CONTROLE DE CONGESTIONAMENTO PARA ENXAMES DE ROBÔS NO ACESSO A ALVOS EM COMUM

Yuri Tavares dos Passos*, Luiz Chaimowicz*

* *VeRLab – Laboratório de Visão e Robótica*
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brasil

Emails: {yuri.passos, chaimo}@dcc.ufmg.br

Abstract— Robotic Swarms are systems formed by a large number of robots with common goals. In these systems, processing is performed in a decentralized manner and each robot has only local information obtained from its sensors. Among the problems encountered in swarms of robots, there exists traffic control. In this problem, robots must be coordinated to avoid congestion. In this paper, we developed an algorithm to coordinate robots efficiently in a scenario that robots have a common target.

Keywords— Robotic swarms, traffic control

Resumo— Enxames de robôs são sistemas formados por um grande número de robôs com objetivos em comum. Nestes sistemas, o processamento é realizado de forma descentralizada e cada robô possui apenas informações locais obtidas de seus sensores. Dentre os problemas enfrentados em enxames de robôs, têm-se o controle de tráfego. Neste problema, os robôs devem se coordenar para evitar congestionamentos. Neste trabalho, foi desenvolvido um algoritmo para coordenar os robôs de forma eficiente em um cenário onde os robôs possuem um alvo em comum.

Palavras-chave— Exames de robôs, controle de tráfego

1 Introdução

Enxames de robôs são sistemas compostos por uma grande quantidade de robôs, relativamente simples, dispostos no mesmo espaço e interagindo entre si para alcançar um objetivo comum. A inspiração dos enxames de robôs surge do comportamento de colônias de insetos, como formigas e abelhas. Nestas colônias, os indivíduos interagem entre si usando comunicação local e processamento individual, ou seja, não existe uma unidade central de controle administrando cada um dos indivíduos.

Num espaço em que existem vários robôs, a tendência de um robô interferir o outro é maior. Um dos problemas existentes na navegação de enxames é o congestionamento, especificamente quando os robôs devem se movimentar para o mesmo local simultaneamente. É comum que robôs possuam alvos distintos que compartilham uma mesma região do espaço ou um mesmo alvo em comum, como ocorre em navegação de pontos de interesse (*waypoint*). Isto também ocorre no problema de coleta (*foraging*) (Sahin et al., 2008) (Sahin, 2004), quando os robôs devem se deslocar para pontos do ambiente para coletar itens e transportá-los para um local específico. Pode-se citar aplicações para o problema da coleta como transporte de materias tóxicos ou escombros provenientes de desabamentos.

O problema do congestionamento poderia ser resolvido utilizando uma unidade central de processamento para computar as melhores trajetórias para cada robô. Porém, há a desvantagem do sistema ficar dependente desta unidade central, além de não ser uma solução escalável para um grande número de robôs. No contexto de enxames, um dos desafios no desenvolvimento destes sistemas é a garantia de que o todo irá funcionar corretamente sem que cada uma das partes tenha conhecimento global do ambiente.

O objetivo deste trabalho é desenvolver um algoritmo distribuído para o controle de tráfego em enxames de robôs que permita evitar congestionamentos tanto na chegada ao alvo em comum quanto na saída. A idéia principal é melhorar o algoritmo proposto em Marcolino and Chaimowicz (2009), incluindo um controle de regiões de acesso, tornando mais eficiente a chegada e saída do alvo. O algoritmo é executado de forma completamente distribuída e os robôs utilizam apenas comunicação e percepção locais.

2 Trabalhos Relacionados

Em Cao et al. (1997), é denominado conflito de recursos quando um recurso único e indivisível é requisitado por vários robôs. No contexto deste trabalho, o recurso em questão é o alvo, representado por uma pequena região do espaço. Há trabalhos das décadas de 80 e 90 que lidam com o controle de tráfego de robôs cooperativos, mas utilizam marcas para delimitar regiões de tráfego (Caloud et al., 1990) (Asama et al., 1991). Em Grossman (1988) é apresentado um algoritmo para controle de tráfego de veículos guiados automaticamente, mas o controle é também feito de maneira centralizada. O trabalho de Kato et al. (1992) trata este

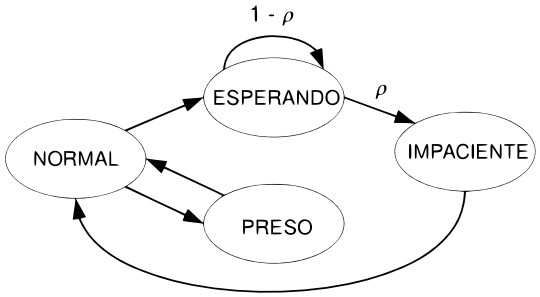


Figura 1: Máquina de estados do algoritmo.

problema de forma descentralizada utilizando regras de tráfego que cada robô obedece, mas não considera o problema específico onde todos desejam chegar no mesmo ponto.

