Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica
Escola de Engenharia

Techniques for Controlling Swarms of Robots

Luciano Cunha de Araújo Pimenta

Thesis presented to the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

Advisor: Prof. Renato Cardoso Mesquita

Co-Advisor: Prof. Guilherme Augusto Silva Pereira

Foreign Advisor: Prof. Vijay Kumar

Belo Horizonte, February, 2009

Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica
Escola de Engenharia

Técnicas para o Controle de Enxames de Robôs

Luciano Cunha de Araújo Pimenta

Tese apresentada ao Curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Prof. Renato Cardoso Mesquita

Co-orientador: Prof. Guilherme Augusto Silva Pereira

Orientador Estrangeiro: Prof. Vijay Kumar

Belo Horizonte, Fevereiro de 2009

To my love, Carina.

Abstract

This thesis addresses the problem of controlling very large groups of robots, refereed as *swarms*. Scalable solutions in which there is no need for labelling the robots are proposed. All the robots run the same software and the success of the task execution does not depend on specific members of the group. Robustness to dynamic addition and deletion of agents is also an advantage of our approaches. In the first methodology, we model the swarm as a fluid immersed in a region where a field of external forces, which is free of local minima, is defined. In this case, the Smoothed Particle Hydrodynamics (SPH) method is applied to model the "robotic fluid", more specifically, to model the interactions among the robots of the group. The Finite Element Method (FEM) is also used in this work to compute the fields that determine external forces. This approach is instantiated in a pattern generation task and also in a coverage task. In the second methodology, a problem of optimal environment coverage using robots equipped with sensors is addressed by means of tools from the Locational Optimization theory. Three important extensions of well-known results in the literature are presented: (i) sensors with different footprints, (ii) disk-shaped robots, and (iii) nonconvex polygonal environments. Both approaches are verified in simulations. The first technique is also implemented and tested in actual robots.

Resumo

Esta tese aborda o problema de controle de grandes grupos de robôs, referidos como *exames*. São propostas soluções escaláveis as quais não necessitam da identificação única dos robôs. Todos os robôs executam o mesmo código e o sucesso na execução de uma tarefa não depende de membros específicos do grupo. Robustez à adição e remoção dinâmica de agentes também é uma vantagem das abordagens propostas. Na primeira metodologia, o enxame é modelado como um fluido imerso numa região onde um campo de forças externas livre de mínimos locais é definido. Neste caso, utiliza-se o método de Hidrodinâmica de Partículas Suavizadas (HPS) para modelar o "fluido robótico", mais especificamente, para modelar as interações entre robôs do grupo. O Método de Elementos Finitos (MEF) também é utilizado neste trabalho para calcular os campos vetoriais que determinam as forças externas. Esta abordagem é instanciada num problema de geração de padrões e também num problema de cobertura de ambientes. Na segunda metodologia, um problema de cobertura ótima de ambientes utilizando robôs equipados com sensores é tratado por meio de ferramentas provenientes da teoria de Otimização Locacional. São apresentadas três extensões importantes de resultados já conhecidos na literatura: (i) sensores com diferentes campos de visão, (ii) robôs com formato circular e (iii) ambientes poligonais não-convexos. Ambas metodologias são verificadas em simulações. A primeira metodologia é também implementada e testada em robôs reais.

Acknowledgments

I would like to thank first my wife, Carina, for her love, support, comprehension, and for always be with me. I would like to thank my whole family, specially my father, my mother, my brother, and my grandmother for their love, support, and for everything they have done for me. I am also very grateful to Carina's family for the great friendship and for always receiving me with happiness.

Thanks to Professor Renato Cardoso Mesquita and Professor Guilherme Augusto Silva Pereira who were fundamental in this work. Thanks for the shared knowledge , the excellent orientation, the trustiness, the friendship and for the doors that were opened.

Thanks also to Professor Vijay Kumar, who was also essential in the development of this work, for the great opportunities that he offered me and for the excellent orientation during the time I worked in his lab, GRASP, at the University of Pennsylvania.

I am grateful to the students of GOPAC, Miguel, Léo Corrêa, and Juliano for their help with the computational implementations and for their friendship. I am grateful to all members of GOPAC, specially to my friends Alexandre, Ricardo, Carrano, Douglas, and Mozelli for creating an excellent work environment. Thanks also to the remaining friends of GEP, VERLAB, and CPDEE for their friendship. Many thanks to the colleagues at the GRASP Lab for the shared knowledge, specially to Nathan Michael and

Quentin Lindsey for the help with experiments.

Many thanks to the members of my thesis committee, professors Fernando Cesar Lizarralde, Geovany Araújo Borges, Marco Henrique Terra, Leonardo Antônio Borges Tôrres, Luiz Chaimowicz, and Elson José da Silva for their valuable suggestions.

Thanks to professors Mário Campos, Hani, Walmir, Paulo Seixas, Marcos Severo, Porfírio, Pedro Donoso, and Jaime for the shared knowledge and for always be available to help when I needed. Thanks also to the remaining professors that somehow were essential in my academic formation.

I would like to thank CNPq and FAPEMIG for the financial support and everyone that contributed to this work.

Agradecimentos

Gostaria de agradecer primeiramente à minha esposa, Carina, pelo amor, apoio, compreensão e por sempre estar ao meu lado. Gostaria de agradecer à toda a minha família, especialmente ao meu pai, à minha mãe, ao meu irmão e à minha avó pelo amor, apoio e por tudo que fizeram por mim. Sou muito grato também à família da Carina pela grande amizade e por sempre me receberem com alegria.

Agradeço ao Professor Renato Cardoso Mesquita e Professor Guilherme Augusto Silva Pereira os quais foram essenciais na realização deste trabalho. Agradeço pelo conhecimento compartilhado, pela excelente orientação, pela confiaça, pela amizade e pelas portas que me foram abertas.

Agradeço também ao Professor Vijay Kumar, o qual foi também essencial no desenvolvimento deste trabalho, pelas grandes oportunidades que ele me proporcionou e pela excelente orientação durante o doutorado sanduíche realizado em seu laboratório, GRASP, na Universidade da Pensilvânia.

Sou grato aos alunos do GOPAC, Miguel, Léo Corrêa e Juliano pelo grande apoio nas implementações computacionais e pela amizade. Sou grato aos demais membros do GOPAC, especialmente aos amigos Alexandre, Ricardo, Carrano, Douglas e Mozelli por proporcionarem um ótimo ambiente de trabalho. Agradeço também aos demais amigos do GEP, do VERLAB e do CPDEE pelo grande companheirismo. Agradeço também aos colegas do GRASP Lab pela troca de conhecimentos, especialmente Nathan Michael e

Quentin Lindsey pela ajuda na realização de experimentos.

Gostaria também de agradecer aos membros da minha banca, professores Fernando Cesar Lizarralde, Geovany Araújo Borges, Marco Henrique Terra, Leonardo Antônio Borges Tôrres, Luiz Chaimowicz e Elson José da Silva pelas valiosas sugestões.

Muito obrigado aos professores Mário Campos, Hani, Walmir, Paulo Seixas, Marcos Severo, Porfírio, Pedro Donoso e Jaime pelo conhecimento ensinado e por estarem sempre disponíveis quando precisei. Agradeço também aos demais professores que foram fundamentais na minha formação.

Meus agradecimentos ao CNPq e FAPEMIG pelo suporte financeiro e a todos aqueles que de alguma forma contribuíram para a realização deste trabalho.

Resumo Estendido

Introdução

A Robótica cooperativa é o campo da robótica dedicado ao estudo de técnicas que permitem que robôs em um time cooperem entre si e com seres humanos para realizar uma dada tarefa. Para uma grande variedade de tarefas, sistemas robóticos cooperativos fornecem soluções que não podem ser obtidas utilizando um único robô. Além disso, mesmo em situações onde um único robô possa ser utilizado, o uso de um time pode ter custo mais baixo, ser mais confiável, ser mais tolerante a falhas e ainda mais flexível.

Controlar grupos de robôs tem sido um desafio e diferentes soluções já foram propostas. O tamanho do grupo é um fator crucial que determina o tipo mais apropriado de abordagem. Basicamente, existem duas categorias de abordagens: centralizadas e descentralizadas. As abordagens centralizadas são aquelas que assumem a existência de uma entidade central, a qual é capaz de planejar as ações de cada robô do grupo. Embora, em geral, se possa provar que a utilização de tal abordagem garante sucesso na realização da tarefa, este tipo de técnica não é escalável para grandes grupos de robôs por razão de limitações computacionais. Por outro lado, abordagens descentralizadas fornecem soluções escaláveis uma vez que neste caso cada agente do grupo planeja suas próprias ações baseadas em informações locais.

Em acordo com as idéias de soluções escaláveis, um novo paradigma

chamado *exame* de agentes autônomos tem sido estudado no campo da robótica. Neste paradigma o objetivo é controlar grandes grupos de robôs (dezenas a centenas) muito simples. A idéia chave é que o sucesso na execução de uma tarefa dependerá dos comportamentos que vão emergir das interações entre agentes. Neste contexto, cada agente deve ser o mais simples possível com capacidades limitadas de comunicação, sensoreamento e atuação. Outra característica importante de um enxame é a flexibilidade. Utilizando diferentes mecanismos de coordenação o mesmo enxame pode ser utilizado em diferentes problemas. Um outro ponto importante é que a coordenação do enxame não deve depender de membros específicos do grupo. Logo, soluções totalmente descentralizadas, onde os agentes podem ser considerados anônimos e podem ser programados com o mesmo código, devem ser consideradas. Além disso, tais soluções devem ser robustas à adição e remoção dinâmica de robôs.

O principal objetivo desta tese é o desenvolvimento de estratégias para a coordenação de enxames de robôs. Na primeira parte do trabalho, o foco está no uso de analogias com modelos provenientes da dinâmica dos fluidos para o controle do enxame. A principal motivação está no fato de uma grande variedade de características desejáveis para um grupo de robôs ser observada em fluidos. Alguns exemplos são: (i) fluidos são facilmente deformáveis; (ii) fluidos podem contornar obstáculos; e (iii) as variáveis do fluxo e também a fase do fluido podem ser manipuladas para projetar os comportamentos desejados. Esta abordagem via dinâmica dos fluidos é verificada em duas tarefas: formação de padrões e cobertura de ambientes. Na segunda parte do trabalho, o problema de cobertura ótima de ambiente é tratado por meio de ferramentas provenientes da teoria de otimização locacional [Okabe et al., 2000].

Contribuições

Este trabalho contribuiu para a área de enxames robóticos com novas técnicas escaláveis para a coordenação de enxames. As contribuições principais são:

- Controladores descentralizados por meio do acoplamento entre dois métodos numéricos: a Hidrodinâmica de Partículas Suavizadas (HPS) e o Método de Elementos Finitos (MEF). Os controladores propostos dependem apenas de informações locais. Além disso, todos os robôs podem ser considerados entidades anônimas. Um tratamento eficiente de obstáculos também é parte da metodologia proposta.

- Uma solução para o problema de geração de padrões utilizando modelo de fluido incompressível. O enxame é modelado como um fluido incompressível sujeito a forças externas. Como os robôs são controlados afim de manter a densidade do fluido constante, esta abordagem possibilita uma maneira indireta de controlar fracamente a conectividade do grupo. Problemas relacionados à implementação em robôs reais como o tamanho do robô e ainda restrições não-holonômicas são abordados. Garantias de desvio de obstáculos são discutidas. Na ausência de obstáculos, apresentam-se pela primeira vez provas de estabilidade e convergência de controladores baseados em HPS. [Pimenta et al., 2006b, Pimenta et al., 2006c, Pimenta et al., 2007a, Pimenta et al., 2008b]

- Uma solução para um problema de cobertura restrita. O modelo de fluido incompressível é aplicado e o time de robôs é guiado para maximizar a cobertura num ambiente genérico desconhecido enquanto a densidade é mantida constante.

- Três novas e importantes extensões de uma abordagem para cobertura ótima utilizando redes de sensores móveis [Cortez et al., 2004]: (i) incorporação de heterogeneidade no time de robôs permitindo que diferentes tipos de sensores sejam utilizados. Utiliza-se o chamado *Power diagram* [Aurenhammer, 1987] para definir a região de dominância de cada agente; (ii) solução das limitações práticas relativas à consideração de robôs pontuais por meio da introdução de um problema de minimização com restrições; e (iii) generalização para ambientes não-convexos com a introdução de diagramas de Voronoi geodésicos [Aronov, 1989] na lei de controle. [Pimenta et al., 2008a]

Coordenação de Enxames Utilizando Modelos de Dinâmica dos Fluidos

Na primeira parte desta tese, modela-se o enxame como um fluido sujeito a forças externas. Nesta abordagem, aplica-se a Hidrodinâmica de Partículas Suavizadas (HPS) para modelar o "fluido robótico". Mais especificamente, a HPS é utilizada para modelar as interações entre robôs do grupo. O método HPS foi originalmente proposto em [Lucy, 1977, Gingold and Monaghan, 1977] para resolver problemas de astrofísica, mas hoje em dia este método tem sido largamente empregado na solução de problemas de dinâmica dos fluidos. Este fato se deve principalmente à habilidade deste método de tratar fronteiras móveis e grandes deformações [Liu and Liu, 2003]. O Método de Elementos Finitos (MEF) [Ida and Bastos, 1992] também é utilizado nesta tese para calcular o campo vetorial que modela forças externas ao fluido. Como os obstáculos podem ter geometrias genéricas, o MEF permite eficiência no cálculo do campo.

Isto se deve à habilidade de tal método no uso de malhas não-estruturadas. Controladores descentralizados são derivados a partir de um acoplamento entre a HPS e o MEF.

A HPS é uma técnica sem malhas, Lagrangiana, baseada em partículas. As equações da HPS são derivadas a partir das equações governantes contínuas por meio de uma interpolação a partir de um conjunto desordenado de partículas. Em problemas de dinâmica dos fluidos, cada partícula representa um pequeno volume do fluido e a interpolação é realizada pelo uso de funções núcleo, W, diferenciáveis que aproximam uma função delta. Neste trabalho utilizam-se *splines* cúbicas:

$$W(\mathbf{r},h) = \frac{10}{7\pi h^2}\begin{cases} 1-\frac{3}{2}\kappa^2+\frac{3}{4}\kappa^3 & \text{se} \quad 0\leq\kappa\leq 1\,, \\ \frac{1}{4}(2-\kappa)^3 & \text{se} \quad 1\leq\kappa\leq 2\,, \\ 0 & \text{caso contrário}\,, \end{cases} \tag{1}$$

onde $\kappa = \|\mathbf{r}\|/h$. Pode-se observar que o suporte desta função é dado por $2h$.

As equações governantes contínuas são convertidas num conjunto de equações diferenciais ordinárias, onde cada uma controla a evolução de um atributo de uma partícula específica. As equações governantes de dinâmica dos fluidos são três: (i) conservação da massa; (ii) conservação do momento; e (iii) conservação da energia. Para fluidos compressíveis na ausência de fluxo de calor, as equações de conservação correspondentes da HPS para a partícula i são:

$$\rho_i = \sum_j m_j W(\mathbf{q}_i - \mathbf{q}_j, h)\,, \tag{2}$$

$$\frac{d\mathbf{v}_i}{dt} = -\sum_j m_j \left(\frac{P_i}{\rho_i^2}+\frac{P_j}{\rho_j^2}+\Pi_{ij}\right)\nabla_i W_{ij} + \mathbf{f}_i\,, \tag{3}$$

$$\frac{de_i}{dt} = \frac{1}{2}\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij}\right) \mathbf{v}_{ij} \cdot \nabla_i W_{ij}\,, \tag{4}$$

onde ρ é densidade, $\mathbf{v}$ é velocidade, P é pressão, e é energia interna por unidade de massa, $W_{ij} = W(\mathbf{q}_i - \mathbf{q}_j)$ é a função núcleo, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ e $\mathbf{f}_i$ é a soma das forças externas normalizadas pela massa m_i. O termo Π_{ij} é o termo de viscosidade artificial incorporado para tratar choques. Existem diversas variantes para este termo e a mais utilizada é dada por [Monaghan, 1992]:

$$\Pi_{ij} = \begin{cases} \frac{1}{\overline{\rho}_{ij}}(-\xi_1 \overline{c}_{ij}\mu_{ij} + \xi_2 \mu_{ij}^2) & \text{if} \quad \mathbf{v}_{ij}\cdot\mathbf{q}_{ij} < 0\,, \\ 0 & \text{if} \quad \mathbf{v}_{ij}\cdot\mathbf{q}_{ij} > 0\,, \end{cases} \tag{5}$$

onde

$$\mu_{ij} = \frac{h\mathbf{v}_{ij}\cdot\mathbf{q}_{ij}}{\|\mathbf{q}_{ij}\|^2 + \eta^2} \quad . \tag{6}$$

Em (5), $\overline{\rho}_{ij}$ é a média entre as densidades das partículas i e j, ξ_1 e ξ_2 são constantes de viscosidade, $\overline{c}_{ij}$ é a média das velocidades do som e η^2 é um termo para evitar singularidades.

