

UMA NOVA ARQUITETURA DE CONTROLE PARA NAVEGAÇÃO DE ROBÔS MÓVEIS: FUSÃO DAS SAÍDAS DE DISTINTOS CONTROLADORES

EDUARDO O. FREIRE^{1,2}, TEODIANO F. BASTOS-FILHO², MÁRIO SARCINELLI-FILHO², RICARDO CARELLI³, OSCAR NASISI³

¹*Grupo de Pesquisa em Automação do Instituto de Tecnologia e Pesquisa da Universidade Tiradentes
Av. Murilo Dantas, 300, Farolândia, 49032-490, Aracaju/SE, Fax: (79) 218-2200*

²*Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo
29060-900, Vitória/ES, Fax: (27) 335-2737*

³*Instituto de Automática, Universidad Nacional de San Juan
Av. San Martín Oeste, 1112, 5400, San Juan, Argentina, Fax: +54-264-421-3672*

E-mail: freire@ele.ufes.br

Resumo: Este artigo apresenta uma nova arquitetura de controle para navegação de robôs móveis que se baseia na fusão das saídas de distintos controladores. Esta arquitetura já apresentou até o momento resultados muito bons, tanto em simulações quanto em experimentos práticos. A fusão das saídas dos vários controladores é obtida através do filtro de informação descentralizado (DIF). A saída de cada controlador está conectada a um dos filtros locais do DIF, e tem associada a ela uma covariância. Quanto menor a covariância do filtro local ao qual está conectado um controlador, maior é a influência da saída dele na saída fusionada. Para a determinação das covariâncias do DIF utiliza-se lógica nebulosa.

Abstract: This paper presents a new architecture for controlling the navigation of a mobile robot based on the fusion of the output of several controllers. This architecture has already presented very good results, as demonstrated by simulations and practical experiments. The fusion of the outputs of the several controllers is realized by the decentralized information filter (DIF). The output of each controller is connected to one of the local filters of the DIF, and has a covariance assigned to it. The lower the covariance assigned to the local filter connected to a certain controller is, the bigger is its influence on the fused output. The covariance assigned to each local filter of the DIF is calculated using fuzzy logic.

Keywords: Mobile Robot; Control Architectures; Fusion of Control Signals; Decentralized Information Filter.

1. Introdução

Uma arquitetura para o controle de robôs móveis deve ser capaz de unir tanto o planejamento deliberativo como o controle reativo. Se estas duas abordagens forem adequadamente reunidas, é possível que uma compense as deficiências da outra (Rosenblatt, 1997 e Arkin e Balch, 1997).

Para obter esta união, normalmente se utilizam arquiteturas híbridas, que possuem um sistema reativo para executar as tarefas de baixo nível (como evitar obstáculos por exemplo), e um sistema deliberativo, que define tarefas de nível mais alto (Coste-Manière et al, 1995).

Neste trabalho é proposta uma arquitetura de controle que consiste na fusão das saídas de distintos controladores através de um filtro de informação descentralizado. Ela é suficientemente flexível para permitir a união das duas abordagens anteriormente mencionadas (dependendo dos controladores implementados), apesar de ser uma arquitetura de controle baseada em comportamentos (e não uma arquitetura híbrida).

Na seqüência do artigo, a seção 2 aborda outros trabalhos que têm relação com a arquitetura aqui proposta. A seção 3 mostra os detalhes da arquitetura

de controle proposta, enquanto a seção 4 mostra como são calculadas as covariâncias do filtro de informação descentralizado utilizando-se lógica nebulosa. A seção 5 apresenta dois experimentos realizados, a fim de demonstrar o funcionamento da arquitetura proposta. Na seção 6 são apresentadas as conclusões deste trabalho e são delineados alguns trabalhos futuros. Na seção 7 estão os agradecimentos e, finalmente, na seção 8 as referências bibliográficas.

