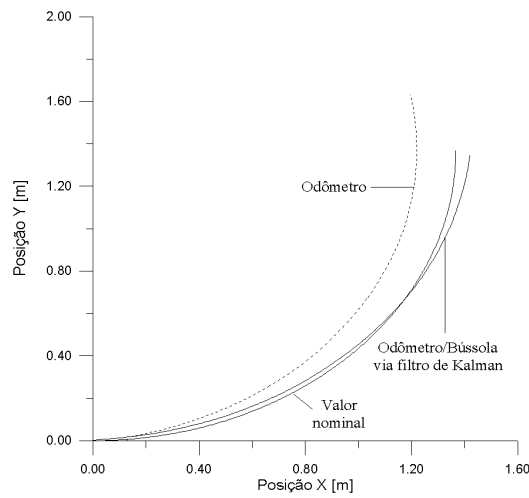


Figura 8: Posição calculada pelo odômetro e pela fusão odômetro/bússola em tempo real.

Note-se que a posição estimada oscila em torno da trajetória nominal crescendo com o tempo (devido à acumulação do erro de posição), mas sempre está mais perto do valor nominal do que o odômetro. A Fig. 9 mostra as várias orientações relativas a esta realização.

4 Conclusão

Foram apresentados neste capítulo métodos para estimação de posição e orientação para robôs móveis, utilizando fusão de dados, via filtro de Kalman. Foram apresentados resultados simulados da fusão de dados obtidos por *dead-reckoning* e dados obtidos por visão computacional. Foram também mostrados resultados simulados e obtidos experimentalmente da fusão de dados obtidos por *dead-reckoning* e dados de orientação obtidos por uma bússola digital.

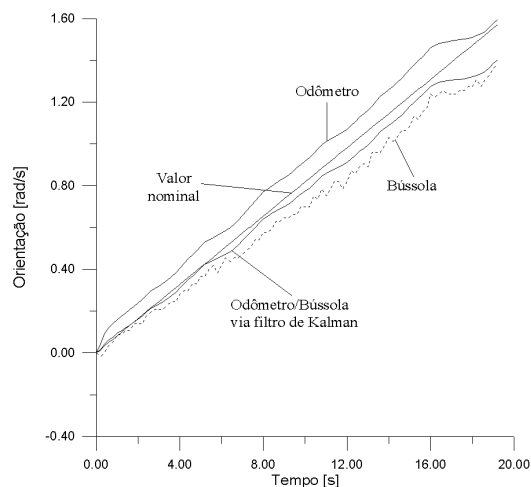


Figura 9: Orientações obtidas em tempo real.

Referências

- [1] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122, 1981.
- [2] J. Borenstein, H. R. Everett, and L. Feng. Where am i? - sensors and methods for mobile robot positioning. Technical report, University of Michigan, Apr 1996.
- [3] J. Borenstein and L. Feng. UMBMark - a method for measuring, comparing and correcting dead-reckoning errors in mobile robots. Technical Report UM-MEAM-94-22, University of Michigan, 1994.
- [4] Ö. Bozma and R. Kuc. A physical model-based analysis of heterogeneous environments using sonar - endura method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):497–506, May 1994.
- [5] E. Byler, C. Wendell, W. Hoff, and D. Layne. Autonomous hazardous waste drum inspection vehicle. *IEEE Robotics & Automation Magazine*, 2(1):6–17, Mar. 1995.
- [6] B. Barshan and H. Durrant-Whyte. Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342, Jun. 1995.
- [7] K. S. Fu, R. C. Gonzales, and C. S. G. Lee. *Robotics Control, Sensing, Vision and Intelligence*. Industrial Engineering Series. McGraw-Hill, New York, 1987.

- [8] G. C. Goodwin and K. S. Sin. *Adaptive Filtering, Prediction and Control*. Prentice-Hall Information and System Sciences Series. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.
- [9] E. M. Hemerly and C. C. Rodrigues. Guiagem de veículos autônomos utilizando sensor de visão. In *10^o Congresso Brasileiro de Automática*, Rio de Janeiro, 1994.
- [10] A. A. Holenstein. *Aufdatierung der Position und der Orientierung eines Mobilien Roboters*. Abhandlung zur erlangung des titels doktor der technischen wissenschaften, Eidgenössischen Technischen Hochschule, Zürich, 1992.
- [11] B. K. P. Horn. *Robot Vision*. The MIT Electrical Engineering and Computer Science Series. McGraw-Hill, New York, 1986.
- [12] A. Knoll. Sensordatenfusion für anwendungen in der robotik. Technical report, Universität Bielefeld, 1994.
- [13] R. Kuc and M. W. Siegel. Physically based simulation model for acoustic sensor robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):766–778, Nov. 1987.
- [14] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Number 175 in Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, june 1992.
- [15] S. Murata and T. Hirose. On board locating system using real-time image processing for a self-navigating vehicle. *IEEE Transactions on Industrial Electronics*, 40(1):145–153, Feb 1993.
- [16] S. Murata and T. Hirose. On board locating system using real-time image processing for a self-navigating vehicle. *IEEE Transactions on Industrial Electronics*, 40(1):145–153, Feb. 1993.
- [17] S. Negahdaripour, B. Y. Hayashi, and Y. Aloimonos. Direct motion stereo for passive navigation. *IEEE Transactions on Robotics and Automation*, 11(6):829–843, Dec. 1995.
- [18] S. Negahdaripour and B. K. P. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):168–176, Jan. 1987.
- [19] R. O. Nielsen. *Sonar Signal Processig*. Artech House, Norwood, 1991.

- [20] S. Perkins and G. Hayes. Real time optical flow based range sensing in mobile robots. Disponível on-line.
- [21] Precision Navigation, Inc. *Vector 2X Compass Module Application Notes Version 1.03*, Jan. 1996.
- [22] F. A. Sandi Lora, E. M. Hemerly, and W. F. Lages. Estimação em tempo real de posição e orientação de robôs móveis utilizando sensores com diferentes taxas de amostragem. In *Anais do III Simpósio Brasileiro de Automação Inteligente*, Vitória, ES, Brazil, 1998. Sociedade Brasileira de Automática.
- [23] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3):393–413, Mar. 1996.
- [24] R. Talluri and J. K. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. *IEEE Transactions on Robotics and Automation*, 8(5):575–584, Oct. 1992.
- [25] R. Talluri and J. K. Aggarwal. Image/map correspondence for mobile robot self-location using computer graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):597–601, June 1993.
- [26] S. J. Vestli. *Fast, Accurate and Robust Estimation of Mobile Robot Position and Orientation*. Dissertação de doutorado, ETH, Zurique, 1995.
- [27] J. Weng, P. Cohen, and N. Rebibo. Motion and structure estimation from stereo image sequences. *IEEE Transactions on Robotics and Automation*, 8(3):362–382, June 1992.
- [28] J. Werb and C. Lanzi. Designing a positioning systems for finding things and people indoors. *IEEE Spectrum*, 35(9):71–78, Sep. 1998.
- [29] Y. Yagi, S. Kawato, and S. Tsuji. Real-time omnidirectional image sensor (COPIS) for vision-guided navigation. *IEEE Transactions on Robotics and Automation*, 10(1):11–22, Feb. 1994.
- [30] A. Ylä-Jääki and N. Kiryati. Adaptive termination of voting in the probabilistic circular hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):911–915, Sep. 1994.
- [31] E. T. Baumgartner and S. B. Skaar. An autonomous vision-based mobile robot. *IEEE Transactions on Automatic Control*, 39(3):493–502, 1994.