Para a modelagem de fluidos incompressíveis considera-se a seguinte equação de estado [Monaghan, 1994]:

$$P_i = B_i \left[\left(\frac{\rho_i}{\rho_0}\right)^{\gamma} - 1\right], \tag{7}$$

onde ρ_0 é a densidade de referência e B_i é o chamado módulo *Bulk*. Este módulo está relacionado à compressibilidade do fluido. Com a utilização de (7), o sistema é forçado a regular sua densidade para a densidade de referência.

Assume-se que cada robô do time é uma partícula HPS. Como as funções núcleo possuem suporte compacto, é possível derivar leis de controle descentralizadas baseadas nas equações da HPS. Os controladores são descen-

tralizados no sentido de que apenas informação local é necessária: o campo externo na posição do robô e posições e velocidades de robôs vizinhos. Para um robô i com configuração $\mathbf{q}_i = [x_i, y_i]^T$, os robôs vizinhos são definidos como aqueles no conjunto de vizinhança $\mathcal{N}_i$:

$$\mathcal{N}_i = \{j \neq i | \|\mathbf{q}_j - \mathbf{q}_i\| < D\}\,, \tag{8}$$

onde a distância D é determinada pelo tamanho do suporte da função núcleo.

Primeiramente, assume-se também que os robôs são pontuais e holonômicos com modelo dado por:

$$\ddot{\mathbf{q}}_i = \mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}, t)\,, \tag{9}$$

onde $\mathbf{q} = [\mathbf{q}_1^T, \ldots, \mathbf{q}_N^T]^T$ é a configuração do grupo.

O primeiro problema tratado é o problema de geração de padrões geométricos bidimensionais:

Problem 0.1 *Seja um grupo de N robôs com distribuição espacial inicial qualquer, o ambiente com obstáculos estáticos definindo um domínio compacto $\Omega \subset \mathbb{R}^2$ e uma curva $\Gamma : I \to \Omega$, onde $I \subset \mathbb{R}$. Encontre um controlador que guie os robôs para formar o padrão descrito por Γ sem colidir uns com os outros e sem colidir com os obstáculos estáticos.*

Para resolver tal problema propõe-se o controlador:

$$\mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{b}_i - \zeta \mathbf{v}_i + k\mathbf{f}_i\,, \tag{10}$$

onde

$$\mathbf{b}_i = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij}\,, \tag{11}$$

k e ζ são constantes positivas de ajuste e $\mathbf{f}_i$ é determinado a partir do vetor $-\nabla\phi$. A função ϕ é uma função potencial que garante convergência das curvas intergrais de $-\nabla\phi$ para o padrão desejado. Em ambientes com obstáculos utilizam-se funções harmônicas calculadas por meio do Método de Elementos Finitos. O problema de valor de contorno a ser resolvido é dado por:

$$\begin{cases} \nabla^2\phi = 0\,, \\ \phi(\Gamma_1) = \phi(\Gamma_2) = 0\,, \\ \phi(\partial\Omega_1) = \phi(\partial\Omega_2) = \phi(P) = V_c\,, \end{cases} \tag{12}$$

onde ϕ é a função harmônica, V_c é uma constante positiva, e P é um ponto definido no interior do padrão para o caso de curvas fechadas. As curvas Γ_1 e Γ_2 definem as fronteiras de uma região tal que Γ está em seu interior. Além disso, $\partial\Omega_1$ define as fronteiras externas do domínio e $\partial\Omega_2$ define as fronteiras dos obstáculos.

Para ambientes sem obstáculos podem-se utilizar as *shape functions*. Estas funções são funções positivas semi-definidas com valor mínimo igual a zero sobre a curva desejada. São obtidas provas de estabilidade e convergência utilizando-se tais funções.

O segundo problema tratado é um problema de cobertura restrita:

Problem 0.2 *Encontre controladores descentralizados que guiem uma rede de sensores móveis para uma configuração final que maximiza a cobertura sensorial total e mantém a densidade maior ou igual a um valor de referência* ρ_0.

Este problema é resolvido utilizando-se a mesma lei de controle anterior em (10) com forças externas iguais a zero. O problema desta abordagem é a dificuldade em se obter provas sobre a otimalidade da configuração final obtida.

Problemas relacionados à implementação da técnica proposta em robôs reais também são tratados. O fato dos robôs reais não serem pontuais é tratado por meio de uma adaptação da viscosidade artificial, uma vez que este termo garante termos repulsivos que evitam colisões entre partículas. A adaptação é dada por:

$$\mu_{ij} = \frac{h\mathbf{v}_{ij} \cdot \mathbf{q}_{ij}}{(\|\mathbf{q}_{ij}\| - (2R + \varepsilon))^2}\,, \tag{13}$$

onde R é o raio do robô e ε é um fator de segurança.

Restrições não-holonômicas também são tratadas. Mais especificamente, consideram-se as restrições a que estão sujeitos robôs diferenciais. Neste caso, utiliza-se a técnica de linearização por realimentação de estados [Murray et al., 1994]:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\frac{\sin(\theta)}{d} & \frac{\cos(\theta)}{d} \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix}\,, \tag{14}$$

onde v é a velocidade linear, ω é a velocidade angular e d é a distância do ponto de referência da linearização ao centro do robô. O vetor de entrada $[\dot{x}_d, \dot{y}_d]^T$ corresponde às velocidades desejadas.

Utiliza-se ainda neste trabalho uma estratégia de partículas virtuais. Estas partículas são colocadas exatamente sobre a fronteira dos obstáculos. Tais partículas são utilizadas para garantir que não existam colisões entre robôs e obstáculos, uma vez que as forças externas podem ser insuficientes para evitar tais colisões.

São realizadas simulações numéricas ideais, empregando-se o modelo em (9). Simulações realísticas em $3D$ são realizadas utilizando-se o ambiente GAZEBO [Gerkey et al., 2003]. Tal ambiente permite simular condições

próximas do mundo real. Os experimentos foram realizados utilizando-se a infra-estrutura do GRASP (General Robotics, Automation, Sensing, and Perception) Lab na Universidade da Pensilvânia, Estados Unidos. Esta infra-estrutura conta com robôs *Scarabs* e um sistema de câmeras no teto que permite localizar cada robô no chão durante o experimento. Detalhes desta infra-estrutura podem ser encontrados em [Michael et al., 2008].

Cobertura Ótima de Ambientes Baseada em Otimização Locacional

Uma abordagem distribuída e assíncrona para cobertura sensorial ótima de uma região convexa utilizando agentes sensores móveis idênticos pontuais é proposta em [Cortez et al., 2004]. Cada agente (robô) segue uma lei de controle baseada em gradiente descendente que tem o objetivo de minimizar um funcional que codifica a qualidade da cobertura sensorial. Esta lei de controle depende apenas de informações de posições dos robôs e dos seus vizinhos imediatos. Neste contexto, vizinhos são definidos como os robôs localizados em células de Voronoi vizinhas. O funcional também utiliza uma função densidade que determina pesos para pontos ou áreas do ambiente que são mais importantes que outras. Assim, áreas com altos valores de função densidade devem ser melhor cobertas que áreas com baixos valores.

Seja uma representação do ambiente $\Omega \subset \mathbb{R}^N$. Seja também $\mathcal{P} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ a configuração de n sensores móveis, onde $\mathbf{q}_i \subset \Omega$. Considere a partição $T = \{T_1, \ldots, T_n\}$ tal que $I(T_i) \cap I(T_j) = \emptyset$, $\forall i \neq j$, onde $I(\cdot)$ representa o interior de uma dada região e $\cup_{i=1}^{n} T_i = \Omega$. A idéia chave é que cada agente i é responsável pela cobertura da região T_i. O *funcional de*

cobertura que mede o desempenho do sistema é definido por:

$$\mathcal{H}(\mathcal{P}, T) = \sum_{i=1}^{n} \mathcal{H}(\mathbf{q}_i, T_i) = \sum_{i=1}^{n} \int_{T_i} f(d(\mathbf{q}, \mathbf{q}_i))\varphi(\mathbf{q}) d\mathbf{q}\,, \tag{15}$$

onde d corresponde a uma função que mede distâncias entre pontos $\mathbf{q} \in \Omega$ e sensores. A função $\varphi : \Omega \to \mathbb{R}_+$ é a função densidade. Esta função reflete um conhecimento da probabilidade de ocorrência de eventos em diferentes regiões, ou simplesmente uma medida de importância relativa de diferentes regiões de Ω. A função $f : \mathbb{R} \to \mathbb{R}$ é uma função suave estritamente crescente sobre a imagem de d, que mede a degradação do desempenho de sensores com a distância. O problema de cobertura do ambiente, Ω, é traduzido para o problema de minimização do funcional em (15).

Primeiramente, prova-se que uma condição necessária para um mínimo do funcional em (15) é que a partição T deve ser uma partição de Voronoi, $V(\mathcal{P})$, de acordo com a função de distância d. Além disso, prova-se a condição necessária:

$$\frac{\partial \mathcal{H}(\mathcal{P})}{\partial \mathbf{q}_i} = \frac{\partial \mathcal{H}(\mathbf{q}_i, V_i)}{\partial \mathbf{q}_i} = \int_{V_i} \frac{\partial}{\partial \mathbf{q}_i} f(d(\mathbf{q}, \mathbf{q}_i))\varphi(\mathbf{q}) d\mathbf{q} = 0. \tag{16}$$

Dado o conjunto de pontos $\mathcal{P} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ distribuídos sobre o domínio Ω, com fronteira $\partial\Omega$, a célula de Voronoi V_i associada ao ponto $\mathbf{q}_i$ de acordo com a função de distância d é definida por:

$$V_i = \{\mathbf{q} \in \Omega | d(\mathbf{q}, \mathbf{q}_i) \leq d(\mathbf{q}, \mathbf{q}_j), \forall j \neq i\}. \tag{17}$$

A coleção de tais regiões forma a chamada partição de Voronoi.

Em [Cortez et al., 2004] utiliza-se distância Euclidiana como d, e $f(d) = d^2$. Além disso, assume-se um ambiente convexo. Neste contexto, prova-se

que as células de Voronoi são politopos convexos. Neste caso o mínimo do funcional de cobertura é obtido quando os agentes estão localizados exatamente no centróide das células de Voronoi correspondentes. O centróide é dado por:

$$\mathbf{q}_i^* = \frac{\int_{V_i} \mathbf{q}\varphi(\mathbf{q})d\mathbf{q}}{\int_{V_i} \varphi(\mathbf{q})d\mathbf{q}}. \tag{18}$$

Considera-se o modelo cinemático para os agentes:

$$\dot{\mathbf{q}}_i = \mathbf{u}_i. \tag{19}$$

A lei de controle proposta em [Cortez et al., 2004] é dada por:

$$\mathbf{u}_i = -k(\mathbf{q}_i - \mathbf{q}_i^*)\,, \tag{20}$$

onde k é um ganho positivo. Esta lei de controle é baseada no gradiente descendente uma vez que:

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} = 2\left(\int_{V_i} \varphi(\mathbf{q})d\mathbf{q}\right)(\mathbf{q}_i - \mathbf{q}_i^*).$$

É importante ressaltar que em geral o sistema converge para um mínimo local já que o funcional $\mathcal{H}$ é não-convexo.

Nesta tese, são propostas três extensões para a estratégia em [Cortez et al., 2004]. A primeira extensão considera sensores heterogêneos com campos de visão circulares com diferentes raios $R_{\mathbf{q}_i}$. O funcional de cobertura, neste caso, é dado por:

$$\mathcal{H}(\mathcal{P}, PV) = \sum_{i=1}^{n} \int_{PV_i} [\|\mathbf{q} - \mathbf{q}_i\|^2 - R_{\mathbf{q}_i}^2]\varphi(\mathbf{q})d\mathbf{q}\,, \tag{21}$$

onde $f(d) = d$ e $d(\mathbf{q}, \mathbf{q}_i) = \|\mathbf{q} - \mathbf{q}_i\|^2 - R_{\mathbf{q}_i}^2$ é a chamada *power dis-*

tance [Aurenhammer, 1987]. A partição requerida neste caso é o chamado *power diagram*, PV.

O *power diagram*, PV, associa uma região, PV_i a cada círculo $B_i(\mathbf{q}_i, R_{\mathbf{q}_i})$ em $\mathbb{R}^2$. Tal região é definida por:

$$PV_i = \{\mathbf{q} \in \mathbb{R}^2 | d(\mathbf{q}, \mathbf{q}_i) \leq d(\mathbf{q}, \mathbf{q}_j), \forall j \neq i\}, \tag{22}$$

onde a *power distance* $d(\mathbf{q}, \mathbf{q}_i) = \|\mathbf{q} - \mathbf{q}_i\|^2 - R^2_{\mathbf{q}_i}$.

A lei de controle para esta extensão é a mesma apresentada anteriormente em (20) com a diferença de que o centróide agora é dado por:

$$\mathbf{q}_i^* = \frac{\int_{PV_i} \mathbf{q}\varphi(\mathbf{q})d\mathbf{q}}{\int_{PV_i} \varphi(\mathbf{q})d\mathbf{q}}. \tag{23}$$

O que é mais interessante na utilização da *power distance* é que as regiões de dominância de cada agente serão novamente politopos convexos.

A segunda extensão trata de agentes sensores não pontuais. Mais especificamente, agentes circulares com raio $r_{\mathbf{q}_i}$. O funcional de cobertura neste caso é o mesmo utilizado em [Cortez et al., 2004], ou seja, d é a distância Euclidiana e $f(d) = d^2$. Define-se, primeiramente, a região de Voronoi livre $\mathcal{F}_{V_i}$:

$$\mathcal{F}_{V_i} = \{\mathbf{q} \in V_i | \|\mathbf{q} - \mathbf{q}_{\partial V_i}\| \geq r_{\mathbf{q}_i}, \forall \mathbf{q}_{\partial V_i}\}, \tag{24}$$

onde $\|\cdot\|$ é a norma Euclidiana e $\mathbf{q}_{\partial V_i}$ é um ponto da fronteira da região de Voronoi, ∂V_i. Pode-se mostrar que $\mathcal{F}_{V_i}$ é um politopo convexo.

Propõe-se então a solução do problema de minimização com restrições:

$$\begin{aligned} &\min_{\mathbf{q}_i} \mathcal{H}(\mathcal{P}, V) \\ &\text{s.t.} \\ &g_{i1}(\mathbf{q}_i) \leq 0 \,, \ldots \,, g_{im}(\mathbf{q}_i) \leq 0 \end{aligned} \tag{25}$$

onde $g_{il}(\mathbf{q}_i) = 0$ define a l-ésima face da região $\mathcal{F}_{V_i}$. A minimização é realizada por meio de uma lei de controle baseada num método de projeção de gradiente:

1. Se nenhuma restrição está ativa

$$\mathbf{u}_i = -k \frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} \,, \tag{26}$$

2. Caso contrário

$$\mathbf{u}_i = k\pi \left(-\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}, \partial \mathcal{F}_{V_i} \right) \,, \tag{27}$$

onde $\pi \left(-\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}, \partial \mathcal{F}_{V_i} \right)$ fornece a projeção do vetor $-\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}$ sobre o vetor $\mathbf{t}_{il}$, o qual é um vetor unitário tangente à l-ésima face da região $\mathcal{F}_{V_i}$.

A terceira e última extensão considera ambientes poligonais não-convexos. O funcional de cobertura é dado por:

$$\mathcal{H}(\mathcal{P}, T) = \sum_{i=1}^{n} \mathcal{H}(\mathbf{q}_i, T_i) = \sum_{i=1}^{n} \int_{T_i} d(\mathbf{q}, \mathbf{q}_i)^2 \varphi(\mathbf{q}) d\mathbf{q} \,, \tag{28}$$

onde $d(\mathbf{q}, \mathbf{q}_i)$ é o comprimento do caminho mais curto entre $\mathbf{q}$ e $\mathbf{q}_i$, ou seja, é a distância geodésica entre os dois pontos em questão. A partição requerida neste caso é o diagrama de Voronoi geodésico [Aronov, 1989].

A lei de controle proposta é dada por:

$$\mathbf{u}_i = -k\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} = 2k\int_{V_i} d(\mathbf{q}_i, \mathbf{q})\varphi(\mathbf{q})\mathbf{z}_{\mathbf{q_i},\mathbf{q}} d\mathbf{q}\,, \tag{29}$$

onde $\mathbf{z}_{\mathbf{q_i},\mathbf{q}}$ é um vetor unitário com direção do vetor suporte do primeiro segmento do caminho mais curto entre $\mathbf{q_i}$ e $\mathbf{q}$.

