

SOLUÇÃO FASTSLAM COM ATUALIZAÇÃO DE MARCOS UTILIZANDO MULTILAYER PERCEPTRON

FÁBIO SILVEIRA VIDAL[†], PAULO FERNANDO FERREIRA ROSA^{*}, ADÃO DE MELO NETO[‡], THIAGO
EUSTAQUIO ALVES DE OLIVEIRA^{*}

**Instituto Militar de Engenharia
Praça General Tibúrcio, 80 - Urca
Rio de Janeiro, RJ, Brasil*

*†Instituto Federal do Tocantins
BR-153 KM 480
Paraíso do Tocantins, TO, Brasil*

*‡Centro Tecnológico do Exército
Av. das Américas, 28750 - Barra de Guaratiba
Rio de Janeiro, RJ, Brasil*

Email: vidalafs@gmail.com, rpaulo@ime.eb.br, adao_de_melo_net@yahoo.com.br,
thiago.eustakio@gmail.com

Abstract— This paper presents a SLAM (Simultaneous Localization and Mapping) solution for multiple robots. Mapping and environment exploration tasks are coordinate by central agent. A common map is done with landmarks extracted by k robots. This reduces exploration time and increase mapping accuracy. For estimate robot pose, an algorithm based on FastSLAM is used, which uses EKF (Extended Kalman Filter) for landmarks update. EKF application performs matrix multiplications, matrix inversions, as well as need to compute Jacobians and covariance matrix. These operations have high computational cost. In proposed method, landmarks update done with a multilayer perceptron. Thus, processing time is 10-12 times lower and generated error is equivalent in both methods. This is important because SLAM is a real-time problem and computational resources are limited in mobile robot applications.

Keywords— SLAM, Robot Navigation, Cooperative Robotics, Artificial Neural Networks, Autonomous Robotics.

Resumo— Este artigo apresenta uma solução SLAM (*Simultaneous Localization and Mapping*) para múltiplos robôs. As tarefas de mapeamento e exploração do ambiente são coordenadas por um agente central. Um mapa comum é construído com marcos extraídos por k robôs. Isto reduz o tempo de exploração e aumenta a acurácia do mapeamento. Para estimar a pose do robô, é utilizado um algoritmo baseado no FastSLAM, o qual utiliza o EKF (*Extended Kalman Filter*) para atualização dos marcos. A aplicação do EKF executa multiplicações de matrizes, inversão de matrizes, Jacobianos e cálculo de matrizes de covariância. Tais operações possuem custo computacional elevado. No método proposto, os marcos são atualizados com o uso de uma *multilayer perceptron*. Assim, o tempo de processamento é de 10 a 12 vezes menor e o erro obtido é equivalente em ambos os métodos. Isto é importante pois o SLAM é um problema de tempo real e os recursos computacionais são limitados em aplicações de robótica móvel.

Keywords— SLAM, Navegação de Robôs, Robótica Cooperativa, Redes Neurais Artificiais, Robótica Autônoma.

1 Introdução

Determinar a localização de um robô e mapear o seu ambiente estão entre os principais desafios da robótica móvel. A complexidade destes problemas é maior sem o uso de sistemas externos de visão computacional e/ou sistemas de posicionamento. Sistemas embarcados possuem algumas limitações, como carga paga, disponibilidade de carga de baterias e poder de processamento reduzido. A localização e o mapeamento devem ser feitos em tempo hábil a prevenir colisões. Estes fatores aumentam a complexidade do contexto.

Localizar um robô e mapear o seu ambiente simultaneamente consistem em um problema chamado SLAM (*Simultaneous Localization and Mapping*), o qual pode ter a seguinte definição (Dissanayake et al., 2001): dado um

robô autônomo em um ambiente desconhecido e usando somente observações relativas a marcos detectáveis no ambiente em relação ao robô, construir um mapa de características para estes marcos e simultaneamente calcular uma estimativa da localização do robô baseada neste mapa.