Existem trabalhos recentes que lidam com controle de tráfego em grupos de robôs. Em Olmi et al. (2009), um algoritmo para coordenar caminhos predefinidos de um grupo de robôs é desenvolvido, mas em sua abordagem existe uma unidade central de processamento definindo a alteração das rotas de cada robô em caso de possível colisão. Em Guo and Parker (2002), um algoritmo distribuído é desenvolvido para planejamento de movimento para múltiplos robôs, contudo seus experimentos não envolvem muitos robôs. O trabalho de Peasgood et al. (2008) trata o problema de colisão de trajetórias para vários robôs, mas num contexto onde os alvos de cada um podem ser diferentes.

Nos trabalhos de Sahin (2004) e Sahin et al. (2008), que fazem uma revisão da área de enxames de robôs, não foram encontradas menções sobre o problema exposto neste trabalho, nem, de forma mais geral, sobre controle de tráfego em enxames de robôs. O único que se enquadra neste problema é o de Marcolino and Chaimowicz (2009) cuja continuação aqui está sendo proposta. Sendo assim, o presente trabalho sugere um diferente subtópico de estudo para a área de enxames de robôs.

3 Algoritmo distribuído para controle de congestionamento

A idéia principal do algoritmo proposto neste trabalho é evitar que muitos robôs cheguem simultaneamente ao mesmo alvo. Este algoritmo trata-se de uma extensão do trabalho proposto por Marcolino and Chaimowicz (2009), onde é descrito um algoritmo que consiste em cada robô usar uma máquina de estados probabilística para se coordenar dentro de uma região em volta do alvo. A idéia é fazer com que os robôs esperem sua vez para alcançar o alvo, se posicionando numa região próxima a ele, de forma a diminuir a chance de um robô atrapalhar o outro.

A máquina de estados, apresentada na Figura 1 (Marcolino and Chaimowicz, 2009), ilustra o comportamento de cada robô. Um robô em estado $NORMAL$ irá se mover para o alvo enquanto desvia de obstáculos. Se ele alcançar a região de perigo, região de raio s ao redor do alvo, e encontrar um robô próximo com o mesmo alvo que ele, então ele muda para o estado $ESPERANDO$. Existe também uma área circular menor, com raio $r < s$, chamada de região livre. Figura 2 (Marcolino and Chaimowicz, 2009) mostra as regiões de perigo e livre. Robôs no estado $ESPERANDO$ ou $PRESO$ não podem entrar nesta região. Cada robô em $ESPERANDO$ tem uma probabilidade ρ de mudar para o estado $IMPACIENTE$, na qual o robô irá para o alvo não se importando se existem outros robôs com o mesmo alvo. Robôs no estado $ESPERANDO$ podem continuar neste estado com probabilidade $1 - \rho$. Se um robô está no estado $NORMAL$ e encontra um robô no estado $ESPERANDO$ na sua frente, ele muda para o estado $PRESO$. No estado $PRESO$, o robô possui o mesmo comportamento do estado $ESPERANDO$, contudo, ele não depende de transições probabilísticas para mudar de estado, basta não haver mais robôs $ESPERANDO$ próximos e ele voltará ao estado $NORMAL$. É importante mencionar que este algoritmo não evita que os robôs se movam ao mesmo tempo para o alvo, mas diminui a chance disso acontecer devido a máquina de estados probabilística utilizada.

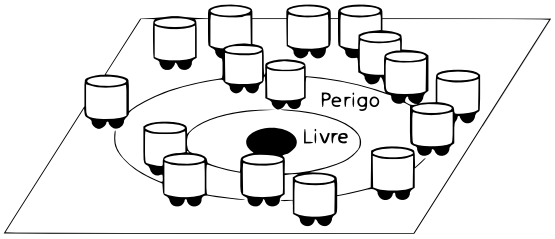


Figura 2: Área de perigo e livre.