2. Trabalhos Correlatos

Existem outros trabalhos que propõem algum tipo de fusão das saídas de distintos controladores ou comportamentos. Em seguida, uma breve análise de três deles é realizada.

2.1. Arquitetura AuRA (Arkin e Balch, 1997)

A arquitetura AuRA (*Autonomous Robot Architecture*), desenvolvida em meados da década de 80, é uma arquitetura híbrida, composta de um sistema hierárquico responsável pelo planejamento deliberativo e um sistema reativo, conhecido como *Controlador de Esquemas*.

O controlador de esquemas é um sistema reativo com um tipo de controle diferente dos seus antecessores, como a arquitetura *subsumption* de Brooks (Brooks, 1986 e Mataric, 1992), pelo fato de que não é necessário arbitrar entre os comportamentos e não existem camadas de comportamentos. Para cada contexto do ambiente de operação, o sistema deliberativo define quais dos comportamentos disponíveis devem ser utilizados para a execução da tarefa desejada.

Os esquemas são codificados utilizando-se uma analogia com o método dos campos potenciais (Latombe, 1991). Porém, os esquemas são muito mais simples do ponto de vista computacional do que o método clássico dos campos potenciais, pois o campo potencial só é calculado na posição do robô, e não em todos os pontos do espaço (Arkin, 1998). Isto permite que os esquemas sejam executados em tempo real, que é um requisito básico para um sistema reativo. Os métodos baseados em campos potenciais são conhecidos por apresentar problemas com mínimos locais e problemas cíclicos (Rosenblatt, 1997, Arkin e Balch, 1997 e Arkin, 1998). Esta é uma das razões para incorporar o planejamento deliberativo e o controle reativo na arquitetura AuRA.

A cada esquema motor está associado um esquema sensor. A percepção é conduzida no sentido de *saber o que é necessário*: cada esquema sensor provê a informação necessária para cada comportamento reagir. As respostas de cada esquema motor aos estímulos recebidos acontecem na forma de vetores. Os esquemas motores podem operar assincronamente, produzindo respostas no menor tempo possível. Para solucionar cada tarefa, alguns comportamentos são agrupados de maneira adequada. A importância relativa de cada comportamento é codificada através de um ganho. Os vetores produzidos como saída de cada comportamento (ou esquema) são multiplicados pelos ganhos associados e são então adicionados vetorialmente para obter a reação do robô ao ambiente de operação. Os ganhos podem ser alterados por um sistema de planejamento ou por um operador humano.

Em resumo, AuRA é uma arquitetura assíncrona que apresenta codificação contínua de respostas, utilizando uma analogia com o método dos campos potenciais. Nesta arquitetura, o método de coordenação entre os diversos comportamentos é cooperativo, através de soma vetorial ponderada (Arkin, 1998).

2.2. Arquitetura DAMN (Rosenblatt, 1997)

A arquitetura DAMN (*Distributed Architecture for Mobile Navigation*) é uma arquitetura reativa onde vários comportamentos, operando de forma independente e assíncrona, determinam de forma cooperativa a ação do robô.

Esta arquitetura consiste em um grupo de comportamentos distribuídos e assíncronos que se comunicam com um árbitro centralizado através do envio de votos a favor daquelas ações que satisfaçam seus objetivos e contra as que não satisfaçam, ou indicando ao árbitro a utilidade de possíveis estados do “mundo”. Um árbitro, então, é responsável por executar a fusão dos votos dos comportamentos e gerar as ações que melhor reflitam os objetivos prioritários do sistema. A cada comportamento está associado um peso, o que permite estabelecer um esquema de prioridade entre os comportamentos. Podem existir múltiplos árbitros operando paralelamente para o controle de velocidade linear e angular, o que é uma característica única da arquitetura DAMN (Arkin, 1998). O uso de controle compartilhado e distribuído permite que múltiplos níveis de planejamento sejam utilizados para a tomada de decisões sem a necessidade de uma estrutura hierárquica. Além disso, a natureza distribuída e assíncrona da arquitetura permite que diversos objetivos e limitações sejam levados em consideração ao mesmo tempo. Assim, a arquitetura DAMN provê um comportamento orientado a objetivo que é coerente e racional, ao mesmo tempo que preserva a capacidade de resposta em tempo real ao ambiente de operação.