Uma vez que um algoritmo SLAM garante a localização de um robô, uma grade de ocupação pode ser gerada simultaneamente para viabilizar a exploração de um ambiente (Makarenko et al., 2002), considerando que o espaço de trabalho tem um tamanho limitado. Uma implementação SLAM é feita para sistemas embarcados, tais como veículos aéreos não tripulados, carros autônomos e outros robôs móveis. Ações em tempo real são necessárias para estes sistemas. Portanto, os algoritmos devem ser computacionalmente eficientes. Neste trabalho, é utilizada

uma técnica SLAM conhecida como FastSLAM (Montemerlo and Thrun, 2003).

Várias soluções SLAM são baseadas na relação entre as medidas de sensores embarcados e comandos dados a seus atuadores. Se as medições dos sensores e os comandos executados pelos atuadores fossem corretos, o problema SLAM seria trivial. Contudo, a fidelidade das informações obtidas por estes sistemas podem sofrer interferências eletromagnéticas, bem como erros de odometria.

O uso de visão computacional é uma alternativa, pois é possível extrair marcos do ambiente e estimar a pose do robô. Para isso, é necessário estimar as coordenadas dos marcos a cada passo do robô a partir da sua pose anterior, posição anterior do marco e uma observação. Este processo é chamado de atualização de marcos. O mesmo é feito em alguns trabalhos com o uso do Filtro de Kalman Estendido (EKF), o qual é adequado para sistemas não-lineares. Neste trabalho, isto é feito com uso de uma *multilayer perceptron*.

O objetivo deste artigo é comparar o desempenho de uma solução FastSLAM em duas situações: (a) usando EKF para atualização dos marcos, e (b) usando uma rede neural artificial para esta tarefa. Dois aspectos são considerados: (a) tempo de processamento, e (b) erro obtido. A Seção 2 discute alguns trabalhos relacionados com SLAM. Na Seção 3, é explicado o processo de atualização de marcos. SLAM considerando dois robôs é tratado na Seção 4. A Seção 5 descreve os experimentos e resultados. As conclusões e trabalhos futuros são explanados na Seção 6.

2 Trabalhos Relacionados

Diversos estudos sobre mapeamento têm sido publicados. Algumas abordagens estimam a pose do robô usando somente odometria, o que frequentemente resulta em erros. Outros usam técnicas de SLAM, como EKF-SLAM (Dissanayake et al., 2001) e FastSLAM (Montemerlo and Thrun, 2003) com extração de características a partir de dados providos por sensores como laser ou sonar (Fulgenzi et al., 2009). Em Gil et al. (2006), um filtro de partículas Rao-Blackwellized (Montemerlo and Thrun, 2003) é aplicado para estimar simultaneamente o mapa e o caminho de um robô.

Abordagens SLAM com múltiplos robôs podem ser divididas em dois grupos. No primeiro, cada robô estima seu mapa individualmente usando suas observações e, posteriormente, um mapa global é formado pela fusão destes mapas individuais, como em Stewart et al. (2002) e Howard (2006). No segundo, a estimativa de trajetória e do mapa é feita conjuntamente. Um único mapa é calculado simultaneamente, usando as observações de todos os robôs, a exemplo de Stewart et al. (2002).

Alguns trabalhos propõem algoritmos para múltiplos robôs que constroem seus próprios mapas, como os de Howard (2006) e Fenwick et al. (2002). Os mapas são fundidos quando os robôs estão na mesma linha de visão. Em outros trabalhos, como os de Gil et al. (2010) e Fenwick et al. (2002), a estimação da trajetória e do mapa é feita conjuntamente.

Em Fenwick et al. (2002) é utilizado um EKF para estimar o vetor de estado formado pelas poses de todos os robôs e um conjunto de marcos 2D. Os robôs obtêm observações e constroem o mapa unificado usando equações de atualização do EKF clássico. A pose inicial dos robôs e a associação de dados devem ser conhecidas. Em Gil et al. (2010), é apresentado um algoritmo em que o mapa é comum para os robôs e supõe-se que os mesmos possuem pose inicial conhecida. Experimentos mostram que a abordagem proposta é adequada para um pequeno grupo de robôs. Sobre a estratégia de exploração, destaca-se as abordagens descritas em Burgard et al. (2005) que usa conceitos de custo e utilidade de células-fronteira.