Para os robôs perceberem quando existem alvos em comum, eles trocam mensagens com robôs detectados em sua proximidade, informando qual seu alvo e o tipo da mensagem. Os robôs só obedecem o algoritmo acima se o alvo é comum. Existem dois tipos de mensagem: de parada e de atenção. A mensagem de parada é enviada para aqueles que estão no estado $NORMAL$ e se encontram atrás dos que estão no estado $ESPERANDO$, fazendo que mudem para o estado $PRESO$. A mensagem de atenção é enviada para aqueles que estão na região livre, fazendo com que os que estão no estado $NORMAL$, mudem para o estado $ESPERANDO$.

Este algoritmo funciona sem congestionamentos até a chegada no alvo. No entanto, a saída dos robôs da região do alvo fica congestionada de-

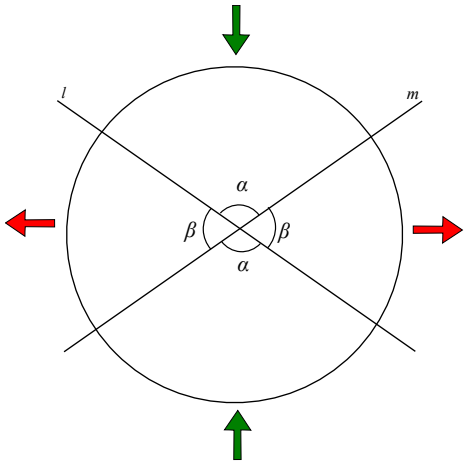


Figura 3: Divisão da região de perigo em áreas de entrada e saída.

vido à presença de outros robôs esperando para entrar. O algoritmo proposto neste trabalho pretende contornar este problema.

4 Algoritmo proposto

O algoritmo proposto neste trabalho visa dividir a área em torno do alvo em regiões para entrada e saída de robôs. Estabelecendo estas regiões, pretende-se diminuir os congestionamentos entre robôs que desejam chegar ao alvo e os que desejam sair. Para isso, a região de perigo será dividida em quatro regiões por ângulos α e β, como ilustra a Figura 3. A fatia dividida pelo ângulo α será usada para entrada dos robôs e a do ângulo β, para saída. Com isso espera-se que os robôs possam entrar e sair da região de alvo sem ocorrência de congestionamentos.

O algoritmo proposto neste trabalho estabelece um novo comportamento aos robôs quando estão próximos do alvo. Quando um robô se encontra a uma distância D do alvo, ele verifica se está dentro da região de entrada. Caso esteja fora, ele se move para dentro da região. A região de entrada é verificada considerando a posição do alvo, a posição atual do robô, usando um referencial global. Considere $\gamma = 90° - \alpha/2$, o ângulo da reta m, e $\delta = 90° + \alpha/2$, da reta l, conforme mostra a Figura 4 (as retas l e m também podem ser vistas na Figura 3). Dados (x_G, y_G), as coordenadas do alvo, e (x, y), a posição atual do robô, a seguinte condição determina se o robô está dentro da região:

$$\begin{aligned}
&(y - y_G - \tan(\gamma)(x - x_G) \geq 0) \wedge \\
&\quad (y - y_G - \tan(\delta)(x - x_G) \geq 0) \quad \text{se } y > y_G \\
&(y - y_G - \tan(\gamma)(x - x_G) \leq 0) \wedge \\
&\quad (y - y_G - \tan(\delta)(x - x_G) \leq 0) \quad \text{se } y \leq y_G
\end{aligned}$$

Se o robô não está na região, ele é impelido a se locomover para o ponto da reta que divide

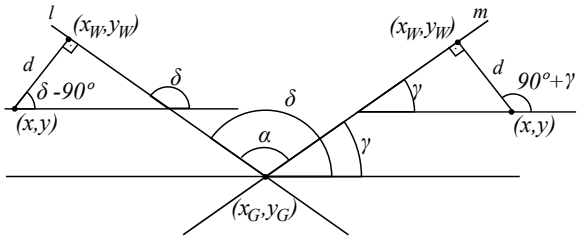


Figura 4: Distância mais próxima entre um ponto fora da região e a fronteira da região de entrada.