DAMN é uma arquitetura assíncrona que apresenta codificação discreta de resposta, através de conjuntos de votos, e um método de coordenação entre os comportamentos que consiste em múltiplos árbitros independentes do tipo *winner-take-all* (Arkin, 1998).

2.3. Abordagem Dinâmica (Bicho, 1999)

A abordagem dinâmica fornece vários conceitos e princípios que são baseados na teoria matemática dos sistemas dinâmicos e em elementos de redes neurais. Estes conceitos podem ser utilizados como uma linguagem teórica que permite o desenvolvimento de arquiteturas de controle para a navegação de robôs móveis. As principais idéias desta linguagem são: (a) O conceito de *variáveis comportamentais*, que consistem em variáveis que podem descrever, parametrizar e representar internamente um certo comportamento (estado do sistema). Estas variáveis definem as dimensões comportamentais ao longo das quais cada comportamento pode variar. Elas devem ser escolhidas de modo que a cada instante um comportamento deve estar associado com valores particulares de suas variáveis comportamentais e os requerimentos de uma tarefa sejam expressos como valores ou conjuntos de valores destas variáveis. (b) O conceito de *dinâmica comportamental*, de acordo com o qual os comportamentos são gerados como soluções atratoras de sistemas dinâmicos. (c) A dinâmica de campos neurais estende os princípios anteriores ao conceito de representação neural da informação.

Os cursos de tempo das variáveis comportamentais são obtidos como soluções de

atratores para sistemas dinâmicos (dinâmica comportamental), formuladas para expressar os requisitos do sistema, como forças atrativas ou repulsivas.

Através da escolha das variáveis e ajuste nas escalas de tempo, o sistema deve ser sintonizado para estar sempre em um atrator, ou ao menos perto de um. Informações sensoriais ou informações de outros módulos comportamentais (sistemas dinâmicos ou outras variáveis comportamentais) determinam a localização, força e alcance das contribuições atrativas e repulsivas da dinâmica comportamental.

A interação entre múltiplas contribuições para a dinâmica comportamental, como fontes de informação sensorial, podem cooperar ou competir, o que pode levar a uma mudança de comportamento, ou simplesmente a um ajuste comportamental.

A abordagem dinâmica aplicada ao controle de navegação de robôs móveis, é uma arquitetura síncrona, que apresenta codificação contínua de resposta, através de campos vetoriais, e o método de coordenação entre os comportamentos consiste na soma dos campos vetoriais produzidos por cada comportamento.

2.4. Multivaluated Logic Approach (Saffiotti et al, 1995)

Esta arquitetura é composta de 3 camadas: esquemas de controle (*control schemas*), esquemas comportamento (*behavior schemas*) e planejadores, que interagem com o robô assim como com o ambiente em distintos níveis de abstração: sinal/estímulo, intermediário (símbolo-para-sinal) e simbólico respectivamente.

Esquemas de controle descrevem tipos de movimentos baseados no estado interno do robô e nos dados fornecidos pelos sensores. Os planejadores, no outro extremo, sintetizam planos que são baseados em descrições simbólicas. Os esquemas comportamentos preenchem o espaço entre os planejadores e os esquemas controladores, contendo partes tanto do nível de abstração simbólico quanto de estímulo.

Os esquemas de controle são definidos como um mapeamento de um conjunto S de estados para preferências em um conjunto A de ações a executar. Formalmente, o mapeamento $D: S \times A \rightarrow [0,1]$, mede o grau de desejo $D(s,a)$ em executar a ação a na situação s . Na prática, cada esquema de controle é implementado por um conjunto de regras nebulosas do tipo IF-THEN. As saídas das regras são então combinadas em uma função de pertinência que reflete o grau de desejo D do esquema de controle implementado. A função de pertinência resultante é *defuzzificada* para calcular uma única saída de controle. Vários esquemas de controle podem ser compostos utilizando-se operadores da lógica nebulosa, como OR, AND, etc.