Em de Melo Neto (2011a), de Melo Neto (2011b) e de Melo Neto (2012) é mostrado que no contexto de uma exploração integrada com múltiplos robôs, a abordagem FastSLAM com o mapa comum aumenta a exatidão da pose dos robôs. FastSLAM difere do EKF-SLAM pelas múltiplas hipóteses de associação de dados.

Este trabalho propõe um método FastSLAM em que a atualização dos marcos é feita usando uma *multilayer perceptron* ao invés de um EKF. O objetivo é reduzir o tempo de processamento da atualização dos marcos. Uma vez que esta abordagem não necessita de multiplicação de matrizes, inversão de matrizes, tampouco calcular matrizes jacobianas e de covariância. Após o treinamento da rede neural, quatro experimentos foram feitos: (a) utilizando os mesmos marcos nas mesmas posições do treinamento; (b) utilizando os mesmos marcos em posições invertidas; (c) utilizando outros marcos nas mesmas posições dos experimentos anteriores; e (d) utilizando diferente marcos em diferentes posições.

3 Atualização de Marcos com Redes Neurais Artificiais

A atualização de marcos consiste em estimar as suas coordenadas em um mapa. Em alguns trabalhos, como os de Montemerlo and Thrun (2003) e de Melo Neto (2012), tal processo é feito com uso do EKF. Contudo, este método requer operações computacionais como multiplicação de matrizes, inversão de matrizes, e o cálculo de matrizes jacobiana e de covariância.

A fim de reduzir o tempo de processamento da atualização dos marcos, foi utilizada outra abordagem: uma rede neural artificial. Tal abordagem

possui diversas aplicações, como o reconhecimento de padrões, visão computacional, previsão de séries temporais e aproximação de funções. Para implementação, foi usada a biblioteca *Fast Artificial Neural Network* (Nissen, 2003).

O neurônio artificial é um modelo matemático inspirado no neurônio biológico. O mesmo é composto por n entradas e uma saída. A saída é a soma ponderada das entradas, submetida a uma função de ativação. A mesma pode ser identidade, degrau, ou sigmoide.

A topologia usada no trabalho proposto é a *multilayer perceptron* com uma camada oculta. A saída dos neurônios de uma camada são a entrada dos neurônios da camada seguinte (vide Fig. 1). Cada conexão entre os neurônios possui um peso.

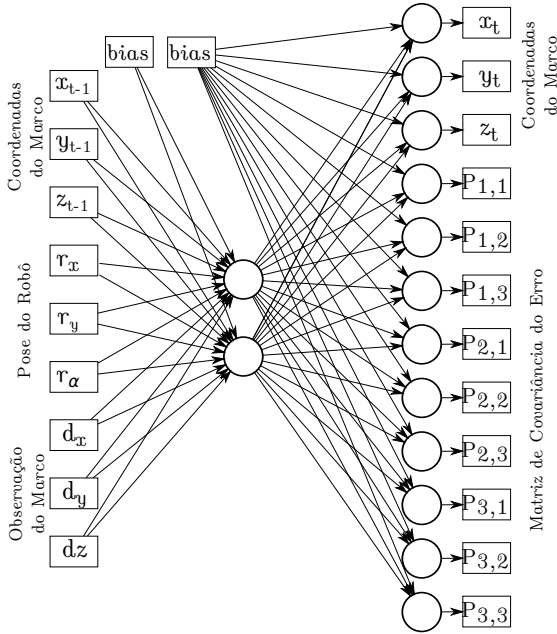


Figura 1: Arquitetura da rede neural utilizada para atualização de marcos.