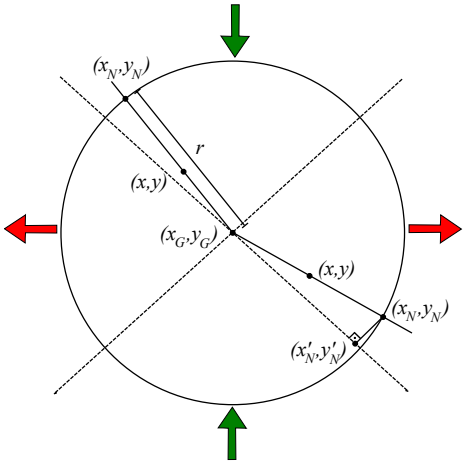


Figura 5: Nova posição do robô caso esteja na área livre.

a região de entrada mais próximo de sua posição atual. A distância d entre sua posição atual e o ponto mais próximo da região de entrada é dado por:

$$d = \frac{y - \tan(\theta)(x - x_G) - y_G}{\sqrt{(\tan^2(\theta) + 1)}} \tag{1}$$

onde

$$\theta = \begin{cases} \gamma & \text{se } ((y > y_G) \wedge (x > x_G)) \vee \\ & \quad ((y \leq y_G) \wedge (x \leq x_G)) \\ \delta & \text{caso contrário} \end{cases}$$

A Equação 1 pode retornar valores positivos ou negativos, a depender da posição do robô em relação a reta mais próxima.

Como ilustrado na Figura 4, o ponto (x_W, y_W) para o qual ele deve se locomover é dado por:

$$x_W = x + d\cos(\phi) \tag{2}$$

$$y_W = y_G + \tan(\theta)(x_W - x_G) \tag{3}$$

onde

$$\phi = \begin{cases} 90° + \gamma & \text{se } ((y > y_G) \wedge (x > x_G)) \vee \\ & \quad ((y \leq y_G) \wedge (x \leq x_G)) \\ \delta - 90° & \text{caso contrário} \end{cases}$$

Além das regiões delimitadas pelas retas, os robôs, ao entrarem nos estados $ESPERANDO$ ou $PRESO$, verificam se não estão na região livre

próxima ao alvo (Figura 2). Um robô em qualquer um destes dois estados, deve estar a no mínimo uma distância r do alvo. Um robô no estado $ESPERANDO$ deve estar a uma distância do alvo que varia de r a s. Se um robô está em $ESPERANDO$ e se afasta do alvo, ficando a mais de s metros, ele passa para o estado $PRESO$. Se um robô $ESPERANDO$ encontra-se a menos de r metros, ele deve ir para a posição (x_N, y_N) (Figura 5).

Para obter esta posição, considere

$$A = \frac{(y - y_G)^2}{(x - x_G)^2} + 1$$

Sendo assim, tem-se que:

$$x_N = \mp \frac{r}{\sqrt{A}} + x_G \tag{4}$$

$$y_N = y_G \pm r\sqrt{1 - \frac{1}{A}} \tag{5}$$

Tendo em vista a escolha da posição (x_N, y_N) mais próxima do robô, se $x < x_G$, x_N será calculado com a Equação 4 usando o sinal negativo (do contrário, com o sinal positivo). Analogamente, se $y < y_G$, y_N será obtido com a Equação 5 usando o sinal negativo (senão, com o sinal positivo). Caso o resultado (x_N, y_N) esteja fora da região de entrada, x_N e y_N são usados como entrada (no lugar de x e y) para as Equações 1-3. É importante ressaltar que se $x = x_G$, A será o resultado de uma divisão por zero. Dessa forma deve-se considerar que $x - x_G$ seja igual a um $\epsilon > 0$.