Simulações numéricas ideais utilizando o modelo em (19) verificam as leis de controle propostas.

Conclusões

Foi proposto o uso da técnica de simulação Hidrodinâmica de Partículas Suavizadas para modelar o enxame como um fluido. Mais especificamente, foi utilizado um modelo de fluido incompressível o qual fornece controle de densidade. Este controle de densidade fornece uma maneira fraca de controlar a conectividade do grupo. Foi utilizado o Método de Elementos Finitos para o cálculo de funcões harmônicas que determinam forças externas para o fluido. Estas forças são principalmente responsáveis por guiar o enxame para a região desejada da área de trabalho. Como os obstáculos podem ter geometrias genéricas, o uso do Método de Elementos Finitos possibilita eficiência no cálculo da função harmônica. Por meio de um acoplamento entre o Método de Elementos Finitos e a Hidrodinâmica de Partículas Suavizadas, os controladores derivados são descentralizados no sentido de que apenas informações locais são utilizadas por cada robô: o gradiente da função harmônica na posição do robô e as posições e velocidades de robôs vizinhos. Pela primeira vez, garantias matemáticas como estabilidade e convergência de controladores derivados a partir das equações da Hidrodinâmica de Partículas Suavizadas são estabelecidas. A abordagem proposta foi instanciada com sucesso no pro-

blema de geração de padrões geométricos e também no problema de cobertura com densidade restrita. Técnicas para acomodar as características de robôs reais como o tamanho e as restrições não-holonômicas são propostas. Basicamente, isto é atingido pelo uso da linearização por realimentação de estados e também pela adaptação do termo de viscosidade artificial das equações de fluido. O campo vetorial calculado a partir da função harmônica ajuda no desvio dos obstáculos estáticos. Entretanto, tal campo vetorial pode ser insuficiente para evitar colisões. Uma estratégia que coloca partículas virtuais nas fronteiras dos obstáculos é proposta para garantir a ausência de colisões. Os termos de viscosidade artificiais associados às partículas virtuais são responsáveis por garantir esta ausência. Simulações computacionais ideais, simulações realísticas e experimentos utilizando robôs reais foram executados para mostrar a eficácia do método.

No caso de tarefas de cobertura sensorial de ambientes, uma propriedade muito desejada é otimalidade. Entretanto, é difícil provar que os controladores propostos baseados em HPS conduzem os robôs para uma configuração ótima para cobertura. Logo, foram investigadas novas ferramentas para garantir otimalidade. Neste trabalho, foram utilizadas ferramentas da otimização locacional para obter leis de controle ótimas e distribuídas. Esta estratégia também mantém uma das principais funcionalidades da abordagem baseada em fluidos que é a habilidade de controlar a densidade dos robôs sobre uma dada região. Foram incorporadas três novas extensões a trabalhos encontrados na literatura [Lloyd, 1982, Cortez et al., 2004] para considerar: (i) sensores com campos de visão circulares de diferentes raios; (ii) robôs com forma geométrica circular; e (iii) ambientes poligonais não-convexos. As extensões são baseadas no uso de diferentes funções de distância, *power distance* e distância geodésica, e na incorporação de restrições para evitar colisões.

Ambas as técnicas, HPS e otimização locacional, são descentralizadas uma vez que apenas informações locais são requeridas. Isto é importante para garantir escalabilidade. Na primeira técnica cada agente necessita de infomação de agentes localizados dentro de um dado raio de distância. Na segunda técnica cada agente busca informações de seus vizinhos de Voronoi.

É importante também ressaltar que a abordagem baseada em HPS necessita de menos recursos computacionais. Para cada agente, apenas um simples somatório que itera sobre os agente vizinhos é executado. Nos controladores baseados em otimização locacional, cada agente precisa calcular o diagrama de Voronoi e depois realizar integrações numéricas sobre a célula de Voronoi correspondente. Por outro lado, a propriedade de otimalidade é inerente à otimização locacional. Contudo, deve-se ressaltar que é garantida apenas otimalidade local. Geralmente, a função objetivo é não-linear e não-convexa. Logo, pode-se concluir que cada abordagem tem suas vantagens e desvantagens e a escolha de uma delas para uma aplicação será fortemente dependente do contexto.

Contents

List of Figures

List of Tables

List of Symbols

$< f(\mathbf{x}) >$	Approximation of $f(\mathbf{x})$, page 34
β	Normalization constant, page 26
$\delta(\mathbf{x} - \mathbf{x}')$	Dirac delta function, page 34
η^2	Term to avoid singularities in the artificial viscosity, page 38
γ	Ratio of specific heats, page 37
∞	Infinite, page 34
$\ddot{\mathbf{q}}$	Robot acceleration, page 17
$\dot{\mathbf{q}}$	Robot velocity, page 16
$\mathbf{f}$	Sum of external forces normalized by mass, page 38
$\mathbf{g}$	Nonlinear vector function, page 19
$\mathbf{M}$	Finite element coefficient matrix, page 32
$\mathbf{q}$	Robot configuration, page 11
$\mathbf{q}^0$	Initial configuration, page 22
$\mathbf{q}^d$	Desired configuration, page 22
$\mathbf{s}$	Finite element source vector, page 32
$\mathbf{u}$	Control input vector, page 16
$\mathbf{v}$	Velocity, page 37
$\mathbf{y}$	State vector, page 19
$\mathcal{C}$	Configuration space, page 11

$\mathcal{C}_{obst}$	Set of obstacle configurations, page 12
$\mathcal{F}$	Free configuration space, page 14
$\mathcal{H}$	Coverage functional, page 93
$\mathcal{O}$	Set of obstacles in a workspace, page 9
$\mathcal{P}$	Set of n points $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$, page 40
$\mathcal{W}$	Robots' workspace, page 9
$\nabla\cdot$	Divergent operator, page 37
∇	Gradient operator, page 22
∇^2	Laplacian operator, page 26
Ω	Domain of solution of Laplace's equation, page 26
ω	Robot angular velocity, page 19
Ω_d	Target region, page 24
Ω_e	Domain inside a mesh element, page 30
$\partial/\partial t$	Partial time derivative, page 37
$\partial\Omega$	Solution domain boundary, page 26
$\Phi(\mathbf{q})$	Velocity field, page 26
$\phi(\mathbf{q})$	Potential function, page 22
ϕ^h	Approximate Laplace's equation solution, page 30
ϕ_l^h	Harmonic function value at a mesh node l, page 30
Π_{ij}	Artificial viscosity, page 38
ψ_l	Interpolation function used in the finite element method, page 30
ρ	Density, page 36
ρ_0	Reference density, page 39
σ	Positive constant related to the mass of dynamic obstacles, page 63

θ	Robot's orientation, page 11
Υ	Integration volume, page 34
φ	Distribution density function, page 93
ξ_1	Viscosity constant, page 38
ξ_2	Viscosity constant, page 38
$\{\eta\}$	Set of all nodes in a finite element mesh, page 30
$\{\eta_g\}$	Set of mesh nodes that lie at the Dirichlet boundary, page 33
$\{W\}$	Reference frame fixed in $\mathcal{W}$, page 9
${}^{W}\mathbf{R}_{G}$	Rotation matrix that transforms the components of vectors in $\{W\}$ into components in $\{G\}$, page 11
a_i	Kinematic constraint, page 15
c	Sound speed, page 38
D/Dt	Total time derivative, page 37
e	Internal energy per unit of mass, page 37
$f(d)$	Sensing degradation function, page 94
g	Gravitational constant, page 40
H	Maximum fluid depth, page 40
h	Smoothing length, page 35
h_i	Holonomic constraint, page 14
l_{ij}	Voronoi bisector, page 41
m	Mass, page 35
O_i	i^{th} obstacle in a set of obstacles, page 9
P	Pressure, page 37
PV	Power diagram, page 99
PV_i	Power region, page 99

$SE(m)$	Special Euclidian group in m dimensions, page 12
T	Tessellation of the environment, page 93
t	Time, page 19
v	Robot linear velocity, page 19
$V(\mathcal{P})$	Voronoi tessellation of set $\mathcal{P}$, page 41
V_i	Voronoi cell of site i, page 41
$W(\mathbf{x} - \mathbf{x}', h)$	Smoothing kernel function, page 34
x	x Cartesian coordinate, page 9
y	y Cartesian coordinate, page 9
BVP	Boundary Value Problem, page 27
FEM	Finite Element Method, page 5
SPH	Smoothed Particle Hydrodynamics, page 5

Chapter 1

Introduction

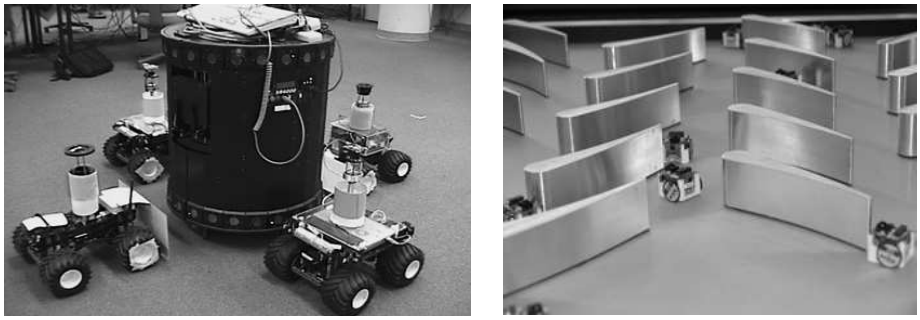
Cooperative robotics is the field dedicated to study techniques that allow robots in a team to cooperate among them and with humans in order to execute a given task. For a great variety of tasks, cooperative robotic systems give solutions that could not be obtained by a single robot. Moreover, even in situations where a single powerful robot could be used, the use of teams of robots may be less expensive, more reliable, more fault-tolerant, and more flexible.

Controlling groups of robots have been a challenge with several proposed solutions. The size of the group is a crucial factor that determines the most suitable approach to use. Basically, most approaches can be categorized into *centralized* or *decentralized* approaches. Centralized approaches assume the existence of a central entity which is able to plan actions for each robot and also to obtain information of the whole group in order to perform the required task in an optimal way. Although such approaches, in general, guarantee completeness of the task, they are not scalable to large groups of robots due to computational limitations. On the other hand, decentralized approaches may use a divide-and-conquer strategy to provide more scalable solutions. In fact, decentralized approaches advocate that each robot should be responsible for planning its own actions based only on the local information available.

Following the ideas of scalable solutions, a novel paradigm named *swarm* of autonomous agents has arisen in the robotics community. In this paradigm the objective is to control very large groups (tens to hundreds) of very simple robots. The key idea in this paradigm is that the success in the execution of a given task depends on the behaviors which emerge from the interactions among agents and between the agents and the environment. In fact, the robotic agents by themselves should be as simple as possible with limited capacity of communication, sensing, and actuation. Another important characteristic of a swarm is flexibility. By using different coordination mechanisms the same swarm should be able to tackle different tasks. Besides, the coordination of the swarm should not rely on specific members of the group. Therefore, totally decentralized methodologies where all agents can be considered anonymous (*i.e.* there is no need to uniquely identify the agents) and can be programmed with the same piece of code must be provided. Furthermore, those methodologies should be robust to dynamic deletion and addition of new robots.

1.1 Motivation

Nowadays, swarms and cooperative robotics are current topics discussed in the academy. Scientists all over the world are currently investigating the theoretical bases of these fields and also the technologies that may provide the necessary tools to make real world applications possible. Most works are still based only on simulations, but efforts have been done to implement the devised techniques in real robots. Figure 1.1 presents two examples of such efforts: Figure 1.1 (a) shows a snapshot of an experiment presented in [Pereira, 2003] where four small robots had the mission of caging a bigger circular robot. A possible application of such caging approach is the



(a) (b)

Figure 1.1: Experiments with real robots. (a) Four robots caging a circular robot [Pereira, 2003]. (b) Distributed boundary coverage with a swarm of miniature robots [Correll, 2007].

manipulation of objects. Figure 1.1 (b) shows a snapshot of an experiment conducted at EPFL (Ecole Polytechnique Fédérale de Lausanne). In this experiment, a task of boundary coverage of a regular structure was executed by a swarm of miniature robots. This task was motivated by a jet turbine inspection application and the experiment was conducted in an arena with 25 blades distributed in a regular pattern, mimicking the rotor and stator blades in a turbine [Correll, 2007].

Although this thesis is dedicated to develop novel theoretical approaches for swarm coordination, it is highly motivated by the wide range of real world applications in which swarm robotics could be successfully used. Some examples of applications are: search and rescue operations in hazardous environments or in places where humans cannot have access, contention of oil spills in the ocean, transport of heavy objects, environment monitoring, surveillance, etc. Moreover, if one thinks about the revolution which will come with the development of nanotechnology, then the possibilities of applications of swarm robotics are unlimited. One could think for example of millions of nanorobots being injected into a human to combat cancer cells. A swarm of nanorobots could also be useful in the construction and manipulation of

other nanostructures. However, to have well succeeded applications in the future, we believe that the theoretical foundations, which include adequate coordination algorithms, should be well established at that time. The focus of this work is the development of decentralized strategies to control very large groups of robots.

1.2 Methodology

To control swarms of robots, we first propose a novel scalable solution which is able to handle obstacles. We model the swarm as a fluid which may be subjected to external forces. The main motivation stems from the fact that a great variety of characteristics desirable for a group of robots can be observed in fluids. Some examples of such characteristics are: (i) fluids are easily deformed, (ii) fluids can easily contour objects, and (iii) the flow field variables and also the fluid phase can be easily manipulated in order to achieve desired behaviors. In our approach the *Smoothed Particle Hydrodynamics* (SPH) method is applied to model the "robotic fluid". More specifically, to model the interactions among the robots of the group. The SPH method was originally proposed in [Lucy, 1977, Gingold and Monaghan, 1977] to solve astrophysics problems, but nowadays this method is extensively used to solve problems in the field of fluid dynamics, mainly because of its ability to easily handle moving boundaries and large deformations [Liu and Liu, 2003]. The Finite Element Method (FEM) [Ida and Bastos, 1992] is also used in this work to compute a vector field which models external forces applied to the fluid. Since obstacles may have generic geometries, the use of FEM allows for efficiency in the field computation. By means of a weak coupling between FEM and SPH we derive decentralized controllers for the robots.

SPH is a mesh-free, Lagrangian, and particle numerical method. SPH equations are derived from the continuum governing equations by interpolating from a set of disordered particles. In fluid dynamics problems, each particle represents a small volume of the fluid and the interpolation is performed by using differentiable interpolation kernels which approximate a delta function. The continuum equations are converted to a set of ordinary differential equations, where each one controls the evolution of an attribute of a specific particle. We assume that each robot of the team is a SPH particle, and since we use kernels with compact support, it is possible to derive decentralized control laws based on the SPH equations. The derived controllers are decentralized in the sense that only local information is necessary: the external field at the location of the robot (computed using FEM) and position and velocity of neighboring robots. One may argue that the external field can only be computed if the map of the environment is known and then consider this knowledge as global information. Assuming that each robot may have its own version of the map of the environment, each robot is able to compute its own version of the external field. Therefore, in this point of view this information may be considered local. In this fluid based approach, neighboring robots are those located within a pre-specified range.

A limitation of the fluid based approach is verified in the solution of an environment coverage task by mobile sensor networks. This limitation is the fact that it is difficult to verify optimal coverage for the fluid based solution and this is a very desirable property for this task.

In order to address the problem of optimal environment coverage by mobile sensor networks, strategies derived from the Locational Optimization theory [Okabe et al., 2000] are considered. More specifically, we consider the distributed approach devised in [Cortez et al., 2004]. In this work, decentral-

ized controllers for identical mobile point sensors operating in convex environments are proposed. These controllers are decentralized in the sense that they rely only on the information of position of the robot and of its immediate neighbors. Neighbors, in this case, are defined to be those robots that are located in neighboring Voronoi cells [Okabe et al., 2000]. In the present thesis, we incorporate three novel and important extensions into the previous work [Cortez et al., 2004] to address: (i) sensors with circular footprints of different radii, (ii) disk-shaped robots, and (iii) nonconvex polygonal environments. The extensions are based on the use of different distance functions, power distance and geodesic distance, and the incorporation of constraints to allow collision avoidance. These extensions are important since they make this approach more feasible in real world applications, where sensors may be heterogeneous, robots have finite size, and environments may have more complex geometries.

It is also important to mention that in all the proposed controllers it is not necessary to label the robots. Therefore, all the robots run the same software and the success of the task execution does not depend on specific members of the group.