Enquanto o processo de *defuzzificação* pode produzir resultados satisfatórios na maioria dos

casos, em outros casos no entanto, onde existe conflito entre dois esquemas de controle, pode ser que produza resultados não apropriados para ambos. Para contornar este problema, Saffiotti propôs associar a cada esquema de controle um contexto, que define as condições sob as quais cada controlador é aplicável. As condições contextuais são implementadas por um conjunto regras que ativam os esquemas de controle, do tipo:

IF *contexto*= A THEN *ative_esquema_de_controle* C
onde A é um termo nebuloso representando o contexto real. Esta regra nebulosa ativa o esquema de controle C a um nível determinado pelo grau de verdade do antecedente da regra. O nível de ativação é utilizado para ponderar a função de grau de desejo do esquema de controle.

Esquemas comportamento são estruturas que conectam certos padrões de ações, implementados por esquemas de controle, com certos estímulos provenientes do ambiente.

Dessa forma, comportamentos estímulo-resposta podem ser formulados para atingirem objetivos específicos. Um comportamento é descrito como uma tripla: $B = (C,D,O)$, onde D é um esquema de controle para um tipo específico de movimento, O é um conjunto de descritores de um objeto com respeito ao qual o movimento deve ser executado, e C é uma condição de contexto, que define o contexto de aplicabilidade do movimento. Descritores do objeto são modelos de objetos no mundo real e eles fazem que o comportamento atue com respeito ao mundo externo, em oposição a um esquema de controle, que atua de acordo com variáveis internas. Os esquemas comportamento também podem ser compostos em esquemas comportamento mais complexos utilizando a mesma técnica de composição utilizada pelos esquemas de controle.

Planejadores (baseados na teoria clássica de inteligência artificial) podem utilizar especificações dos comportamentos, em termos de pré-condições e pós-condições, para construir planos que permitam atingir objetivos específicos.

Esta é portanto uma arquitetura síncrona, que apresenta uma codificação de resposta discreta e um método de coordenação entre os comportamentos denominado *context-dependent blending*.

2.5. Multiple Objective Decision Making Control (Pirjanian, 2000)

Nesta arquitetura de controle, cada comportamento calcula uma função objetivo para o conjunto de ações de controle admissíveis. A ação que maximiza a função objetivo corresponde à ação que melhor satisfaz o objetivo do comportamento.

Os múltiplos comportamentos são misturados em um único comportamento mais complexo que busca selecionar a ação que simultaneamente satisfaz aos objetivos de cada comportamento da melhor maneira possível, o que consiste em um método de otimização vetorial.