Os robôs também reagem à forças repulsivas relativos a seus vizinhos de modo diferente a depender do estado em que se encontram. Para gerar forças repulsivas foi usada a seguinte função (Siegwart and Nourbakhsh, 2004):

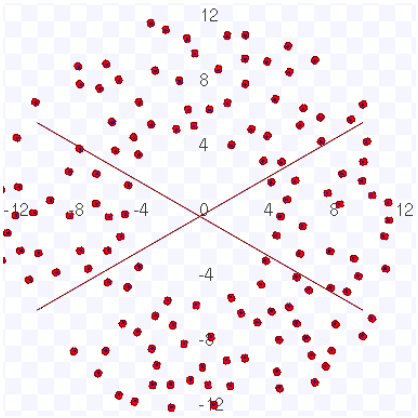
$$\mathbf{F} = \begin{cases} -K * \left(\frac{1}{d} - \frac{1}{I}\right)\left(\frac{\mathbf{o} - \mathbf{p}}{d^3}\right) & \text{se } d < I \\ 0 & \text{caso contrário} \end{cases}$$

onde $\mathbf{p} = [x, y]^T$ é a posição atual do robô, $\mathbf{o} = [x_o, y_o]^T$, a posição de outro robô, $d = ||\mathbf{o} - \mathbf{p}||$, a distância Euclidiana entre $\mathbf{o}$ e $\mathbf{p}$ e I a influência do outro robô.

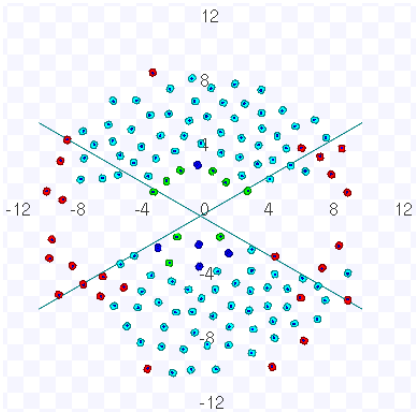
Se um robô estiver dentro da região de entrada, as forças repulsivas aplicadas a ele são reduzidas pela metade caso tentem expulsar o robô da região de entrada. Esta redução ocorre verificando se o sentido do vetor da força repulsiva cruza a linha delimitadora mais próxima de dentro para fora. Esta redução não impede que ele saia, mas reduz a quantidade de robôs na região de saída.

5 Resultados e Discussões

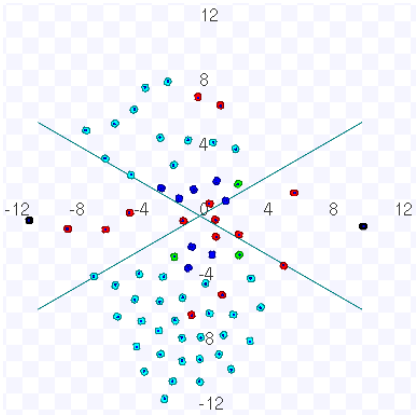
O algoritmo foi testado em simulações usando o simulador *Stage* (Gerkey et al., 2003). Os robôs



(a) Início da execução



(b) Formação das regiões de entrada e saída



(c) Execução mais à frente

Figura 6: Imagens da execução.

usaram um algoritmo de controle em comum, baseado em campos potenciais. Uma força atrativa é utilizada para alcançar o alvo, ao mesmo passo que forças repulsivas são aplicadas ao se aproximar de outros robôs. Na simulação, foram considerados robôs não holonômicos usando equações de controle baseados em Luca and Oriolo (1994).

Foi considerado um cenário onde os robôs encontram-se em posições aleatórias, mas distantes da região de perigo. Os robôs devem se movimentar para um alvo em comum e, após alcançá-lo, sair da região livre se deslocando para um novo alvo. Nesse artigo, considera-se que o novo alvo

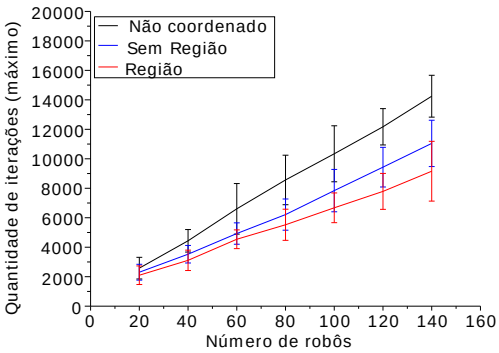


Figura 7: Tempo de execução dos algoritmos.