1.3 Contributions

This work has contributed to the area of swarm robotics with novel scalable techniques to swarm coordination. The major contributions are:

- Decentralized controllers by using a weak coupling between two numerical methods: the Smoothed Particle Hydrodynamics (SPH) and the Finite Element Method (FEM). The proposed controllers rely only on local information. Besides, all the robots can be considered as anony-

mous entities. An efficient treatment of obstacles is also part of the proposed methodology. (Chapter 4)

- A solution for the pattern generation task using incompressible fluid models. The swarm is modelled as an incompressible fluid subjected to external forces. Since the robots try to keep the density of the fluid model constant, this approach allows for a loose way of controlling the group connectivity. Actual robot issues such as finite size and nonholonomic constraints are also addressed. Collision avoidance guarantees are discussed. In the absence of obstacles, we prove for the first time stability and convergence of controllers based on the SPH. (Chapter 4) [Pimenta et al., 2006b, Pimenta et al., 2006c, Pimenta et al., 2007a, Pimenta et al., 2008b]

- A solution for a constrained coverage problem. The incompressible fluid model is applied and the team of robots is guided to maximize the coverage in generic unknown environments while keeping the density constant. (Chapter 4)

- Three novel and important extensions to a basic approach for optimal coverage by mobile sensor networks [Cortez et al., 2004]: (i) incorporation of heterogeneity in the robot team by allowing the robots to have different types of sensors and considering Power diagrams [Aurenhammer, 1987] to define the dominance region of each robot ; (ii) solution for the practical limitations of the point robot assumption in the original algorithm by introducing a constrained minimization problem; and (iii) generalization to nonconvex environments with the introduction of geodesic Voronoi diagrams [Aronov, 1989] in the control laws. (Chapter 5) [Pimenta et al., 2008a]

1.4 Organization

This document is organized in six chapters. In this chapter we presented a brief introduction to the main theme that we discuss in the rest of the thesis: *swarm robotics*. A brief overview of the nomenclature and the tools we use in this work is provided in Chapter 2. Chapter 3 reviews the related works found in the literature. Chapter 4 details the proposed fluid based approach for controlling swarms of robots. In Chapter 4, we instantiate our approach in a pattern generation task and also in a constrained coverage task. We present some simulation results, mathematical guarantees and experimental results. Chapter 5 considers a solution for the optimal coverage task based on Locational Optimization tools. Concluding remarks are discussed in Chapter 6.

Chapter 2

Background

This chapter intends to familiarize the reader with the nomenclature and the tools used throughout this text.

2.1 Workspace

The workspace, $\mathcal{W}$, of a robot R is the region of the world which can, in principle (if there is no obstacle), be reached by the robot. In the case of mobile robots, we usually have two-dimension ($2D$) and three-dimension ($3D$) workspaces. A wheeled mobile robot navigating constrained to a plane, for example, has a 2D workspace which is composed by the whole plane (x, y). On the other hand, an Unmanned Aerial Vehicle (UAV) navigating in the sky, for example, has a $3D$ workspace (x, y, z).

Usually, there are some forbidden regions inside the workspace which are called obstacles. Due to computational issues, it is common to model the set of obstacles in a workspace as a set of polygons ($2D$) or polyhedra ($3D$), $\mathcal{O} = \{O_1, \ldots, O_n\}$. Figure 2.1 shows a typical two-dimensional workspace with a set of obstacles. This figure also shows the coordinate system, commonly called world reference frame $\{W\}$, which is used to localize obstacles and robots inside the workspace.

In the general case, there are two types of obstacles in a workspace:

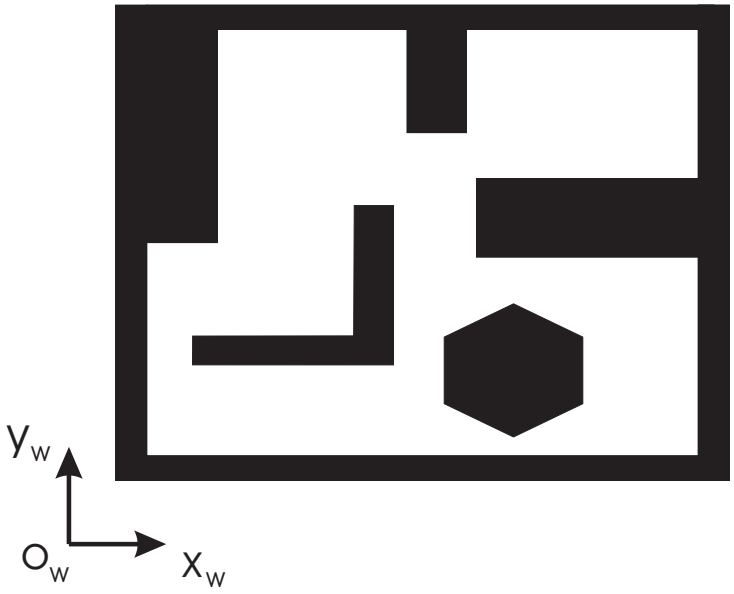


Figure 2.1: Typical mobile robot workspace.

static obstacles and *dynamic obstacles*. The former type corresponds to those obstacles that have fixed shapes, positions, and orientations over time. On the other hand, the latter type corresponds to those obstacles that have time-varying shapes, positions, and/or orientations. Normally, walls, trees, buildings, holes, and furniture are examples of static obstacles, while animals, other robots, and people are examples of dynamic obstacles.

In practice, it is not common to work directly with workspaces since collision detection for generic shaped robots is a difficult and time consuming operation. In the next section, we present a representation tool that allows the transformation of a workspace into a space where a robot with generic shape can be considered as a point.

2.2 Configuration Space

Given a robot R, a workspace $\mathcal{W}$, and a world reference frame $\{W\}$, we define a *configuration*, $\mathbf{q}$, as the vector of minimum dimension capable of completely characterize the robot's *placement* in the workspace. For example, consider a mobile robot navigating in a planar surface, and a reference point G which is fixed on the robot. The robot's configuration $\mathbf{q} = [x, y, \theta]^T$ is composed by the x and y coordinates of G and the robot's orientation θ. Then, by definition, the configuration space, $\mathcal{C}$, is the set of all possible configurations of the robot, while the robot's trajectory is a continuous sequence of configurations in $\mathcal{C}$. Figure 2.2 shows these concepts. The number of dimensions of the configuration space corresponds to the number of *degrees of freedom* of the robot. Clearly, in Figure 2.2 we have three degrees of freedom.

In the case of mobile robots navigating in three-dimensional workspaces, the configuration space has 6 dimensions, and the robots configurations may be represented by $\mathbf{q} = [x, y, z, \alpha, \beta, \theta]^T$. The first three dimensions correspond to the cartesian coordinates of the reference point G, and the other three are *Roll*, *Pitch*, and *Yaw* angles. These angles define rotations around the axes X, Y, and Z respectively of a body fixed frame.

Let $\{G\}$ be a reference frame fixed at point G. Each configuration corresponds to a homogeneous transformation matrix that converts points represented in $\{G\}$ to points represented in $\{W\}$. Such matrix can be written as:

$$ {}^{W}\mathbf{T}_{G} = \begin{bmatrix} {}^{W}\mathbf{R}_{G} & {}^{W}\mathbf{r}_{O} \\ \mathbf{0}_{1\times m} & 1 \end{bmatrix} , \tag{2.1} $$

where ${}^{W}\mathbf{R}_{G}$ is a rotation matrix that transforms the components of vectors in $\{G\}$ into components in $\{W\}$, ${}^{W}\mathbf{r}_{O}$ is the position vector of the origin of $\{G\}$ with respect to $\{W\}$, and m is the dimension of the rotation matrix.

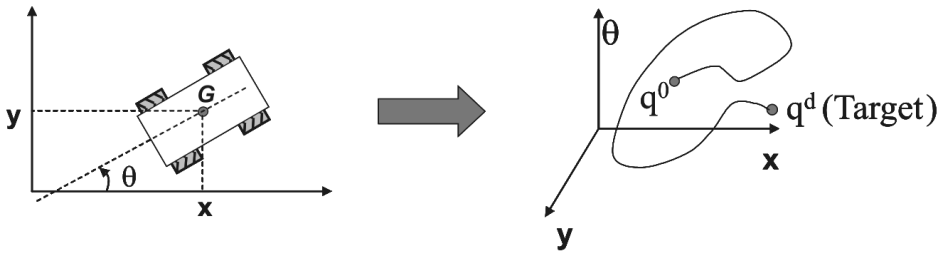


Figure 2.2: A robot is represented by a point in its configuration space. A trajectory is then a continuous sequence of configurations that starts at $\mathbf{q}^0$ (the initial robot's configuration) and ends at $\mathbf{q}^d$ (the desired final configuration) [Pereira, 2003].

The set of all transformation matrices in m dimensions defines the so-called *special Euclidean group*, $SE(m)$ [Murray et al., 1994]:

$$SE(m) = \left\{ \mathbf{T} \in \mathbb{R}^{(m+1)\times(m+1)} | \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{0}_{1\times m} & 1 \end{bmatrix} \right\}, \tag{2.2}$$

where $\mathbf{R} \in \mathbb{R}^{m\times m}$, $\mathbf{r} \in \mathbb{R}^m$, $\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}$, and $\det(\mathbf{R}) = 1$. The matrix $\mathbf{I}$ corresponds to the identity matrix and $\det(\mathbf{R})$ corresponds to the determinant of $\mathbf{R}$.

Therefore, in the case of robots navigating in a planar surface we can say that $\mathbf{q} \in SE(2)$. This nomenclature just indicates that a configuration $\mathbf{q} = [x, y, \theta]^T$ is equivalent to translations and rotations in $2D$. Similarly, in the case of the robot navigating in three-dimensional workspaces we can say that $\mathbf{q} \in SE(3)$.

The obstacles are represented in the robot's configuration space as a set of forbidden configurations, $\mathcal{C}_{obst}$. The computation of the robot's configuration space is usually performed by constructing the $\mathcal{C}_{obst}$. This is done by growing the obstacles by the size of the robot. Figure 2.3 shows the result of growing a rectangular obstacle by the size of a triangular robot which is

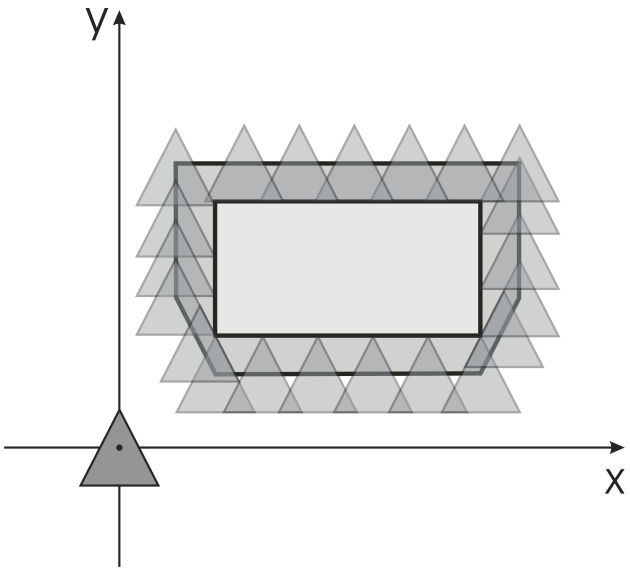


Figure 2.3: Result of the growth of a rectangular obstacle by the size of a triangular robot with constant orientation [Pimenta, 2005].

not allowed to rotate. In this case, it is clear that the configuration space is two-dimensional. From Figure 2.3, we can conclude that it is possible to avoid collisions between the robot and the obstacle if we are able to limit the excursion of the point used to construct the $\mathcal{C}_{obst}$ in the configuration space.

When $SE(2)$ or $SE(3)$ spaces are considered, the complexity of the configuration spaces increases. The efficient computation of the robot's configuration space can be done by using *Minkowski Sums* [de Berg et al., 2000]. This technique was implemented in [Pimenta, 2005] and will be omitted here, since it is out of the scope of this thesis.

The main advantage of developing solutions for robotic problems in the configuration space is that the robot can be considered as a point, since its shape is considered only during the construction of the space. The subset of the configuration space formed by collision-free configurations is referred to

as the free configuration space, $\mathcal{F}$ [Latombe, 1991]:

$$\mathcal{F} = \mathcal{C} \setminus \mathcal{C}_{obst}\,, \tag{2.3}$$

Obviously, the objective of most approaches in robotics is to control the robot such that it never leaves $\mathcal{F}$. An interesting approach that guarantees such condition is based on *Navigation Functions* which is presented later in this chapter.

2.3 Motion Constraints

It is very common that mechanical systems have their motion subjected to constraints. These constraints may arise from the structure of these mechanisms, or from the way in which they are actuated and controlled. We will consider here only constraints that may be expressed as equalities and that are independent of time. Such constraints are called *bilateral scleronomic* constraints [Luca and Oriolo, 1995].

Consider a n-dimensional configuration space, $\mathcal{C}$. Constraints are called *holonomic* if they may be put into the form

$$h_i(\mathbf{q}) = 0\,, \tag{2.4}$$

where $i = 1, \ldots, k < n$. For convenience, the functions $h_i : \mathcal{C} \rightarrow \mathbb{R}$ are assumed to be smooth and independent.

Holonomic constraints are actually geometric constraints. The effect of such constraints is to limit the configurations that are reachable by the robot. In fact, holonomic constraints reduce the number of dimensions of the configuration space. In this case, it is convenient to describe the system in terms

of $n-k$ new coordinates that represent the actual degrees of freedom of the system.

Constraints that involve configurations, $\mathbf{q}$, and velocities, $\dot{\mathbf{q}}$, are named *Kinematic constraints*. These constraints may be written as:

$$a_i(\mathbf{q}, \dot{\mathbf{q}}) = 0\,, \tag{2.5}$$

where $i = 1, \ldots, k < n$. If it is possible to put them into the form

$$\mathbf{a}_i^T(\mathbf{q}) \cdot \dot{\mathbf{q}} = 0\,, \tag{2.6}$$

then these constraints are referred to as *Pfaffian constraints*. Conveniently, the vector functions $\mathbf{a}_i : \mathcal{C} \to \mathbb{R}^n$ are assumed to be smooth and linearly independent.

Holonomic constraints may always be put in the Pfaffian form by writing $\mathbf{a}_i^T = \partial h_i / \partial \mathbf{q}$. However, the converse may not be true. Kinematic constraints that are not *integrable*, *i.e.*, that cannot be put into the form (2.4) are called *nonholonomic constraints*.

The effect of nonholonomic constraints is completely different from the effect of the holonomic ones. If a system is subjected only to k nonholonomic constraints then the number of dimensions of the configuration space is preserved. On the other hand, the instantaneous system mobility is restricted to a $(n-k)$-dimensional subspace. In fact, such constraints limit the velocities that can be imposed to the system in a given instant of time.

Next, we give examples of mobile robots subjected to holonomic and nonholonomic constraints.

2.3.1 Holonomic Mobile Robot

Holonomic mobile robots are those mobile robots which are subjected only to holonomic constraints. Therefore, the number of dimensions of the configuration space of these robots is reduced. On the other hand, there is no restriction on their instantaneous velocities in the reduced configuration space. This means that these robots can move in any desired direction in the reduced configuration space for all instants of time.

Consider a point robot in a $3D$ space that is only allowed to move in a planar surface. This surface defines a holonomic constraint for this robot. Suppose we place a cartesian reference frame with its origin located on the plane and with its z-axis perpendicular to the plane. Thus, it is clear that the plane may be completely described by the x and y coordinates and the holonomic constraint is given by $z = 0$. In the Pfaffian form:

$$[0, 0, 1] \cdot \dot{\mathbf{q}} = [0, 0, 1] \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = 0 \,. \tag{2.7}$$

If the robot allows actuation directly in its velocities and (2.7) is the only constraint imposed, then this robot may be modelled by:

$$\dot{\mathbf{q}} = \mathbf{u}(\mathbf{q}, t) \,, \tag{2.8}$$

where now $\mathbf{q} = [x, y]^T$, and $\mathbf{u}(\mathbf{q}, t)$ is the control input vector which may

depend or not of time, t:

$$\mathbf{u}(\mathbf{q},t) = \begin{bmatrix} u_1(\mathbf{q},t) \\ u_2(\mathbf{q},t) \\ 0 \end{bmatrix}. \tag{2.9}$$

If the robot provides acceleration inputs, then:

$$\ddot{\mathbf{q}} = \mathbf{u}(\mathbf{q},\dot{\mathbf{q}},t). \tag{2.10}$$

In this case:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{v}\,, \\ \dot{\mathbf{v}} &= \mathbf{u}. \end{aligned}$$

None of the models constrain the direction of movement on the plane.