Um processo de treinamento deve ser feito com a *multilayer perceptron*, antes de seu uso para solução de um problema. Este processo consiste em adaptar os pesos entre os neurônios de modo a minimizar os erros na saída. Faz-se isso para uma base de dados de treinamento e outra de validação. As mesmas são conjuntos reais de pares entrada e saída do problema em questão. De acordo com o erro obtido para os exemplos do conjunto de treinamento, os pesos são corrigidos seguindo um algoritmo específico. Tal algoritmo pode ter parâmetros como taxa de aprendizado, épocas e momento. Este processo é repetido até que se chegue a um erro desejado ou a um número determinado de iterações (épocas). Para que o método funcione, os erros para os conjuntos de treinamento e validação devem ser próximos.

Como pode ser observado na Fig. 1, a *multilayer perceptron*, utilizada no modelo proposto, tem as seguintes entradas: (a) x_{t-1} , y_{t-1} e z_{t-1} ,

coordenadas do marco no mapa no instante anterior; (b) r_x , r_y e r_α , pose do robô; e (c) d_x , d_y e d_z , observação do marco. A saída é compreendida por: (a) x_t , y_t e z_t , estimativa das coordenadas do marco no mapa; (b) $P_{i,i}$ é um elemento da matriz de covariância do erro das coordenadas do marco. As bases de dados para treinamento e validação foram coletadas durante a execução de uma solução FastSLAM, com atualizações feitas com o EKF. Nos casos em que um marco é detectado pela primeira vez, a sua posição corresponde à observação realizada.

O treinamento foi feito com o algoritmo *back-propagation*, executado em 1000 épocas e taxa de aprendizado 0,7 sem o uso de momento. A melhor configuração encontrada foi de dois neurônios na camada oculta e função de ativação identidade para todos os neurônios. O acréscimo de neurônios à camada oculta não melhorou o erro obtido. Assim, o método proposto obteve erro RMS de, aproximadamente, 0,0007 para o treinamento e o equivalente no processo de validação.

4 SLAM com Múltiplos Robôs

Quando múltiplos robôs se comunicam com um agente central, os mesmos podem, cooperativamente, observar os marcos do ambiente e reduzir o tempo de exploração.

Se k robôs exploram um ambiente, a partir dos marcos observados por cada um deles, é construído um mapa comum (Fig. 2 e Fig. 3). Para tanto, utiliza-se uma grade de ocupação. Em um instante t , o robô (i), na pose $\chi_{t,(i)}$, faz uma observação única $z_{t,(i)}$. A distribuição de probabilidade posterior do caminho $\chi_{(1:k)}$ dos k robôs e mapa Θ pode ser estimada seguindo a seguinte função (Gil et al., 2010):

$$p(\chi_{(1:k)}^t, \Theta | z_{(1:k)}^t, u_{(1:k)}^t, a^t) = \underbrace{p(\chi_{(1:k)}^t | z_{(1:k)}^t, u_{(1:k)}^t, a^t)}_{Path} \prod_{n=1}^N \underbrace{p(\theta_n | \chi_{(1:k)}^t, z_{(1:k)}^t, u_{(1:k)}^t, a^t)}_{Landmark}$$

em que $\chi_{(1:k)}^t = \{\chi_{(1)}^t, \dots, \chi_{(k)}^t\}$, $u_{(1:k)}^t = \{u_{(1)}^t, \dots, u_{(k)}^t\}$ e $z_{(1:k)}^t = \{z_{(1)}^t, \dots, z_{(k)}^t\}$ são, respectivamente, o conjunto de caminhos, ações e observações dos k robôs. $a^t = \{a_1, \dots, a_t\}$ é o histórico de associações de dados. A distribuição de probabilidade posterior de $\chi_{(1:k)}^t$ é estimada usando k filtros de partículas. Por outro lado, a distribuição de probabilidade posterior dos N marcos θ_i , correspondente a cada partícula, é estimada por kN EKF independentes, condicionados aos caminhos $\chi_{(1:k)}^t$. Uma vez que o mapa é comum aos robôs, k filtros de partículas produzem o mesmo conjunto S_t de partículas $S_t^{[m]} =$

$$\left[\chi_{(1:k),t}^{[m]}, \underbrace{\psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, d_1^{[m]}, \dots, \psi_{N,t}^{[m]}, \Sigma_{N,t}^{[m]}, d_N^{[m]}, w_t^{[m]}}_{\Theta^{[m]}} \right].$$

O estado a ser estimado é composto pela pose

$\chi_{t,(1:k)}$ de k robôs.