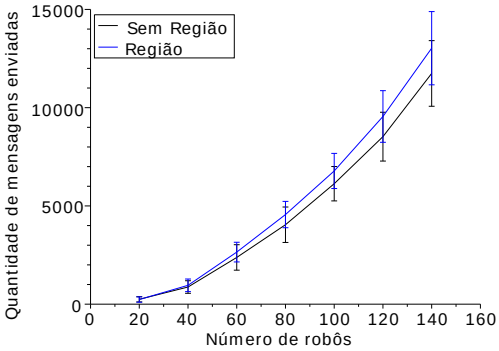


Figura 8: Quantidade de mensagens enviadas.

estará distante e alinhado com as direções de saída do alvo original (essa condição pode ser relaxada com algumas alterações no algoritmo, que não são tratadas nesse artigo). Foi medido a quantidade de iterações em que cada robô gasta para alcançar o alvo e depois sair da região, se movendo a uma distância de R metros do alvo original. O intuito desta medida de tempo é verificar a eficiência da chegada do robô ao alvo e de sua saída desta região. Cada simulação foi executada 20 vezes para cada quantidade de robôs usadas no experimento e para cada algoritmo. Foi determinado um valor fixo para o raio da área de perigo ($s = 3{,}5$ m) e da área livre ($r = 1{,}5$ m). O valor de D e R é 10 m.

A Figura 6 apresenta algumas imagens da execução do algoritmo proposto. Robôs em estado $NORMAL$, $ESPERANDO$, $PRESO$ e $IMPACIENTE$ são representados com as cores vermelha, verde, ciano e azul, respectivamente. Robôs que terminaram a execução estão representados com a cor preta. Na Figura 6(a), tem-se o início da execução. Algumas iterações após o início, a Figura 6(b) mostra quando as regiões estão começando a serem formadas. Observa-se que todos os robôs, independente de qual estado esteja, vão para a região de entrada mais próxima. Nota-se, na região próxima ao alvo, que alguns robôs estavam no estado $ESPERANDO$ e passaram para o estado $IMPACIENTE$. Percebe-se também que os robôs que ainda estão na região de saída estão apontados para as retas que dividem estas regiões, indicando que eles estão se movendo para a área de entrada. A Figura 6(c) ilustra uma iteração mais a frente quando alguns poucos robôs ainda não terminaram de entrar na região do alvo. Nota-se, tanto na regiões de entrada superior quanto inferior, que há dois robôs no estado NORMAL. Como não havia robôs no estado $ESPERANDO$, estes robôs, antes no estado $PRESO$, mudaram para o estado $NORMAL$ e seguiram em frente. Contudo, como discutido na Seção 3, ao chegarem próximo do alvo e se detectarem, eles deverão entrar no estado $ESPERANDO$, visto que haverá outros robôs tentando alcançá-lo.

Para avaliar o algoritmo proposto, ele foi comparado com o algoritmo de coordenação sem Regiões, proposto em Marcolino and Chaimowicz (2009), e sem algoritmo de coordenação, ou seja, usando apenas campos potenciais atrativos e repulsivos. O desempenho dos três algoritmos foi analisado quanto ao número de iterações gastas em relação à quantidade de robôs. A Figura 7 apresenta o resultado da execução dos três algoritmos quanto a quantidade de iterações máxima de cada execução, ou seja, o tempo de execução do último robô a sair (com intervalo de confiança de 95%). A quantidade de robôs foi variada de 20 a 140, com intervalos de 20. Para os dois algoritmos com máquinas probabilísticas, foi usado $\rho = 0,3$. Para o ângulo da região de entrada foi usado $\alpha = 120°$. Nota-se que, o algoritmo proposto possui um ganho considerável em relação ao algoritmo sem regiões. Isto se explica pela seleção de regiões de entrada e saída para impedir congestionamentos quando os robôs estão saindo da região do alvo.

A Figura 8 mostra a quantidade de mensagens enviadas para a execução dos dois algoritmos de coordenação. Nota-se que o algoritmo proposto é menos eficiente na quantidade de mensagens enviadas. Isto ocorre porque no algoritmo proposto os robôs $PRESO$ ou $ESPERANDO$ tendem a se concentrar na região de entrada. Com isso, os robôs ficam mais próximos, fazendo com mais robôs estejam em seu raio de percepção e consequentemente troquem mais mensagens. Apesar de não ter sido feita regressões, observa-se que a quantidade de mensagens enviadas é uma função que tende a ser quadrática quanto à quantidade de robôs e o tempo de execução é uma função que tende a ser linear.