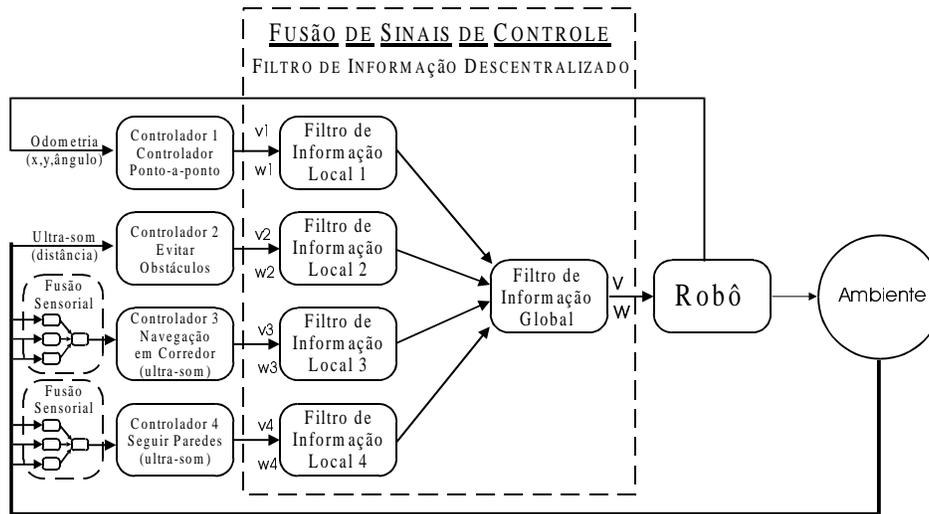


Fig. 1: Fusão das saídas de distintos controladores.

Trata-se portanto de uma arquitetura síncrona que apresenta uma codificação de resposta discreta e um método de coordenação entre os comportamentos baseado na teoria conhecida como *Multiple Objective Decision Making* (Chankong and Haimes, 1983).

3. A Arquitetura de Controle Proposta

A Fig. 1 apresenta um exemplo de implementação da arquitetura proposta neste trabalho. Os controladores utilizados podem ser diferentes, e o número deles pode ser maior ou menor, desde que apresentem como variáveis de saída velocidades lineares e angulares. Como mostra a Fig. 1, cada controlador tem como entrada informações sensoriais (que podem ser fusionadas para uma melhor representação do ambiente de operação e reduzir o ruído associado às medições) e apresentam como saída velocidades lineares ou angulares (ou ambas), que são as entradas dos filtro de informação locais. Estes, juntamente com o filtro de informação global, compõem o filtro de informação descentralizado (DIF) (Freire, 2001).

A cada filtro local está associada uma covariância que é uma medição da confiabilidade dos dados que estão sendo observados por este filtro local. A covariância de cada filtro pode ser calculada utilizando-se lógica nebulosa, tendo como base as informações disponibilizadas pelos sensores (seção 4). O valor da saída do filtro de informação global se aproxima mais do valor da saída do filtro de informação local que tem associado a ele a menor covariância (indicando que é ele que tem informações mais confiáveis). Desse modo, o sistema combina informações de velocidade angular e linear de vários controladores usando o filtro de informação descentralizado, que, assim como o filtro de informação e o filtro de Kalman, é um método de fusão otimizado (Mutambara 1998). A técnica de fusão proposta neste trabalho, diferentemente das que são usadas nas arquiteturas analisadas na seção 2, é uma técnica de fusão otimizada e realiza a fusão diretamente nos valores das saídas de velocidade

linear e angular que normalmente são produzidas pelos controladores, dispensando qualquer tipo de pré-processamento antes da fusão. Ela apresenta codificação contínua de resposta, através das componentes de velocidade linear e angular, sendo que o método de coordenação entre os controladores consiste na fusão das suas saídas através do DIF.

4. Uso de Lógica Nebulosa para Determinação das Covariâncias do DIF

As covariâncias associadas aos vários controladores envolvidos no processo de fusão podem ser definidas de diferentes maneiras. A primeira é através de equações matemáticas, que, usando como variáveis os dados medidos pelos sistemas sensoriais do robô, tentam medir de alguma maneira o grau de adequação de cada controlador a cada momento. Quanto mais adequado for um controlador, menor deve ser o valor da equação que define a sua covariância. O segundo modo de determinar as covariâncias é usando a lógica nebulosa. Neste caso é possível usar variáveis linguísticas e modelar o conhecimento que o projetista tem sobre o sistema de navegação do robô e seu ambiente de operação, através de uma base de regras, o que é mais simples do que encontrar uma equação matemática que defina a covariância de uma maneira adequada.

Para determinar as covariâncias através de lógica nebulosa são utilizadas duas variáveis nebulosas (antecedentes): o produto das distâncias medidas pelos sensores de ultra-som à direita e à esquerda do robô, e a menor distância medida pelos sensores de ultra-som frontais.