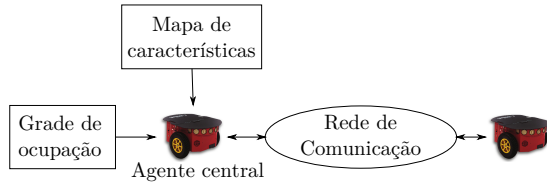


Figura 2: Esquema da exploração com dois robôs em que o agente central é o responsável pela construção do mapa e da grade de ocupação.

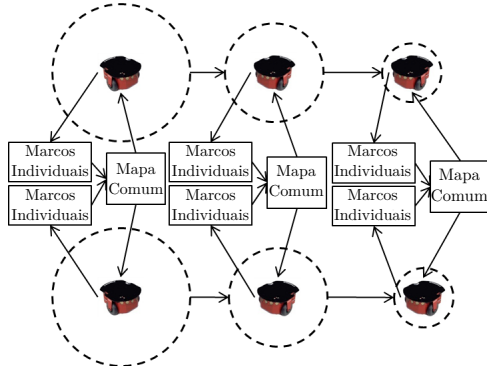


Figura 3: Esquema de SLAM com múltiplos robôs em que o mapa comum é construído a partir dos conjuntos individuais de marcos.

Considerando que os robôs partem de posições próximas, suas poses relativas podem ser obtidas por sensor laser. Como o mapa é comum aos robôs, para cada partícula $S_t^{[m]}$, k pesos são calculados e o peso total associado com a partícula $S_t^{[m]}$ é definido como $w_t = \prod_{i=1}^k w_{t,(i)}^{[m]}$.

Os resultados experimentais deste trabalho são mostrados na próxima seção.

5 Experimentos e Resultados

Considerando o uso de mapas comuns, foram realizados quatro experimentos no simulador Player/Stage (Fig. 4) para a plataforma Pioneer 3DX, sensor laser SICK LMS200 e representando cada marco visual 3D por uma imagem contendo apenas uma característica SIFT (Lowe, 1999). O computador utilizado possui a seguinte configuração: Intel Core 2 Duo (1.80GHz), e memória RAM DDR2 - 2GB - 667MHz.

A cada passo do robô, o sensor visual estereoscópico extrai imagens do ambiente e um detector de marcos gera observações. A *multilayer perceptron* estima as coordenadas atuais do marco, tendo como entradas: as coordenadas do marco no instante anterior, sua observação e a pose do robô. As coordenadas do marco são incluídas no conjunto de partículas. O filtro de partículas estima a pose atual do robô, a qual é incluída na

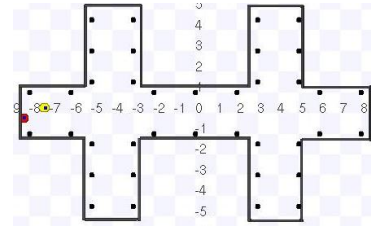


Figura 4: Ambiente explorado (17m x 10.5m) com os robôs em suas posições iniciais.

lista de poses do mesmo. O planejador de trajetória define os comandos, tendo como parâmetros as coordenadas do marco e dados do sensor laser. Os comandos executados pelo robô são incluídos em uma lista de comandos. Este processo é ilustrado na Fig. 5.