2.3.2 Nonholonomic Robot

Differential drive is probably the simplest possible drive mechanism for a ground-contact mobile robot [Dudek and Jenkin, 2000]. This is a quite popular technology and several commercial robots, such as Khepera II and III [Khepera, 2007] and Pioneer P3-DX [Pioneer, 2009], use it. A differential drive robot consists of two actuators mounted on a common axis controlled by independent motors. It is very common to assume that these actuators cannot move in the lateral direction (see Figure 2.4). Therefore, such constraint limits the instantaneous direction of movement of the robot and can be classified as a nonholonomic constraint. In order to enforce this nonholonomic robot to make curves, one must impose different speeds on the two

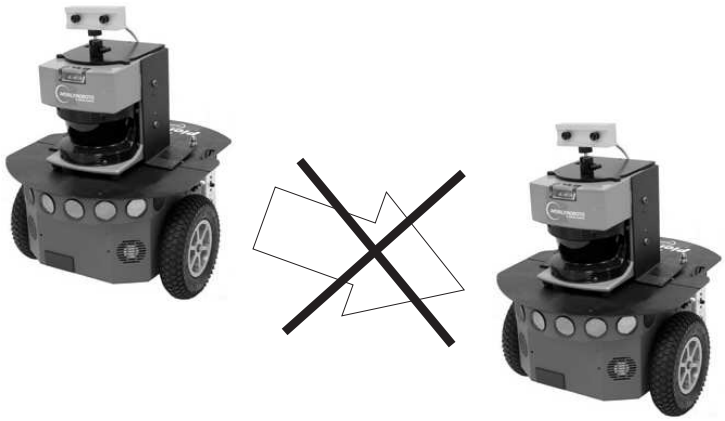


Figure 2.4: Nonholonomic constraint in differential drive robots.

motors.

If the robot is also subjected to the holonomic constraint mentioned before, where the robot is confined to a planar surface, then we can assume $\mathbf{q} = [x, y, \theta]^T$ and it is possible to characterize the nonholonomic constraint of non-sideslipping using the following expression:

$$[\sin(\theta), -\cos(\theta), 0] \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0\,, \tag{2.11}$$

where θ is the robot orientation.

A kinematic model for this robot may be derived by considering the null space of the matrix $A^T = [\sin(\theta), -\cos(\theta), 0]$:

$$\mathcal{N} = \text{span}\left\{ \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{2.12}$$

The model is then given by

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{2.13}$$

where v is the robot's linear velocity and ω is the robot's angular velocity. Since the relation between the speeds of the motors and v and ω is given by simple invertible algebraic expressions [Dudek and Jenkin, 2000], we can consider v and w as real control inputs. In fact, most commercial robots provide access only to these two variables.

2.4 Lyapunov Stability Theory

In this section we review some features of the *Lyapunov Stability Theory* that will be used in the convergence proofs presented in Chapter 4. All definitions are given according to [Slotine and Li, 1991].

We will represent, in general, a nonlinear dynamic system by a set of n nonlinear differential equations in the form:

$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}, t), \tag{2.14}$$

where $\mathbf{g}$ is a $n \times 1$ nonlinear vector function, $\mathbf{y}$ is the $n \times 1$ state vector, and t is time. A solution of equation (2.14), $\mathbf{y}(t)$, corresponds to a curve in the state space. This curve is generally referred to as a *state trajectory* or a *system trajectory*.

Nonlinear systems are traditionally classified as either *autonomous* or *non-autonomous*.

Definition 2.1 (Autonomous and Non-autonomous systems) *The nonlinear system in (2.14) is said to be autonomous if* $\mathbf{g}$ *does not depend explicitly on time,* i.e.*, if the system's state equation can be written as*

$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}). \tag{2.15}$$

Otherwise, the system is called non-autonomous.

In this work we will concentrate on autonomous systems. Therefore, the further definitions in this section assume systems of this type. The first important definition refers to a special class of system trajectory which corresponds to only a single point. Such points are called *equilibrium points*.

Definition 2.2 (Equilibrium State or Equilibrium Point) *A state* $\mathbf{y}^*$ *is an equilibrium state (or equilibrium point) of the system if once* $\mathbf{y}(t) = \mathbf{y}^*$*, it remains equal to* $\mathbf{y}^*$ *for all future time. Mathematically, this means:*

$$\mathbf{g}(\mathbf{y}^*) = 0. \tag{2.16}$$

Equilibrium points can be classified as *stable* or *unstable*.

Definition 2.3 (Stable and Unstable Equilibrium Points) *The equilibrium point* $\mathbf{y}^*$ *is said to be stable if, for any* $\delta > 0$*, there exists* $\epsilon > 0$*, such that if* $\|\mathbf{y}(0) - \mathbf{y}^*\| < \epsilon$*, then* $\|\mathbf{y}(t) - \mathbf{y}^*\| < \delta$*,* $\forall t \geq 0$*. Otherwise, the equilibrium point is unstable.*

If besides being stable there exists some $\epsilon > 0$ such that $\|\mathbf{y}(0) - \mathbf{y}^*\| < \epsilon$ implies that $\mathbf{y}(t) \to \mathbf{y}^*$ as $t \to \infty$, then $\mathbf{y}^*$ is said to be *asymptotically stable*. The set of all points such that trajectories initiated at these points eventually converge to $\mathbf{y}^*$ is called the *domain of attraction* of $\mathbf{y}^*$.

A generalization of the concept of equilibrium points is that of *invariant sets.*

Definition 2.4 *A set $\mathcal{G}$ is an invariant set for a dynamic system if every system trajectory which starts from a point in $\mathcal{G}$ remains in $\mathcal{G}$ for all future time.*

It is interesting to note that any equilibrium point is an invariant set. In fact, the whole domain of attraction of an equilibrium point is an invariant set. The following theorem allows for devising proofs of convergence of autonomous systems.

Theorem 2.1 (Local Invariant Set Theorem) *Consider an autonomous system of the form (2.15), with* $\mathbf{g}$ *continuous, and let $V(\mathbf{y})$ be a scalar function with continuous first partial derivatives. Assume that*

- *for some $l > 0$, the region $\mathcal{G}_l$ defined by $V(\mathbf{y}) < l$ is bounded*
- *$\dot{V}(\mathbf{y}) \leq 0$ for all $\mathbf{y}$ in $\mathcal{G}_l$*

Let $\mathcal{N}$ be the set of all points within $\mathcal{G}_l$ where $\dot{V}(\mathbf{y}) = 0$, and $\mathcal{M}$ be the largest invariant set in $\mathcal{N}$. Then, every solution $\mathbf{y}(t)$ originating in $\mathcal{G}_l$ tends to $\mathcal{M}$ as $t \to \infty$.

Proof: Refer to [Slotine and Li, 1991]. ∎

In the above theorem, the word "largest" is understood as the union of all invariant sets within $\mathcal{N}$.

2.5 Navigation Functions

One of the most studied problems in the robotics literature is the *motion planning* problem. This problem can be defined as [Latombe, 1991]:

Definition 2.5 (Robot motion planning problem) *Let a single robot R in the world $\mathcal{W}$ be represented by the configuration $\mathbf{q} \in \mathcal{C}$, and consider $\mathcal{F} \subseteq \mathcal{C}$ to be the free configuration space for R. Steer the robot from its initial configuration $\mathbf{q}^0 \in \mathcal{F}$ at time $t = t_0$ to the desired configuration $\mathbf{q}^d \in \mathcal{F}$ at some time $t = t_f > t_0$, such that $\mathbf{q} \in \mathcal{F}\ \forall t \in [t_0, t_f]$.*

As pointed out in [Pereira, 2003], this problem consists of three basic subproblems: (i) computing the free configuration space, $\mathcal{F}$, by considering the obstacles in $\mathcal{W}$ as described in Section 2.2; (ii) generating a trajectory τ, which is a continuous sequence of configurations in $\mathcal{F}$ for R; and (iii) controlling the robot to follow τ. In [Latombe, 1991], several solutions to the motion planning problem are presented.

One of the most popular techniques to plan trajectories and control robots in their configuration spaces is the *Artificial Potential Field* approach [Khatib, 1986]. This approach is based on the computation of a scalar potential function, $\phi(\mathbf{q})$, which is designed to have a minimum at the goal location and maxima at the boundaries of obstacles. The key idea is to use the descent gradient, $-\nabla\phi(\mathbf{q})$, to drive the robot to the goal. The descent gradient may be treated as a virtual force that simultaneously repels the robot from the obstacles and attracts the robot to the goal. This approach gives all possible trajectories independently of the initial configuration, since these trajectories are determined by the integral curves of the vector formed by $-\nabla\phi(\mathbf{q})$.

Most potential field approaches present spurious local minima in the potential function, which causes problems in the convergence to the target [Latombe, 1991]. Figure 2.5 presents an example of a local minimum caused by a U-shaped obstacle. In this figure, since the attractive virtual force is equal to the repulsive virtual force at the point where the robot is

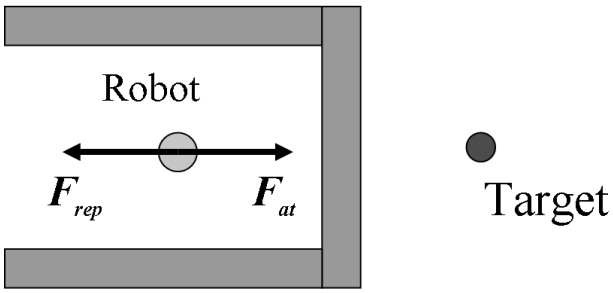


Figure 2.5: Local minimum caused by a U-shaped obstacle.

located, the robot is trapped inside the obstacle and never reaches the target.

As proved in [Koditschek, 1987], it is impossible to find a smooth non-degenerate vector field[1] on the configuration space of a point robot with m obstacles admitting a globally asymptotically stable equilibrium state. In other words, it is impossible to find a global potential function with integral curves given by the descent gradient converging to the goal from every initial configuration. However, it is possible to construct a potential function that is almost global, in the sense that the integral curves converge to the goal except from a set of measure zero[2] initial configurations. It should be clear that in the practical point of view such initial configurations are not problematic since they characterize saddle points. Therefore, they are unstable equilibrium points and any perturbation compels the system to abandon them and converge to the goal.

Rimon and Koditschek [Rimon and Koditschek, 1988] devised special artificial potential fields called *Navigation Functions*. Although saddle points are not prevented, these functions have a single minimum that coincides with $\mathbf{q}^d$. A procedure to construct these navigation functions and also the conditions of existence are given in [Rimon and Koditschek, 1992]. A potential

[1]The vector field Jacobian matrix has full rank at the equilibrium points.
[2]A set of measure zero can be thought of as a set with zero volume/area.

function ϕ defined over $\mathcal{F}$ with target Ω_d is characterized as a navigation function if [Pimenta, 2005]:

1. it is smooth on $\mathcal{F}$ (at least C^2)[3];
2. it has no other minima than the connected region Ω_d;
3. it is uniformly maximal on the boundary of $\mathcal{F}$ (*i.e.* ϕ has the same value, that corresponds to the maximum value, at all points of the boundary);
4. it is a Morse function (*i.e.* there is no degenerate critical point) [Milnor, 1963].

In comparison to the original definition given in [Rimon and Koditschek, 1992], a small adaptation is made in [Pimenta, 2005]. Originally, a navigation function is defined to be minimum only at a specific goal configuration $\mathbf{q}^d$ and not in a region Ω_d. This is changed in [Pimenta, 2005] due to practical reasons. In this way, it is guaranteed the robot stops at the goal even in the presence of small localization errors and noise, which are disturbances that cannot be neglected in real world.

In [Rimon and Koditschek, 1992], a recipe for constructing navigation functions is detailed. This recipe consists of both computing a navigation function, ϕ, over a generalized sphere world, $\mathcal{M}$, using a closed-form expression and finding a diffeomorphism[4], d, between $\mathcal{M}$ and the free configuration space, $\mathcal{F}$. Thus, by composing the navigation function, ϕ, with the diffeomorphism, d, it is possible to construct a navigation function over $\mathcal{F}$. In practice

[3] At least the second derivative must be continuous.

[4] A map between two spaces that is smooth, one-to-one and onto, and has a smooth inverse is called a diffeomorphism [Rimon and Koditschek, 1992].

this method has some limitations: (i) it is computationally expensive; (ii) it is difficult to be implemented for generic configuration spaces; and (iii) it is only applicable for a portion of all possible generalized sphere worlds.

Differently from the authors of [Rimon and Koditschek, 1992], who pursuit analytical functions, many authors have proposed different methodologies to construct numerical navigation functions [Konolige, 2000, Valavanis et al., 2000, Wang and Chirikjian, 2000]. The main drawback of most of these approaches is that they are based on regular grid discretization, which is not suitable to represent complex obstacles. As proposed in our previous works [Pimenta et al., 2005a, Pimenta, 2005, Pimenta et al., 2005b, Pimenta et al., 2006a], in this work we compute numerical navigation functions by using the *Finite Element Method* (FEM). In Chapter 4, we use these numerically computed navigation functions in the control of swarms of robots. Actually, as originally proposed in [Sato, 1987] and later popularized in [Connolly et al., 1990], we compute harmonic functions which are solutions to Laplace's equation. In [Charifa and Masoud, 2005], it is proved that it is possible to compute harmonic functions which fulfills all the navigation functions properties described before. In fact, specific boundary conditions must be imposed when solving the Laplace's equation to ensure such properties. In the next section, we show how to use harmonic functions as navigation functions.

2.6 Harmonic Functions

The Laplace's equation is given by:

$$\nabla^2 \phi = 0\,, \tag{2.17}$$

where ϕ is a harmonic function and ∇^2 is the Laplacian operator. Equation (2.17) is valid in the domain $\Omega \subset \mathbb{R}^n$. So far in this work, we have considered the function domain to be a two-dimensional space $\mathcal{F}$ (see Chapter 4).

A common type of boundary condition for the Laplace's equation is the Constant Dirichlet condition. Such condition is described as:

$$\phi_D = \phi|_{\partial\Omega} = V_c\,, \tag{2.18}$$

where V_c is a constant. This equation implies that the gradient of the harmonic potential will be perpendicular to the boundary where the condition is imposed.

Under the following conditions, the computed harmonic function has all the necessary properties presented in the previous section to be a navigation function:

- Target boundary conditions are Dirichlet with value V_c equal to zero;
- Obstacles boundary conditions are Dirichlet with identical positive V_c values.

In [Pimenta et al., 2005a], it is proved that, under these boundary conditions, the following velocity vector field guarantees convergence of its integral curves to the goal region Ω_d:

$$\Phi(\mathbf{q}) = \begin{cases} -\mathbf{G} \cdot \frac{\nabla\phi(\mathbf{q})}{\|\nabla\phi(\mathbf{q})\|^\beta} & \text{if } \nabla\phi(\mathbf{q}) \neq 0 \\ 0 & \text{if } \nabla\phi(\mathbf{q}) = 0 \end{cases}, \tag{2.19}$$

where $\|.\|$ is the euclidian norm operator, and $\mathbf{G}$ is a diagonal positive definite matrix used to scale the vector field to a robot compatible velocity value.

The constant $\beta \in \mathbb{N}_+$, when larger than 1, causes the velocity to be inversely proportional to the gradient norm. As the gradient norm is larger close to the goal region (for the domains we consider in this work), the velocity becomes lower in this region and the robot may reach the target with reduced speed. Inside Ω_d we have $\nabla\phi = 0$, and thus (2.19) explicitly makes the robot stop when the goal is reached.

As mentioned before, if a robot following the field in (2.19) started exactly at a saddle point it would not converge to the target. However, this fact has no practical value since any minimal disturbance, such as sensors and actuators noise, is able to force the robot to leave this point.

Therefore, if a holonomic robot with actuation in its velocities (see Equation (2.8)) is subjected to $\mathbf{u}(\mathbf{q}) = \Phi(\mathbf{q})$, then it is guaranteed that this robot will reach the target and stop there. Such result was also experimentally verified in [Pimenta et al., 2005a, Pimenta et al., 2006a]. Figure 2.6 shows a path described by an actual holonomic robot (the robot is constrained to a plane and can move laterally) in the presence of a U-shaped obstacle. In this test, it was used $\beta = 1$ which produces a constant velocity field which is represented in the figure by the arrows.

By using the Laplace's equation, we can consider the navigation function computation problem as a *Boundary Value Problem* (BVP). Finite element methods have been successfully used to solve BVP's on domains with generic geometries. The result on Figure 2.6 was obtained using the Finite Element Method (FEM). The triangles in the figure correspond to the elements used by this method. Since in this work we also intend to control robots in geometric complex environments, FEM is used. Next section is dedicated to present the main features of FEM.