No caso em que a distância mínima medida pelos sensores de ultra-som frontais é “pequena” ou “muito pequena”, o risco de colisão é grande ou muito grande respectivamente. Assim, o algoritmo responsável por evitar obstáculos deve ter associado a ele uma covariância muito pequena, enquanto que todos os outros algoritmos de controle devem ter

associadas a eles covariâncias maiores. No caso em que a distância mínima medida pelos sensores de ultra-som frontais é “grande” ou “muito grande”, a covariância associada ao controlador responsável por evitar obstáculos deve ter um valor alto. Se além disso o produto das distâncias medidas à direita e à esquerda do robô através dos sensores de ultra-som é “pequena” ou “muito pequena”, significa que o robô está em um corredor, logo o algoritmo de navegação em corredores deve ter associada a ele uma covariância pequena, enquanto que os outros controladores devem ter associadas a eles covariâncias maiores. No caso em que a menor distância medida pelos sensores de ultra-som frontais e o produto das distâncias medidas à direita e à esquerda do robô sejam “grandes” ou “muito grandes”, o controlador ponto-a-ponto (responsável por atingir o ponto de destino) é que estará associado a uma covariância pequena, enquanto os outros estarão associados a covariâncias maiores.

Os conjuntos nebulosos e as funções de pertinência das duas variáveis de entrada (anteriores) são idênticos e estão mostrados na Fig. 2. Os conseqüentes, que são as covariâncias associadas a cada controlador (R1: controlador de navegação em corredores, R2: controlador ponto-a-ponto e R3: evitar obstáculos), são modelados como *singletons* nebulosos, ou seja, como números reais (Babuska, 1998). A *defuzzificação* é realizada usando o método *fuzzy mean*, ou seja, de acordo com

$$y = \frac{\sum_{i=1}^k \beta_i b_i}{\sum_{i=1}^k \beta_i} \quad (1)$$

onde y é o valor que será atribuído à covariância, b_i são os valores *singletons* e β_i é o grau de pertinência do antecedente.

As Tabelas 1 e 2 mostram as regras nebulosas empregadas, onde MP significa muito pequeno, P significa pequeno, G significa grande e MG significa muito grande.

Tabela 1: Regras nebulosas usadas para determinar a covariância do controlador de seguir corredores

$d_1 \times d_2$	MP	P	M	G	MG
MP	R ₁ =MG				
P	R ₁ =P	R ₁ =M	R ₁ =G	R ₁ =MG	R ₁ =MG
M	R ₁ =MP	R ₁ =MP	R ₁ =MP	R ₁ =MG	R ₁ =MG
G	R ₁ =MP	R ₁ =MP	R ₁ =MP	R ₁ =MG	R ₁ =MG
MG	R ₁ =MP	R ₁ =MP	R ₁ =MP	R ₁ =MG	R ₁ =MG

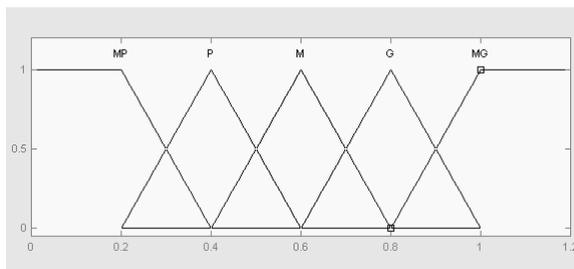


Fig. 2: Funções de Pertinência das Variáveis de Entrada.

Tabela 1: Regras nebulosas usadas para determinar a covariância do controlador ponto-a-ponto, evitar obstáculos e seguir paredes