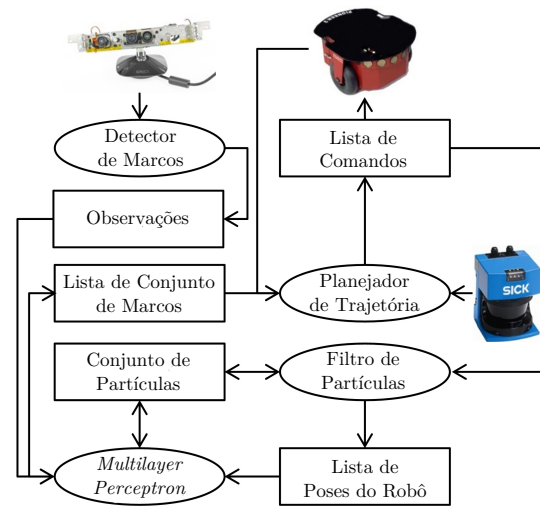


Figura 5: Esquema do método proposto para a solução SLAM com *multilayer perceptron* para atualização de marcos.

Um sensor laser SICK LMS200 embarcado em cada Pioneer 3DX foi utilizado para construção da grade de ocupação. Por outro lado, de modo a simular um sensor visual estereoscópico com um campo visual de 180° , utilizou-se marcos 3D representados por imagens.

A fim de comparação de resultados, também utilizou-se uma solução com EKF ao invés de uma rede neural, como no processo de treinamento descrito na Seção 3. As simulações foram feitas usando 1000, 2000, 4000, 6000, e 8000 partículas para o FastSLAM.

Em todos os experimentos, o tempo de processamento da atualização dos marcos foi menor que usando o EKF (vide Fig. 6). Uma vez que a *multilayer perceptron* não necessita fazer multiplicações de matrizes, inversões de matrizes, tampouco calcular matrizes jacobianas e matrizes de covariância.

O erro RMS obtido em de Melo Neto (2012), utilizando EKF, e neste trabalho foram similares.

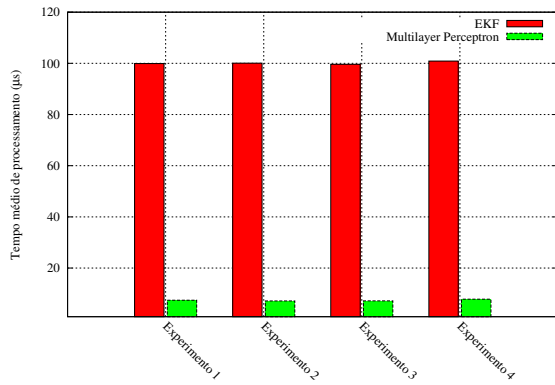


Figura 6: Tempo médio de processamento/atualização de marcos em todos experimentos, utilizando EKF e *multilayer perceptron*.

Como pode ser visto na Tabela 1, o erro obtido para x , y e α em de Melo Neto (2012) são 0,038, 0,041 e 0,073, respectivamente. Por outro lado, com o método proposto, o erro obtido para x , y e α estão nas faixas 0,022-0,07, 0,024-0,075 e 0,041-0,114, respectivamente. A seguir, os experimentos deste trabalho são descritos.

Tabela 1: Erro RMS usando *multilayer perceptron* neste trabalho e usando EKF em de Melo Neto (2012), ambos com 2000 partículas.

Caso	x	y	α
de Melo Neto (2012)	0.038	0.041	0.073
Experimento 1	0.070	0.026	0.114
Experimento 2	0.026	0.048	0.091
Experimento 3	0.022	0.075	0.077
Experimento 4	0.053	0.024	0.041

No experimento 1, foram utilizados os mesmos marcos nas mesmas posições do processo de treinamento (Seção 3). Como é percebido na Fig. 7a, o erro gerado é pequeno para ambos os métodos, e está concentrado na faixa entre 0,03 e 0,15.

No segundo experimento, utilizou-se os marcos do primeiro; porém, em posições invertidas. Como é percebido na Fig. 7b, o erro gerado é pequeno para ambos os métodos, e está concentrado na faixa entre 0,03 and 0,15.

No terceiro experimento, utilizou-se marcos diferentes do treinamento. Os mesmos possuíam apenas uma característica SIFT e suas coordenadas $[\theta_x, \theta_y, \theta_z]^T$ foram atribuídas artificialmente. Como é percebido na Fig. 7c, o erro gerado é pequeno para ambos os métodos, e está concentrado na faixa entre 0,03 and 0,2.