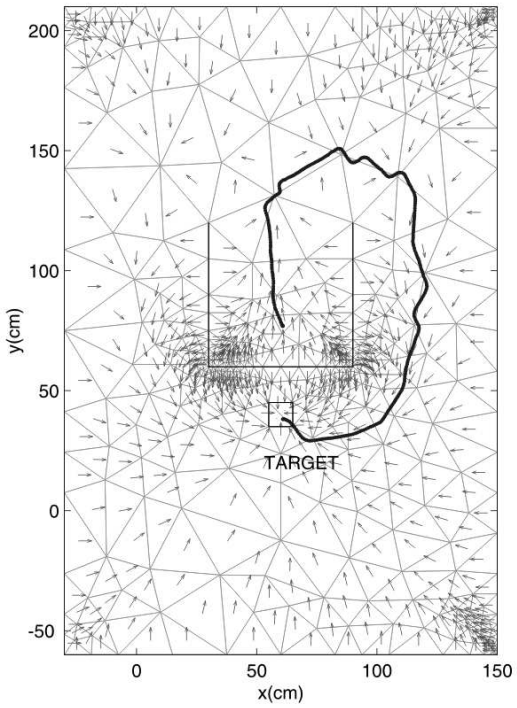


Figure 2.6: Path followed by an actual holonomic robot under the influence of a discretized harmonic field [Pimenta et al., 2006a].

2.7 Finite Element Method

The finite element method is a powerful numerical method which is widely used to solve partial differential equations. The main features of FEM in the solution of Laplace's equation are presented in this section. Further details may be obtained in [Hughes, 2000] and [Ida and Bastos, 1992].

2.7.1 Discretization

In order to apply the finite element method, the solution domain must be divided into small sub-domains. These sub-domains are called *Finite Elements*. Different element sizes can be used in this process of discretization, which makes this method powerful and flexible. In regions

where larger variations in the gradient of the computed function are observed, higher element densities are used. On the other hand, in regions with lower variations in the gradient, fewer elements may be used. This is important for reducing the computational complexity in the presence of complex domains. In other methods, such as the standard Finite Differences Methods and Resistive Grids, which have been applied in previous methodologies that use Laplace's equation solution for navigation [Connolly et al., 1990, Connolly, 1992, Valavanis et al., 2000], this kind of flexibility is difficult to implement.

Although several geometric shapes are possible, triangular elements are widely used for solving two-dimensional problems (see Figure 2.6). In this case, we define the intersection of edges as a *node* (or *vertex*). Also, the collection of elements forms a *mesh*.

The required discretization can be performed by *Delaunay Refinement* [Shewchuk, 1998], which is a technique for generating unstructured meshes of triangles for the use in the finite element method. This technique operates by maintaining a *Delaunay Triangulation*, which is refined by the insertion of additional vertices. The placement of these vertices is chosen to enforce boundary conformity and to improve the quality of the mesh. A Delaunay triangulation of a set of points, $\mathcal{P}$, is a set of triangles connecting the points satisfying an "empty circle" property: the circumcircle of each triangle does not contain any point of $\mathcal{P}$ in its interior (see Figure 2.7). Further details of mesh generation can be found in [Shewchuk, 1998] and [de Berg et al., 2000].

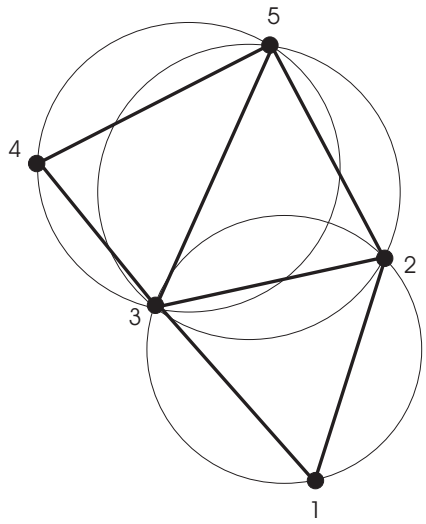


Figure 2.7: Delaunay triangulation of a set of 5 points in the plane.

2.7.2 Potential and Gradient Computation

The finite element method provides approximate solutions, ϕ^h, of Laplace's equation by means of interpolation:

$$\phi^h(\Omega) = \sum_{l \in \{\eta\}} \phi_l^h \psi_l \,, \tag{2.20}$$

where $\{\eta\}$ is the set of all mesh nodes, ϕ_l^h are the harmonic function values at the corresponding nodes l, and ψ_l are interpolation functions with compact support. These interpolation functions have three properties: (i) they are continuous; (ii) they have value equal to 1 at the corresponding node; and (iii) they have value equal to 0 at the elements that are not incident to the corresponding node. Therefore, to compute the potential value inside an element Ω_e, only the function values at the vertices of Ω_e are needed:

$$\phi^h(\Omega_e) = \sum_{j=1}^{3} \phi_j^h \psi_j. \tag{2.21}$$

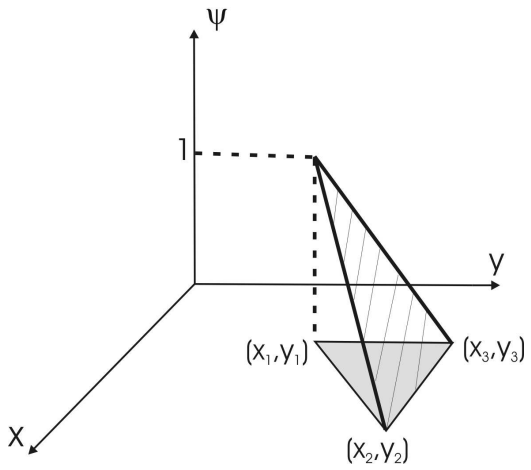


Figure 2.8: Linear interpolation function defined over a triangular element.

In this work we use linear elements. This means that our interpolation functions are first order polynomials. Figure 2.8 shows a linear interpolation function defined over a triangle. The reasons for using linear elements are computational efficiency and simplicity. The linear interpolation functions inside an element Ω_e with vertices 1, 2, and 3 are given by:

$$\psi_l(\mathbf{q}) = \begin{cases} \alpha_l + \beta_l x + \gamma_l y & \text{if } \mathbf{q} \in \Omega_e \\ 0 & \text{if } \mathbf{q} \notin \Omega_e \end{cases} , \tag{2.22}$$

where $l = 1, 2, 3$, and α_l, β_l, and γ_l are constants. Assuming these nodes are counterclockwise ordered, we have for ψ_1:

$$\alpha_1 = \frac{1}{2A}(x_2 y_3 - x_3 y_2) , \tag{2.23}$$

$$\beta_1 = \frac{1}{2A}(y_2 - y_3) , \tag{2.24}$$

$$\gamma_1 = \frac{1}{2A}(x_3 - x_2) , \tag{2.25}$$

and for ψ_2, and ψ_3 we have the same expressions with cyclic permutations of the indices $(1, 2, 3)$. The constant A corresponds to the triangle area.

An important conclusion is that the approximate function is continuous over the whole domain. From (2.22), it is clear that the function is continuous inside an element. The continuity over the edges, which are the interfaces between two elements, is guaranteed by the fact that ϕ^h over an edge depends only on the interpolation functions associated to the nodes incident to the edge. All the other interpolation functions are zero at that edge. Therefore, continuity over the whole domain is guaranteed.

Since we use linear elements, the gradient $\nabla\phi^h = [\frac{\partial\phi^h}{\partial x}\ ,\ \frac{\partial\phi^h}{\partial y}]^T$ is constant inside each element Ω_e:

$$\frac{\partial\phi^h}{\partial x}(\Omega_e) = \sum_{l=1}^{3}\beta_l\phi_l^h\,, \tag{2.26}$$

$$\frac{\partial\phi^h}{\partial y}(\Omega_e) = \sum_{l=1}^{3}\gamma_l\phi_l^h. \tag{2.27}$$

To use the expressions presented so far, we need to evaluate the harmonic function at the mesh nodes. This can be done by solving a linear system:

$$\mathbf{M}\cdot\phi = \mathbf{s}\,, \tag{2.28}$$

where $\mathbf{M}$ is a square, sparse, symmetric, and positive definite matrix, ϕ is the solution vector with dimension equal to the number of nodes that are not located at the Dirichlet boundaries, and $\mathbf{s}$ is the source vector where the boundary conditions appear. The elements of $\mathbf{M}$ and $\mathbf{s}$ are given by:

$$\mathbf{M}_{ab} = \int_\Omega(\nabla\psi_a\cdot\nabla\psi_b)d\Omega\,, \tag{2.29}$$

$$\mathbf{s}_a = -\sum_{v\in\{\eta_g\}}\left[\int_\Omega(\nabla\psi_a\cdot\nabla\psi_v)d\Omega\right]c_v\,, \tag{2.30}$$

where $a, b \in \{\eta - \eta_g\}$, $\{\eta\}$ is the set of all mesh nodes, $\{\eta_g\}$ is the set of mesh nodes that lie at the Dirichlet boundary, and c_v is the Dirichlet value assigned to the node $v \in \{\eta_g\}$. Due to the fact that the interpolation functions have compact support, several elements of the matrix and the vector are equal to zero. Details concerning the derivation of these expressions can be found in [Hughes, 2000].

It is important to mention that due to the interesting properties of matrix $\mathbf{M}$, it is possible to use efficient methods, such as the *Conjugate Gradient*, to solve the linear system in (2.28). The Conjugate Gradient method has computational complexity given by $O(m\sqrt{k})$, where m is the number of non null elements in $\mathbf{M}$, and k is the condition number of $\mathbf{M}$ [Shewchuk, 1994]. When a two-dimensional BVP is solved using FEM, we have $m = O(N)$ and $k = O(N)$, where N is the number of mesh nodes. Therefore, the computational complexity becomes $O(N^{3/2})$. If a pre-conditioner is used, the matrix condition number is reduced and an even lower complexity can be obtained. In this work we use the *Incomplete Cholesky Decomposition* as a pre-conditioner [Shewchuk, 1994], which makes the time of computation smaller than one second for all the configuration spaces we tested. In these tests, the computational platform was a laptop endowed with an Intel Mobile Pentium 4 (2.8 GHz), 512 MB DDR SDRAM, running Windows XP.

2.8 Smoothed Particle Hydrodynamics

The *Smoothed Particle Hydrodynamics* (SPH) is a *mesh-free particle* numerical method which was originally introduced in [Lucy, 1977] and [Gingold and Monaghan, 1977] to solve problems in astrophysics. It is a particle numerical method since it employs a set of finite number of dis-

ordered discrete particles to represent the state of the simulated system. It is mesh-free due to fact that it is not necessary to generate a mesh to provide connectivity of the particles as in the case of the finite element method. Besides, SPH is considered a *Lagrangian* method, which means that the particles are not fixed in space while the material is moving. The particles are actually attached to the material and move with the flow. Due to all of these characteristics, this method has been extensively used to solve fluid dynamics problems [Liu and Liu, 2003] where issues such as large deformation, moving interfaces between different materials, moving boundaries, and free surfaces appear very often. Most of the time, these issues are very troublesome to other numerical methods such as finite elements and finite differences.

SPH is based on the integral representation of a function:

$$f(\mathbf{x}) = \int_{\Upsilon} f(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}')d\mathbf{x}' \,, \tag{2.31}$$

where Υ is the volume that contains $\mathbf{x}$ and $\delta(\mathbf{x}-\mathbf{x}')$ is the Dirac delta function

$$\delta(\mathbf{x} - \mathbf{x}') = \begin{cases} \infty, & \mathbf{x} = \mathbf{x}' \\ 0, & \mathbf{x} \neq \mathbf{x}' \end{cases} \,, \tag{2.32}$$

where ∞ means infinite and such that

$$\int_{\Upsilon} \delta(\mathbf{x} - \mathbf{x}')d\mathbf{x}' = 1. \tag{2.33}$$

If the Delta function is replaced by a smoothing function $W(\mathbf{x} - \mathbf{x}', h)$ then the integral representation is approximated by

$$f(\mathbf{x}) \approx < f(\mathbf{x}) > = \int_{\Upsilon} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}', h)d\mathbf{x}' \,, \tag{2.34}$$

where $< f(\mathbf{x}) >$ is an approximation of $f(\mathbf{x})$, W is the so-called *smoothing kernel function* or simply *kernel* in the SPH literature, and h is the *smoothing length* that defines the influence area of W. The kernel is chosen to satisfy:

$$\int_{\Upsilon} W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = 1 \qquad \text{and} \qquad \lim_{h\to 0} W(\mathbf{x}-\mathbf{x}', h) = \delta(\mathbf{x}-\mathbf{x}'). \quad (2.35)$$

Besides, the smoothing function must be differentiable, since its gradient is needed in most applications. Usually, the function is chosen to be an even function and also to have compact support controlled by the parameter h. The spline kernel satisfies all these properties and is recommended by many authors such as [Monaghan, 1992]. In this work, we use cubic splines recommended for two dimensions:

$$W(\mathbf{r}, h) = \frac{10}{7\pi h^2} \begin{cases} 1 - \frac{3}{2}\kappa^2 + \frac{3}{4}\kappa^3 & \text{if} \quad 0 \leq \kappa \leq 1\,, \\ \frac{1}{4}(2-\kappa)^3 & \text{if} \quad 1 \leq \kappa \leq 2\,, \\ 0 & \text{otherwise}\,, \end{cases} \quad (2.36)$$

where $\kappa = \|\mathbf{r}\|/h$. It can be observed that the function support is determined by $2h$. Figure 2.9 shows the appearance of this function when it is centered at the origin and the parameter h is equal to 1.

The continuous integral in (2.34) can be converted to summation over all the N particles in the support domain of $\mathbf{x}$. This can be done by considering that a particle j has a finite volume ΔV_j which is related to the mass, m_j, of the particle by

$$m_j = \Delta V_j \rho_j\,, \quad (2.37)$$

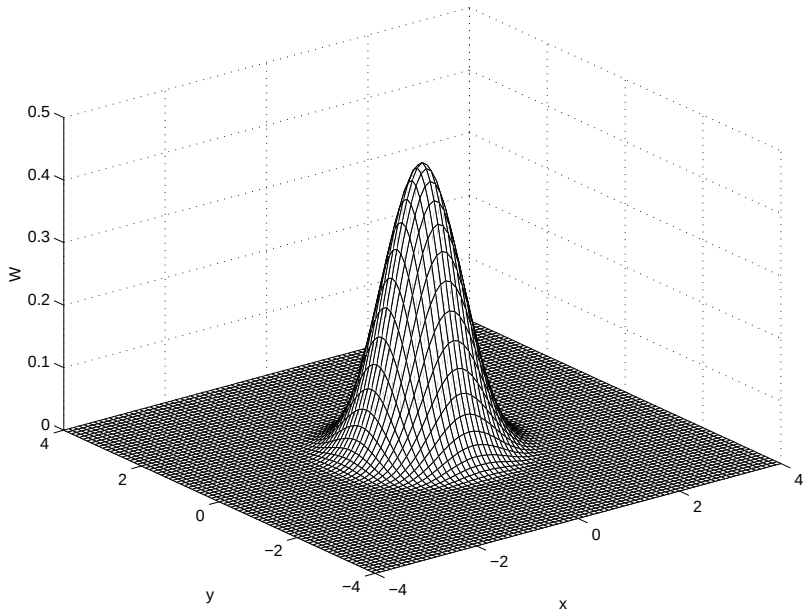


Figure 2.9: Graphic of the kernel function W with $h = 1$.

where ρ_j is the density of the particle. Thus, using ΔV_j instead of $d\mathbf{x}'$

$$< f(\mathbf{x}) > \approx \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) W(\mathbf{x} - \mathbf{x}_j, h). \tag{2.38}$$

The error in approximating the integral representation of a function by summations of the function evaluated at particle locations weighted by interpolation kernels depends on the disorder of the particles and is normally $O(h^2)$ or better [Monaghan, 1992].

Spatial derivatives of f, such as the gradient, can also be approximated. If integration by parts is used in the simplification process, it is possible to write the spatial derivative of f in terms of the gradient of the kernel

$$< \nabla_{\mathbf{x}} f(\mathbf{x}) > \approx \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) \nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}_j, h)\,, \tag{2.39}$$

where $\nabla_{\mathbf{x}}$ is the gradient taken with respect to $\mathbf{x}$.