Min(d_1, d_2)	MP	P	M	G	MG
MP	R ₂ =MG R ₃ =MP R ₄ =MG				
P	R ₂ =MG R ₃ =MP R ₄ =P	R ₂ =MG R ₃ =MP R ₄ =M	R ₂ =MG R ₃ =MP R ₄ =G	R ₂ =MG R ₃ =MP R ₄ =MG	R ₂ =MG R ₃ =MP R ₄ =MG
M	R ₂ =MG R ₃ =MG R ₄ =MP	R ₂ =MG R ₃ =MG R ₄ =MP	R ₂ =MG R ₃ =MG R ₄ =MP	R ₂ =MG R ₃ =MG R ₄ =MG	R ₂ =MG R ₃ =MG R ₄ =MG
G	R ₂ =MG R ₃ =MG R ₄ =MP	R ₂ =MG R ₃ =MG R ₄ =MP	R ₂ =G R ₃ =MG R ₄ =MP	R ₂ =P R ₃ =MG R ₄ =MG	R ₂ =MP R ₃ =MG R ₄ =MG
MG	R ₂ =MG R ₃ =MG R ₄ =MP	R ₂ =MG R ₃ =MG R ₄ =MP	R ₂ =G R ₃ =MG R ₄ =MP	R ₂ =P R ₃ =MG R ₄ =MG	R ₂ =MP R ₃ =MG R ₄ =MG

Tabela 3: Índices de Medição de Desempenho

Índice	Valor Obtido	Valor Ideal
Segurança	350mm	∞
Erro de orientação	33°	0
Velocidade média	260mm/s	300mm/s
Suavidade	0,79°	0
Tempo	57,9s	56,7s

5. Resultados Obtidos

Para demonstrar o desempenho da técnica de fusão das saídas de distintos controladores foi realizado repetidas vezes um experimento padrão, que consiste em fazer o robô ir de um ponto de origem (localizado nas coordenadas [0,0]) a um ponto de destino (localizado nas coordenadas [12m,5m]), enquanto navega por alguns corredores e evita colidir com obstáculos presentes em seu caminho. A Fig. 3 ilustra o caminho percorrido pelo robô e demonstra como o robô é capaz de executar a tarefa proposta

Os experimentos foram realizados usando um robô PIONEER 2DX, com 16 sensores de ultra-som e uma câmara CCD (que não foi utilizada nestes experimentos). O robô tem um microcontrolador Siemens 88C166 de 20 MHz para o controle de baixo nível das rodas e dos sistemas de sensoriamento. A navegação é controlada, através de um sistema de rádio, por um computador remoto que está executando a arquitetura proposta na Fig. 1.

A Tabela 3 mostra os valores obtidos para os índices de desempenho da arquitetura proposta e o que seriam os valores ideais. O índice *segurança*, indica a média das mínimas distâncias medidas pelos sensores de ultra-som ao longo do percurso. O índice *erro de orientação* indica a diferença média ao longo do percurso entre a orientação do robô e a direção do ponto destino. O índice *velocidade média*, indica a média da velocidade linear do robô ao longo do percurso e finalmente, o índice *suavidade*, é medido calculando a média do valor absoluto da subtração do valor atual da orientação do robô, pelo valor da orientação no instante imediatamente anterior.

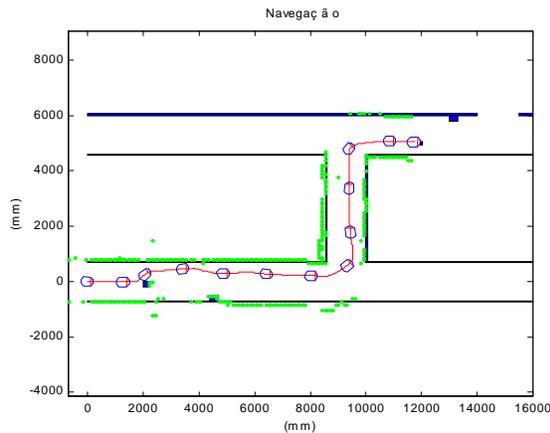


Fig. 3: Navegação do robô Pioneer 2DX utilizando a técnica de navegação proposta neste trabalho.

6. Conclusões

Neste trabalho foi proposta uma nova arquitetura de controle que consiste na fusão das saídas de controladores distintos, utilizando-se o filtro de informação descentralizado, que é um método de fusão otimizado. Também foi mostrado como realizar o cálculo das covariâncias utilizando-se lógica nebulosa, e destacadas as vantagens que isso acarreta.