No quarto experimento, foram utilizados os marcos do terceiro experimento mas em outras posições. Como é percebido na Fig. 7d, o erro gerado é pequeno para ambos os métodos, e está concentrado na faixa entre 0,03 e 0,1.

6 Conclusões e Trabalhos Futuros

Este trabalho mostrou que é viável explorar um ambiente fechado com múltiplos robôs de uma forma eficiente ao utilizar o algoritmo FastSLAM para localizá-los, considerando marcos extraídos com o algoritmo SIFT e um mapa de características comum aos robôs. Alguns trabalhos implementam o FastSLAM utilizando EKF para a atualizar os marcos, este trabalho mostrou que usando *multilayer perceptron* para esta tarefa, consegue-se reduzir significativamente o tempo de processamento. Além disso, observou-se que o erro gerado é equivalente ao da outra metodologia.

Os experimentos mostraram que a *multilayer perceptron* teve desempenho equivalente quando os marcos e/ou suas posições foram trocados no processo de validação. Além disso, foi notado que a partir de 1000 partículas, um acréscimo no número de partículas não influencia a margem de erro.

A cada passo da exploração do ambiente, são realizadas várias atualizações de marcos. Estas operações podem ser feitas de modo paralelo. Assim, como trabalho futuro, propõe-se implementar o método apresentado utilizando computação paralela em Unidade de Processamento Gráfico, *Graphics Processing Unit* (GPU), com a plataforma CUDA (*Compute Unified Device Architecture*). Além de fazer os experimentos em ambiente real, usando o mesmo *hardware* das simulações.

7 Agradecimentos

Este trabalho foi amparado pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) sob a concessão nº 484767/2011-9.

Referências

- Burgard, W., Moors, M., Stachniss, C. and Schneider, F. (2005). Coordinated multi-robot exploration, *Robotics, IEEE Transactions on* **21**(3): 376–386.
- de Melo Neto, A. e Rosa, P. d. O. T. e. P. P. (2011a). A slam approach for multiple homogeneous autonomous vehicles, *3rd Int. Congress on Ultra Modern Telecommunications and Control Systems*. **1**: 1–6.
- de Melo Neto, A. e Rosa, P. e. d. O. T. e. P. P. (2011b). Environment exploration with multiple vehicles and fastslam technique, *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 122–128.
- de Melo Neto, A. e Rosa, P. e. d. O. T. e. P. P. (2012). Multiple robots in a cooperating task: Exploration an mapping, *IROS12 4th*