It is interesting to observe that the particle approximation in (2.38) and (2.39) introduces mass and density into the equations. Since density is a key variable in hydrodynamic problems, this particle approximation can be conveniently applied in such problems. According to [Liu and Liu, 2003] this is probably one of the major reasons for the SPH method being popular for fluid dynamics problems. The continuum governing equations of fluid dynamics are three: (i) conservation of mass; (ii) conservation of momentum; and (iii) conservation of energy. For *inviscid compressible fluids*, in the absence of heat flux, these equations, in the Lagrangian description, are given by

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{v}\,, \tag{2.40}$$

$$\frac{D\mathbf{v}}{Dt} = -\frac{\nabla P}{\rho}\,, \tag{2.41}$$

$$\frac{De}{Dt} = -\left(\frac{P}{\rho}\right) \nabla \cdot \mathbf{v}\,, \tag{2.42}$$

where $\mathbf{v}$ is velocity, P is the hydrostatic pressure, and e is the internal energy per unit of mass. The operator D/Dt is the *total time derivative* that is physically the time rate of change following a moving fluid element. This derivative is composed by two factors: (i) the time fluctuation of the flow property itself, and (ii) the variation of the flow property due to the movement of the fluid element. Mathematically this is written as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + (\mathbf{v} \cdot \nabla). \tag{2.43}$$

An additional *equation of state* must be used to fully characterize a fluid. For many compressible fluids the model of an ideal gas can be used. In this

case, the following equation of state can be applied:

$$P = (\gamma - 1)\rho e\,, \tag{2.44}$$

where γ is the ratio of specific heats, a parameter which depends on the gas being simulated.

In the SPH method, the continuum equations of fluid dynamics are converted to a set of ordinary differential equations, where each one controls the evolution of an attribute of a specific particle. This conversion is performed by using the particle approximation method described before in the conservation equations. The resultant SPH conservation equations for particle i are:

$$\rho_i = \sum_j m_j W(\mathbf{x}_i - \mathbf{x}_j, h)\,, \tag{2.45}$$

$$\frac{d\mathbf{v}_i}{dt} = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij}\right) \nabla_i W_{ij} + \mathbf{f}_i\,, \tag{2.46}$$

$$\frac{de_i}{dt} = \frac{1}{2}\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij}\right) \mathbf{v}_{ij} \cdot \nabla_i W_{ij}\,, \tag{2.47}$$

where $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j)$, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, and $\mathbf{f}_i$ is the sum of external forces normalized by the mass m_i. It should be clear that e_i is only a component of the total energy of the system. The total energy also takes into account the kinetic energy and the energy related to the external forces. The term Π_{ij} is an artificial viscosity term added to handle shocks. There are several variants for this viscosity term. The most used one is given by [Monaghan, 1992]:

$$\Pi_{ij} = \begin{cases} \frac{1}{\bar{\rho}_{ij}}(-\xi_1 \bar{c}_{ij}\mu_{ij} + \xi_2 \mu_{ij}^2) & \text{if} \quad \mathbf{v}_{ij}\cdot\mathbf{x}_{ij} < 0\,, \\ 0 & \text{if} \quad \mathbf{v}_{ij}\cdot\mathbf{x}_{ij} > 0\,, \end{cases} \tag{2.48}$$

where

$$\mu_{ij} = \frac{h\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2 + \eta^2} \quad . \tag{2.49}$$

In (2.48), $\overline{\rho}_{ij}$ is the average between the densities of particles i and j, ξ_1 and ξ_2 are viscosity constants, $\overline{c}_{ij}$ is the average sound speed, and η^2 is a term added to avoid singularities. The term η^2 should be small enough to avoid severe smoothing of the viscosity term. Usually, this term is made equal to $0.01h^2$.

The sound speed of a particle i, which represents the speed at which sound travels through the fluid element represented by the particle, is given by:

$$c_i = \sqrt{\frac{\gamma P_i}{\rho_i}} \quad . \tag{2.50}$$

The motion of *incompressible fluids*, such as water, can also be simulated using the SPH equations. The key idea is to make a compressible fluid behave like a nearly incompressible one. This can be done by employing the equation of state below [Monaghan, 1994]:

$$P_i = B_i \left[\left(\frac{\rho_i}{\rho_0} \right)^{\gamma} - 1 \right] , \tag{2.51}$$

where ρ_0 is the reference density (1000Kg/m^3 in the case of water) and B_i is the bulk modulus [5]. The bulk modulus is a property which characterizes the compressibility of the fluid. When simulating incompressible fluids by means of the SPH method, the bulk modulus is computed to guarantee a small Mach number [6], M, (typically $0.1 - 0.01$). The following expression

[5] The bulk modulus may be expressed by $B = -V\partial P/\partial V$.

[6] The Mach number is given by v/c, where v is the speed of an object moving through the fluid and c is the speed of sound in the fluid.

may be used [Monaghan, 1994]:

$$B_i = \left(\frac{\|\mathbf{v}\|_{max}}{M} \right)^2 \rho_i \,, \tag{2.52}$$

where $\|\mathbf{v}\|_{max}$ is the maximum velocity of the flow. For liquids, the speed of sound of a particle i, which represents the speed at which sound travels through the fluid element represented by the particle, is given by

$$c_i = \sqrt{\frac{B_i}{\rho_i}}. \tag{2.53}$$

In [Roy, 1995] the bulk modulus is computed using the practical equation:

$$B_i = \frac{200 \rho_i g H}{\gamma} \,, \tag{2.54}$$

where H is the maximum fluid depth and g is the gravitational constant. The sound speed is also adapted in [Roy, 1995]:

$$c_i = \sqrt{\frac{\gamma(P_i + B_i)}{\rho_i}} \quad . \tag{2.55}$$

One should observe that when Equation (2.51) is used, very large values of pressure are necessary to change density. This is the effect that endows the system with the desired behavior of incompressible fluids.

In a usual SPH simulation the differential equations presented before are integrated over time by means of finite difference methods [Roy, 1995]. In this work, we use the SPH technique to control swarms of robots. This is done by considering each robot as a particle and, in this case, the actual movement of the robots is responsible for the time integration. The details of this approach are given in Chapter 4. Details of the full derivation of the

equations presented in this section can be found in [Liu and Liu, 2003].

2.9 Voronoi Tessellation

An important geometric tool which will be used in our Locational Optimization based controllers (see Chapter 5) is the Voronoi Tessellation. Given the set of points $\mathcal{P} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$, often called *sites*, distributed over a bounded domain Ω, with boundary $\partial\Omega$, we define the Voronoi region, or Voronoi cell, V_i, associated to the point $\mathbf{q}_i$ according to a given distance function d as:

$$V_i = \{\mathbf{q} \in \Omega | d(\mathbf{q}, \mathbf{q}_i) \leq d(\mathbf{q}, \mathbf{q}_j), \forall j \neq i\}. \tag{2.56}$$

The definition in (5.2) is in fact a generalization of the ordinary definition of Voronoi regions based on the Euclidean distance [Okabe et al., 2000]. The generalized Voronoi tessellation of the set $\mathcal{P}$, $V(\mathcal{P})$, is the collection of such regions. The Voronoi boundary ∂V_i is defined as:

$$\partial V_i = \cup_{j=1}^{n} l_{ij} \cup \{\partial\Omega \cap V_i\}. \tag{2.57}$$

where l_{ij} is the bisector:

$$l_{ij} = \{\mathbf{q} \in \Omega | d(\mathbf{q}, \mathbf{q}_i) = d(\mathbf{q}, \mathbf{q}_j), j \neq i\}. \tag{2.58}$$

Assuming that Ω is a convex polytope and d is the Euclidean distance, the boundaries are hyperplanes and the Voronoi cells are convex. In this case, two neighbor sites i and j are associated with cells that share a hyperplane, and this hyperplane intersects the segment $\overline{\mathbf{q}_i\mathbf{q}_j}$ at its midpoint, and

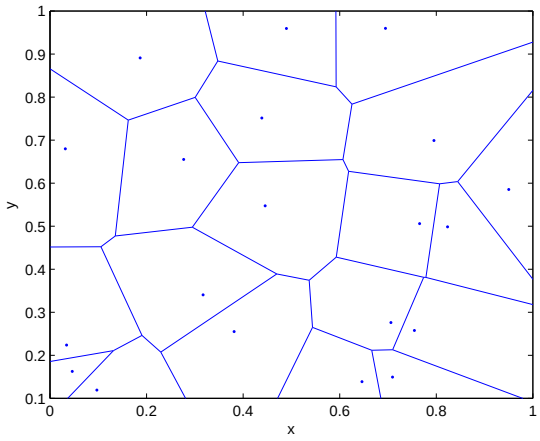


Figure 2.10: Example of a Voronoi tessellation in the case of Euclidean distance in a $2D$ domain.

perpendicular to the segment. The equation for this hyperplane bisector is given by:

$$(\mathbf{q}_i - \mathbf{q}_j)^T \mathbf{q} = \frac{1}{2}(\|\mathbf{q}_i\|^2 - \|\mathbf{q}_j\|^2). \tag{2.59}$$

An example of a Voronoi tessellation according to the Euclidean distance in a $2D$ domain is presented in Figure 2.10. For an extensive treatment of Voronoi tessellations we refer to [Okabe et al., 2000].

Chapter 3

Related Work

The term *Swarm Intelligence* was created in [Beni and Wang, 1989] to characterize a novel approach to control distributed cellular robotic systems. Later, in [Bonabeau et al., 1999], this expression was extended to describe a new computational paradigm for solving distributed problems based on the principles of intelligent collective behavior of natural systems, such as ant colonies and bird flocks. Basically, such systems rely on the property that complex group behaviors emerge from simple interactions among group agents. In the past, the use of Swarm Intelligence to control teams of robots was called *Swarm Robotics*. Swarm Robotics is now understood as a novel field of research whose main objective is to design and implement robotic systems composed by a large number of simple limited robots that cooperate to reach a pre-specified goal.

3.1 Biologically Inspired Approaches

Since the beginning, biological systems have been the inspiration of many works in Swarm Robotics. Inspired by the clustering behavior of ants, [Deneubourg et al., 1991] explored the idea of sorting objects in a closed arena with robots employing minimal rules. Also, based on hypotheses on the behavior of ants, many other works followed. Some ex-

amples are [Wilson et al., 2004], which explored the mechanisms that enable the sorting of any number of object types into an annular pattern, [Koenig and Liu, 2001], which addressed the problem of terrain coverage by allowing the robots to leave markings in the terrain just like ants do, and [Kumar and Sahin, 2003], which applied heuristics based on cognitive maps in ants for a mine detection application. The authors of [Berman et al., 2007] were also inspired by ants behavior. In their work, the problem of deploying robots to multiple sites in a specified ratio is addressed by using a model of ant house hunting. This model reflects the process through which an ant colony selects a new home from several sites and emigrates through quorum-dependent recruitment mechanisms. In the same spirit of ant based algorithms, [Krishnanand et al., 2006] proposed a glowworm metaphor based distributed algorithm to solve the problem of multiple source localization. The exploration strategy used by ladybugs when hunting for aphids is applied to control a network of robots to cover and explore an area in [Schwager et al., 2008a].

Other works have tried to mimic the evolution process. They have applied artificial evolution of individuals to synthesize effective collective behaviors. In [Reynolds, 1993] the control system of a group of virtual creatures, called boids, was evolved to avoid collisions with static obstacles and to escape from manually programmed predators. In [Ward et al., 2001], groups of artificial fish were evolved to display schooling behavior. [Spector et al., 2003] used genetic programming to evolve group behaviors for flying agents in a simulated environment. In [Gaudiano et al., 2005] and [Soto and Lin, 2005], genetic algorithms were applied to design controllers for swarms of Unmanned Air Vehicles (UAVs). [Dorigo et al., 2004] also used artificial evolution to derive effective controllers for both the aggregation and the coordinated motion of

a self-assembling and self-organizing artifact, called swam-bot (s-bot). The human immune system has also been a source of inspiration for many researchers as in [Jun et al., 1999] and [Sun et al., 2001]. The main drawback of those biologically-inspired approaches is the difficulty to establish mathematical guarantees such as convergence, stability, computational complexity, etc.

3.2 Physics Based Methods

Another class of works in robotics is the one with inspiration in physics phenomena. One of the earliest physics-based method is the *Potential Fields* approach. Most works using Artificial Potential Fields (APF) deal with just one robot that navigates through a field of obstacles to get to a goal position [Khatib, 1986], [Connolly et al., 1990], [Rimon and Koditschek, 1992], [Valavanis et al., 2000], and [Pimenta et al., 2006a]. In these cases, an APF is generated such that obstacles exert repulsive virtual forces while the goal exerts attractive ones.

Recently, APF methodology has been used to control teams of robots by considering inter-robots virtual forces. In [Song and Kumar, 2002], potential fields were used to control a group of robots for cooperative manipulation tasks. Actually, a composition of potential fields was used in that work. The authors defined three control modes for the derived decentralized controllers: approach, organization and transportation. Depending on the mode the controller was working, the potential field changed. In the same spirit, the works in [Pereira et al., 2003] and [Fink et al., 2008] also used compositions of vector fields for multi-robot manipulation.

[Chaimowicz et al., 2005] proposed a scalable potential field based solu-

tion for the problem of two-dimensional pattern generation by swarms of robots. The authors proposed to build functions by interpolating from points placed at the pattern in such a way that the pattern characterized a region of minimum of these functions. Therefore, by applying control laws based on the descent gradient of those functions, the patterns were formed by the robots. Artificial repulsive forces among the robots were also added to the control laws to avoid inter-robot collisions. Based on the results of [Chaimowicz et al., 2005], [Hsieh and Kumar, 2006] also applied artificial potential fields to address the synthesis of decentralized controllers that guarantee the convergence to a specified shape in two-dimensional spaces, and the stability of the resulting formation. Furthermore, [Hsieh and Kumar, 2006] considered the problem of staying "connected" during the pattern formation task. In [Hsieh et al., 2007] convergence and circulation of the static pattern in two-dimensional spaces was considered. A limitation of the above mentioned strategies is the fact that the environment where the robots navigate must be free of obstacles. In [Barnes et al., 2006], obstacle avoidance was also considered in the proposed potential field, but the resulting potential could contain local minima, which are points where the robots may get stuck.

In this work we propose a scalable physics-based solution which is able to handle obstacles. Similarly to [Shimizu et al., 2003a, Shimizu et al., 2003b, Kerr et al., 2004, Kerr and Spears, 2005, Pac et al., 2006], we propose to control swarms of robots by mimicking fluids behaviors. As we mentioned in Section 1.1, the main motivation comes from the fact that a great variety of characteristics desirable for a group of robots may be observed in fluids. [Shimizu et al., 2003a] and [Shimizu et al., 2003b] used *Stokesian Dynamics* to control a swarm of mobile robots. By using this technique, the robots had the behavior of particles suspended in a fluid and the swarm shape could be

controlled. [Kerr et al., 2004] and [Kerr and Spears, 2005] used the kinetic theory of gases to sweep a swarm of robots through a very long bounded region in a task of maximum coverage. [Pac et al., 2006] applied finite differences methods to solve fluid dynamics equations in a task of coverage by mobile sensor networks. In our approach, the Smoothed Particle Hydrodynamics (SPH) method is applied to model the "robotic fluid", more specifically, to model the interactions among the robots of the group. Besides, we use the Finite Element Method (FEM) to compute harmonic functions that determine external forces to the fluid. By means of a weak coupling between FEM and SPH, the derived controllers are decentralized in the sense that only local information is needed by the robots.

In parallel to our work, the authors of [Perkinson and Shafai, 2005] also proposed the use of SPH to control a swarm of robots in a task of maximum sensor coverage. They considered a two-dimensional environment with obstacles and objects of interest which needed additional sensor coverage. In [Perkinson and Shafai, 2005], the SPH equations for compressible fluids were used to mimick the behavior of air at 20^o C. More recently, another work [Pac et al., 2007] also proposed to use the SPH method to model a robotic swarm as a fluid and control its flow by tuning its flow parameters. In [Pac et al., 2007], simulations of coverage, dispatching through waypoints, and flocking are presented.

Differently from previous works, in the present work it is also considered the computation of harmonic functions that are used to generate external forces to the fluid. By using FEM in this computation, static obstacles of generic geometries can be modelled. Besides, we propose to address dynamic obstacles by considering them as sets of very dense SPH particles. We model the swarm as an incompressible fluid since we consider that it is interesting

to keep the group moving together if possible. We consider two tasks: (i) the pattern generation task, and (ii) the density constrained coverage task. Moreover, to the best of our knowledge this is the first time that real robot issues such as finite size and nonholonomic constraints are considered and stability and convergence proofs are devised for SPH based controllers. In the next chapter we detail the proposed approach.

3.3 Techniques from Mathematical Tools

Some researchers appealed to more formal approaches based on well established mathematical tools, such as graph theory, control theory, differential geometry, etc. A good example of such approaches is [Belta and Kumar, 2004] where the group of robots was treated as an abstraction in a space with smaller dimensions than the composite configuration space of all agents. Actually, the group of robots in two-dimensional environments was modelled as a deformable ellipse or a spanning rectangle. The problem of motion planning for the abstract group was then solved by devising decentralized controllers that allowed the control of the shape and the position of the ellipse or rectangle with guaranteed convergence. These results were extended in [Chaimowicz and Kumar, 2004] and [Michael et al., 2006]. In the former, a hierarchy of ground and air vehicles was built and the groups were allowed to split and merge. In the latter, groups of robots in three-dimensional environments were also considered. Extensions to consider finite size agents and kinematic constraints were devised in [Michael et al., 2007] and [Michael and Kumar, 2008].