Foi realizada, também, uma análise de trabalhos correlatos e efetuada uma comparação entre eles e a arquitetura proposta neste trabalho. Estas outras arquiteturas também realizam fusão das saídas de distintos comportamentos. No entanto, os métodos de fusão que empregam não são métodos otimizados. A arquitetura proposta também apresenta como vantagem o fato de realizar diretamente a fusão das variáveis de saída dos controladores (velocidade linear e angular).

Como trabalhos futuros, se propõe o desenvolvimento de novos controladores e a realização de estudos no sentido de demonstrar a estabilidade da arquitetura proposta.

7. Agradecimentos

Este trabalho foi desenvolvido parcialmente com o apoio da CAPES e do ITP-UNIT, através do PAPI (Programa de Apoio à Pesquisa Institucional). O apoio da CAPES se deu através de um projeto de intercâmbio entre a Universidade Federal do Espírito Santo e a Universidad Nacional de San Juan, de San Juan, Argentina. O projeto também foi parcialmente apoiado pelo ANPCyT e CONICET (Argentina). Dentro deste projeto, Eduardo Oliveira Freire esteve por três meses em San Juan, onde desenvolveu os experimentos aqui relatados.

8. Referências Bibliográficas

- Arkin, R. e T. Balch, "AuRA: Principles and Practice in Review", *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2), pp. 175-189 (1997).
- Arkin, R. C., *Behavior-Based Robotics*, The MIT Press, (1998).
- Babuska, R., "Fuzzy Modeling for Control", Kluwer Academic Publishers, USA, (1998).
- Bicho, E. "The Dinamic Approach to Behavior-Based Robotics", PhD. Thesis, University of Minho, Portugal, (1999).
- Brooks, R. A., "Achieving artificial intelligence through building robots", MIT A. I. Memo 899, Massachusetts, USA (1986).
- Brown, R. y P. Hwang, "Introduction to Randon Signals and Applied Kalman Filtering", 3rd edition, John Wiley & Sons, New York, USA, (1997).
- Chankong, V. and Haimes, Y., "Multiobjective Decision Making – Theory and Methodology, Vol. 8, North-Holland, Amsterdam, (1983).
- Coste-Manière, E., H. Wang, e A. Peuch, "Control architectures: what's going on?" In: *Proceedings of the International Program Development in URIC*, p. 54-60, Lisboa, Portugal (1995).
- Freire, E. O., Carelli, R., Mut, V., Soria, C., Bastos Filho, T. F., e Sarcinelli Filho, M., "Mobile Robot Navigation Based on the Fusion of Control Signals from Different Controllers". *Proceedings of the 2001 European Control Conference, ECC 01, Porto, Portugal (2001).*
- Latombe, J. C., "Robot Motion Planning", Kluwer Academic Publishers, Boston, USA (1991).
- Mutambara, A. G. O., "Decentralized Estimation and Control for Multi-sensor Systems", CRC Press, USA (1998).
- Pirjanian, P., Christensen, H. and Fayman, J., "Application Voting to Fusion of Purposive Modules: An Experimental Investigation", 1998.
- Pirjanian, P., "Multiple Objective Behavior-Based Control", *Robotics and Automation Systems*, 31:53-60, (2000).
- Rosenblatt, J., "DAMN: A Distributed Architecture for Mobile Navigation", PhD Thesis, Carnegie Melon University, USA (1997).
- Saffiotti, A., Konolige, K., and Ruspini, E., "A Multivaluated Logic Approach to Integrating Planning and Control", *Artificial Intelligence*, 76:481-526, March, (1995).
- Schöner, G., and Dose, M., "A Dynamics System Approach to Task Level System Integration Used to Plan and Control Autonomous Vehicle Motion", *Robotics and Automation Systems*, 10:253-267, (1992).