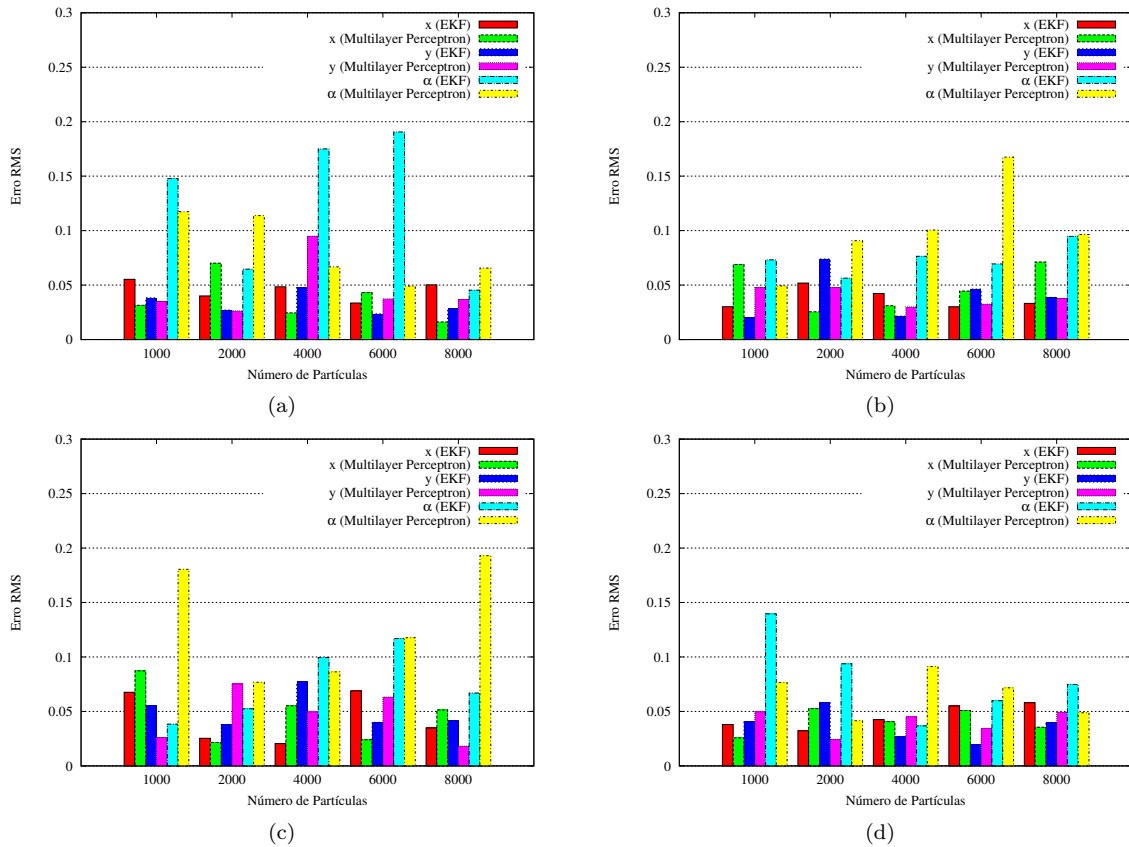


Figura 7: (a), (b), (c) e (d) mostram o Erro RMS da estimaco da pose do rob (x , y e α), gerado com *multilayer perceptron* e EKF para atualizaco de marcos usando 1000, 2000, 4000 e 8000 partulas, nos experimentos 1, 2, 3 e 4, respectivamente.

International Workshop on Planning, Perception and Navigation for Intelligent Vehicles. pp. 79–86.

Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem, *Robotics and Automation, IEEE Transactions on* **17**(3): 229–241.

Fenwick, J., Newman, P. and Leonard, J. (2002). Cooperative concurrent mapping and localization, *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, Vol. 2, pp. 1810–1817 vol.2.

Fulgenzi, C., Ippoliti, G. and Longhi, S. (2009). Experimental validation of fastslam algorithm integrated with a linear features based map, *Mechatronics* **19**(5): 609 – 616.

Gil, A., Reinoso, Ó., Ballesta, M. and Juliá, M. (2010). Multi-robot visual slam using a rao-blackwellized particle filter, *Robotics and Autonomous Systems* **58**(1): 68–80.

Gil, A., Reinoso, O., Mozos, O., Stachniss, C. and Burgard, W. (2006). Improving data association in vision-based slam, *Intelligent*

Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 2076–2081.

Howard, A. (2006). Multi-robot simultaneous localization and mapping using particle filters, *International Journal of Robotics Research* **25**(12): 1243–1256.

Lowe, D. (1999). Object recognition from local scale-invariant features, *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, Vol. 2, pp. 1150–1157 vol.2.

Makarenko, A., Williams, S., Bourgault, F. and Durrant-Whyte, H. (2002). An experiment in integrated exploration, *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, Vol. 1, pp. 534–539 vol.1.

Montemerlo, M. and Thrun, S. (2003). Simultaneous localization and mapping with unknown data association using fastslam, *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, Vol. 2, pp. 1985–1991 vol.2.

Nissen, S. (2003). Implementation of a fast artificial neural network library (fann), *Technical*

report, Department Computer Science University of Copenhagen.

Stewart, B., Ko, J., Fox, D. et al. (2002). The revisiting problem in mobile robot map building: A hierarchical bayesian approach, *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., pp. 551–558.