O algoritmo proposto possui, na média, um desempenho melhor ao algoritmo sem regiões de entrada e saída, mas a mesma tendência de crescimento. Isto é alcançado devido a divisão da área

próxima ao alvo em regiões de entrada e de saída. Consequentemente, os robôs terão espaços livres para mover sair da região próxima ao alvo. Esta melhoria traz benefícios ao sistema como um todo, pois menos tempo é usado para atingirem a região próxima do alvo, com o custo de poucas mensagens a mais sendo enviadas.

6 Conclusões

Este trabalho apresentou um algoritmo utilizando máquinas de estados probabilísticos para o problema do controle de tráfego em enxames de robôs. Tratou-se de uma melhoria do algoritmo proposto por Marcolino and Chaimowicz (2009). A idéia principal desta melhoria era dividir a área em torno do alvo em regiões para entrada e saída de robôs. Estabelecendo estas regiões, pretendeu-se diminuir os congestionamentos entre robôs que desejavam chegar ao alvo e os que desejavam sair. O algoritmo proposto foi comparado com o algoritmo sem coordenação, ou seja, usando apenas campos potenciais atrativos e repulsivos, e o algoritmo original sem regiões. O algoritmo proposto possui um desempenho melhor no tempo gasto para alcançar o alvo gastando poucas mensagens a mais em relação ao algoritmo sem uso de regiões.

O problema do controle de tráfego em enxames de robôs não foi abordado na literatura de forma específica, por conseguinte o presente trabalho sugere um diferente subtópico de estudo para a área de enxames de robôs. Foi demonstrado que o algoritmo proposto neste trabalho possui um desempenho melhor para situações em que os robôs ficam congestionados durante sua saída de áreas com bastante tráfego.

Como trabalhos futuros pretende-se fazer testes com robôs reais e melhorar este algoritmo para situações em que os robôs possuem novos destinos no lado oposto a região de entrada. Também pretende-se desenvolver outros algoritmos para controle de tráfego usando outras abordagens e compará-los com o algoritmo proposto.

Referências

Asama, H., Habib, M., Endo, I., Ozaki, K., Matsumoto, A. and Ishida, Y. (1991). Functional distribution among multiple mobile robots in an autonomous and decentralized robot system, *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, Vol. 3, pp. 1921–1926.

Caloud, P., Choi, W., Latombe, J.-C., Le Pape, C. and Yim, M. (1990). Indoor automation with many mobile robots, *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, Vol. 1, pp. 67–72.

Cao, Y. U., Fukunaga, A. S. and Kahng, A. B. (1997). Cooperative mobile robotics: Antecedents and directions, *Autonomous Robots* **4**: 226–234.

Gerkey, B. P., Vaughan, R. T. and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems, *In Proceedings of the 11th International Conference on Advanced Robotics*, pp. 317–323.

Grossman, D. (1988). Traffic control of multiple robot vehicles, *Robotics and Automation, IEEE Journal of* **4**(5): 491 –497.

Guo, Y. and Parker, L. E. (2002). A distributed and optimal motion planning approach for multiple mobile robots, *In Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2612–2619.

Kato, S., Nishiyama, S. and Takeno, J. (1992). Coordinating mobile robots by applying traffic rules, *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, Vol. 3, pp. 1535–1541.

Luca, A. D. and Oriolo, G. (1994). Local incremental planning for nonholonomic mobile robots, *In Proceedings of 1994 International Conference on Robotics and Automation*, pp. 104–110.

Marcolino, L. and Chaimowicz, L. (2009). Traffic control for a swarm of robots: Avoiding target congestion, *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1955 –1961.

Olmi, R., Secchi, C. and Fantuzzi, C. (2009). A coordination technique for automatic guided vehicles in an industrial environment, *IFAC 9th International IFAC Symposium on Robot Control*.

Peasgood, M., Clark, C. and McPhee, J. (2008). A complete and scalable strategy for coordinating multiple robots within roadmaps, *Robotics, IEEE Transactions on* **24**(2): 283–292.

Sahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application., *Swarm Robotics'04*, pp. 10–20.

Sahin, E., Girgin, S., Bayindir, L. and Roboticsurgut, A. E. T. (2008). Swarm robotics, *Swarm Intelligence*, Springer, pp. 87–100.

Siegwart, R. and Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*, Bradford Company, Scituate, MA, USA.