A distributed and asynchronous approach for optimal coverage of a domain with identical mobile sensing agents is proposed in [Cortez et al., 2004]

based on a framework for optimized quantization derived in [Lloyd, 1982]. This approach is also in the class of mathematical tools based approaches since it is derived from the *Locational Optimization* theory [Okabe et al., 2000]. Locational optimization addresses how to place facilities or resources to minimize some cost [Weber, 1929, Drezner, 1995]. The canonical example is placing retail facilities to minimize the aggregate travel time of a population of customers.

In [Cortez et al., 2004], each agent (robot) follows a control law, which is a gradient descent algorithm that minimizes a functional encoding the quality of the sensing coverage. Further, this control law depends only on the information of position of the robot and of its immediate neighbors. Neighbors are defined to be those robots that are located in neighboring Voronoi cells. Besides, these control laws are computed without the requirement of global synchronization. The functional also uses a *distribution density function* which weights points or areas in the environment that are more important than others. Thus it is possible to specify areas where a higher density of agents is required. Furthermore, this technique is adaptive due to its ability to address changing environments, tasks, and network topology.

Different extensions of the framework devised in [Cortez et al., 2004] have been proposed in the literature. In [Cortez et al., 2005] the problem of limited-range interaction between agents was addressed. In [Salapaka et al., 2003], constraints were added to the minimization problem to deal with agents with heterogeneous resource capabilities and a modified deterministic annealing algorithm was used to overcome local minima issues in the context of *Unmanned Aerial Vehicles* (UAVs) mission planning. The problem of learning the distribution density function online while moving toward the optimal locations was addressed in [Schwager et al., 2006]

and [Schwager et al., 2008b]. In [Kwok and Martínez, 2008] the basic approach was extended to deal with agents with limited energy. In this case, generalized Voronoi diagrams such as *power diagrams* [Aurenhammer, 1987] are employed.

In the present work we propose three important extensions to the basic work in [Cortez et al., 2004]. First, we address the problem of incorporating heterogeneity in the robot team by allowing the robots to have different types of sensors. This first extension is actually a minor contribution since we use power diagrams, similarly to [Kwok and Martínez, 2008], with a different motivation. Second, we overcome the practical limitations of the point robot assumption in the original algorithm. Finally, we generalize the basic method to nonconvex environments. To the best of our knowledge, the last two extensions are not similar to any other extension found in the literature. The proposed extensions are presented in Chapter 5.

Chapter 4

A Fluid Based Approach for Swarm Control

The focus of this chapter is to use analogies with fluid dynamics models to control swarms of robots. As we mentioned in Chapter 1, the main motivation stems from the fact that a great variety of characteristics desirable for a group of robots can be observed in fluids. Some examples of such characteristics are: (i) fluids are easily deformed, (ii) fluids can easily contour objects, and (iii) the flow field variables and also the fluid phase can be easily manipulated in order to design desired behaviors. In this chapter we apply an incompressible fluid model to solve two different robotic tasks: (i) the *pattern generation task* and (ii) the *density constrained coverage task*. In the first four sections we address the pattern generation task. Possible applications of an efficient solution to this task are surveillance and cordoning off hazardous areas. The problem statement is presented in Section 4.1. The proposed solution is presented in Section 4.2. Simulations and experimental results are presented in Sections 4.3 and 4.4, respectively. In the last section of the chapter we show how the same solution proposed for pattern generation may be adapted to address multi-robot coverage.

4.1 The Pattern Generation Problem

The so-called *pattern generation problem* may be stated as follows:

Problem 4.1 *Given N robots and any initial spatial distribution, the geometry of the environment with static obstacles defining a compact domain $\Omega \subset \mathbb{R}^2$, and a curve $\Gamma : I \to \Omega$, where $I \subset \mathbb{R}$, find a controller which enables the robots, without colliding with static obstacles and each other, to form the pattern described by Γ.*

To simplify the problem we make the following assumptions:

Assumption 4.1 The number of robots is sufficiently large to form the desired pattern, and, at the same time, it is sufficiently small to guarantee enough space along the curve for all the robots.

Assumption 4.2 The robots are able to estimate positions and velocities of themselves and of other robots of the group located within a distance D.

Assumption 4.1 is evident and if it is not verified the problem becomes unfeasible. Since most actual robots have sensors and communication equipments, the estimation of positions and velocities in Assumption 4.2 is feasible in a large variety of scenarios.

A hard constraint of the problem is the fact that a map of the environment must be provided. However, this is a constraint of most motion planning approaches in robotics and its solution is out of the scope of this work. Next, we discuss the proposed methodology to solve the formulated problem.

4.2 Pattern Generation Task Solution

We assume that each robot of the team is a SPH particle subjected to an external force and, since we use kernels with compact support, it is possible to derive decentralized control laws based on the SPH equations. The resulting controllers are decentralized in the sense that only local information is necessary: the gradient of a potential function at the location of the robot i and position and velocity of the robot i itself and of the robots in the neighborhood of i. For a robot i with configuration $\mathbf{q}_i = [x_i, y_i]^T$ we define $\mathcal{N}_i$ as the set of robots in the neighborhood of robot i:

$$\mathcal{N}_i = \{j \neq i | \|\mathbf{q}_j - \mathbf{q}_i\| < D\}\,, \tag{4.1}$$

where the distance D is determined by the kernel support size.

Our approach is composed by two steps. In the first step, we compute a global potential function. This potential function is responsible to drive the robots to the desired pattern. The second step consists of controlling the robots by using control laws based on the SPH equations. These equations provide interaction forces among the agents of the group. Subsections 4.2.1 and 4.2.2 describe the first and the second steps, respectively.

4.2.1 Global Potential Functions

Our approach relies on the computation of a global potential function. In this subsection we present two examples of such functions: *harmonic functions* [Pimenta et al., 2007a] and *shape functions* [Hsieh and Kumar, 2006]. Harmonic functions can be efficiently computed in obstacle-filled environments. To numerically compute these functions, we use the Finite Element Method (FEM) as proposed in [Pimenta et al., 2005a]

and [Pimenta et al., 2007a]. The efficiency of such a method is due to its ability to work properly with unstructured meshes which are used to exactly decompose the solution domain. In the absence of obstacles, shape functions can be defined and they are used in Section 4.2.4, where we prove stability and convergence.

Harmonic Functions

If a safety factor, ϵ, is defined such that the desired pattern is represented by a region between two curves Γ_1 and Γ_2, we can define a harmonic function which drives the robots toward the goal region and, at the same time, drives the robots away from the obstacles. If the desired pattern, Γ, is parameterized by a function $s(x, y) = 0$, then Γ_1 is such that $s(x, y) = \epsilon$ and Γ_2 is such that $s(x, y) = -\epsilon$. Figure 4.1 presents an example of a domain with an obstacle and a circular pattern with a safety factor added. In fact, the presence of this safety factor is interesting in practical applications due to the presence of noise in the information provided by real localization systems.

As previously defined in Subsection 2.6, harmonic functions are solutions to the Laplace's equation. In order to guarantee uniqueness in the solution, we must define boundary conditions. We use constant Dirichlet boundary conditions such that a maximum value is obtained at the boundaries of the configuration space and a minimum value is obtained at the desired pattern. This boundary value problem (BVP) is given by:

$$\begin{cases} \nabla^2\phi = 0\,, \\ \phi(\Gamma_1) = \phi(\Gamma_2) = 0\,, \\ \phi(\partial\Omega_1) = \phi(\partial\Omega_2) = \phi(P) = V_c\,, \end{cases} \tag{4.2}$$

where ϕ is the harmonic function, V_c is a positive constant, and P is a

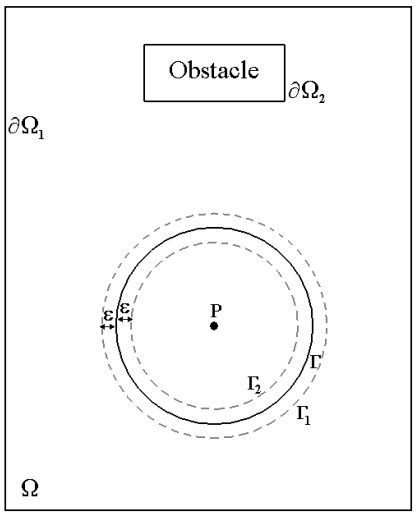


Figure 4.1: Domain example.

point defined inside the pattern in the case of closed curves to guarantee convergence from the interior of the pattern (see Fig. 4.1).

By using the boundary conditions presented in (4.2) we guarantee convergence of the integral curves of $-\nabla\phi$ to the desired pattern. Therefore, a robot that follows the field lines defined by $-\nabla\phi$ reaches the target (the region delimited by Γ_1 and Γ_2) in a finite time without any collision with obstacles.

Remark 4.1 The computation in (4.2) can be seen as the solution of an analogous electrostatic problem [Pimenta et al., 2006a]. Such an electrostatic analogous problem considers a homogeneous isotropic medium in the absence of density of charge. In this case, ϕ corresponds to the scalar electric potential and $-\nabla\phi$ corresponds to the electric field.

Remark 4.2 Since the harmonic functions we compute here are designed to satisfy the properties of navigation functions (see Section 2.5), this first step of our approach could be replaced by any other navigation function

computation method.

As in [Pimenta et al., 2006a], we propose to compute harmonic functions numerically by means of FEM (see Section 2.7). This is useful when the obstacles have generic geometries.

Shape Functions

In obstacle-free environments with desired smooth star shapes, we can use shape functions. As defined in [Rimon and Koditschek, 1992], star-shaped sets are those characterized by the possession of a distinguished "center point" $\hat{\mathbf{q}}$ from which all rays cross their boundary once and only once. Star-shaped sets are topologically equivalent to discs.

According to [Hsieh, 2007], given a desired pattern Γ, a shape function, ϕ, is a positive semi-definite function with a minimum value equal to zero at the boundary Γ. For a desired curve parameterized by a function $s(x, y) = 0$ we have

$$\phi = s(x, y)^2 \tag{4.3}$$

as a candidate shape function. As proposed in [Hsieh, 2007], for star shapes, it is interesting to have $\phi = s^2$ such that:

1. $s(x, y)$ is at least twice differentiable; and

2. $s(x, y)$ has a unique minimum at $\hat{\mathbf{q}}$.

These properties will be important in the stability and convergence proofs presented in Section 4.2.4.

4.2.2 Controllers based on a Fluid Model

The second stage of our methodology consists of applying decentralized controllers that drive the robots to the region where the pattern is located and distribute them inside it. These controllers are derived by considering each robot as a SPH particle subjected to an external force. First, we will present a controller under the assumption of fully-actuated, holonomic, point robots. Second, we will show how our control law can be adapted to accommodate practical robots issues.

Holonomic Point Robot Abstraction

Our controller is derived by considering each robot as a SPH particle at $\mathbf{q}_i = [x_i, y_i]^T$ subjected to an external force provided by the descent gradient of a global potential function. In this subsection we consider vehicles with second order dynamics. Under the assumption of fully-actuated, holonomic, point robots, each robot's acceleration is given by

$$\ddot{\mathbf{q}}_i = \mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}, t)\,, \tag{4.4}$$

where $\mathbf{q} = [\mathbf{q}_1^T, \ldots, \mathbf{q}_N^T]^T$ is the configuration of the group. In principle, N is the total number of robots, but as we will show later the control law for agent i will depend only on the agents in the neighborhood $\mathcal{N}_i$.

The control law for each robot is given by:

$$\mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{b}_i - \zeta \mathbf{v}_i + k\mathbf{f}_i\,, \tag{4.5}$$

where

$$\mathbf{b}_i = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij}\,, \tag{4.6}$$

k and ζ are positive tuning constants and $\mathbf{f}_i$ is given by a vector given by $-\nabla\phi$. In fact, we use vector fields of the form:

$$\mathbf{f}_i = \begin{cases} -\frac{\nabla\phi(\mathbf{q}_i)}{\|\nabla\phi(\mathbf{q}_i)\|^\beta} & \text{if } \nabla\phi(\mathbf{q}_i) \neq 0 \\ 0 & \text{if } \nabla\phi(\mathbf{q}_i) = 0 \end{cases}, \tag{4.7}$$

where β is a non-negative integer number. In (4.6) the SPH conservation of momentum equation (see (2.46), page 38) is used. In this work we use the density, ρ_i, defined in ((2.45), page 38), the cubic spline kernel, W, defined in ((2.36), page 35), the artificial viscosity, Π_{ij}, defined in ((2.48), page 38), the sound speed defined in ((2.55), page 40), and the equation of state that determines the pressure, P_i, for incompressible fluids defined in ((2.51), page 39) with B_i given by ((2.54, page 40)). In (4.5) we include a dissipative term proportional to the robot velocity $\mathbf{v}_i$, which represents a damping to stabilize the system.

It is important to mention that $\mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}})$ in (4.5) can be computed by taking into account only robots in the neighborhood $\mathcal{N}_i$ defined in (4.1). This is because of the compact support of the kernel, W, that guarantees that robots outside the given neighborhood do not contribute to the sum in (4.6).

Remark 4.3 If we use a harmonic function as the global potential function, the proposed solution is analogous to the solution of a problem where a charged fluid is confined in a region where an electrostatic field is applied. Moreover, if the Finite Element Method is used to compute the harmonic function, this solution establishes a weak coupling [1] between FEM and SPH.

[1] This coupling is said to be weak because the Finite Element Method is executed only once and does not take into account the current distribution of the particles of the SPH.

Finite-size, Nonholonomic Robots

Now, we will describe how our approach may be adapted to take into account practical robot issues. The first issue we address is the finite size of actual robots. The static obstacles are directly taken into account since we plan our potential functions in the robots configuration space. We also assume that our robots are circular in shape with radius R. Given two robots, we guarantee that the robots do not collide with each other if $\|\mathbf{q}_{ij}\| \geq 2R + \varepsilon$, where ε is a safety factor. The collision avoidance of our approach is performed by the artificial viscosity term in (2.48), page 38, with

$$\mu_{ij} = \frac{h\mathbf{v}_{ij} \cdot \mathbf{q}_{ij}}{(\|\mathbf{q}_{ij}\| - (2R + \varepsilon))^2}. \tag{4.8}$$

This adaptation guarantees a repulsive term in (4.6) between robots which are moving toward each other. This term is repulsive since $\Pi_{ij} \geq 0$ and $\nabla_i W_{ij}$ points in the direction of $-\mathbf{q}_{ij}$. Note that $\Pi_{ij} \to \infty$ when $\|\mathbf{q}_{ij}\| \to (2R + \varepsilon)$, *i.e.*, when the robots are about to collide.

Besides the robot size, motion constraints are also an important consideration. Most actual mobile robots have their movements subjected to nonholonomic constraints. As mentioned in Chapter 2, several comercial platforms can be controlled by specifying their linear and angular velocities, v and ω respectively, and are subjected to the no-slip constraint $\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0$, where θ is the robot orientation. Therefore, the following model may be used:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{4.9}$$

We are interested in controlling the robot position $[x, y]^T$, but it is not

possible to use the model in (4.9) since the angular velocity does not appear explicitly in the equations of $\dot{x}$ and $\dot{y}$. To circumvent this problem we redefine the system output as $[x_d, y_d]^T = [x+d\cos(\theta), y+d\sin(\theta)]^T$, which corresponds to the position of the point $[d, 0]^T$ in the robot frame. Therefore,

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -d\sin(\theta) \\ \sin(\theta) & d\cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{4.10}$$

The robot may then be controlled by using feedback linearization [Murray et al., 1994]:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\frac{\sin(\theta)}{d} & \frac{\cos(\theta)}{d} \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_d^{ref} \\ \dot{y}_d^{ref} \end{bmatrix}. \tag{4.11}$$

Therefore, by applying (4.11) in (4.10) we obtain:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = \begin{bmatrix} \dot{x}_d^{ref} \\ \dot{y}_d^{ref} \end{bmatrix}.$$

Note that the evolution of the robot orientation, $\theta(t)$, is not controlled.

Since our controllers were devised for robots fully actuated in their acceleration, we control differential drive, kinematically controlled robots by integrating the acceleration inputs in (4.5):

$$\begin{bmatrix} \dot{x}_{d_i}^{ref} \\ \dot{y}_{d_i}^{ref} \end{bmatrix} = \int \mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}})dt. \tag{4.12}$$

Now, each robot is represented in its configuration space by the feedback linearization point $[x_{d_i}, y_{d_i}]^T$ such that the physical extent of the robot lies within the circle of center $[x_{d_i}, y_{d_i}]^T$ and radius $R' = R + d$ (see Figure 4.2).

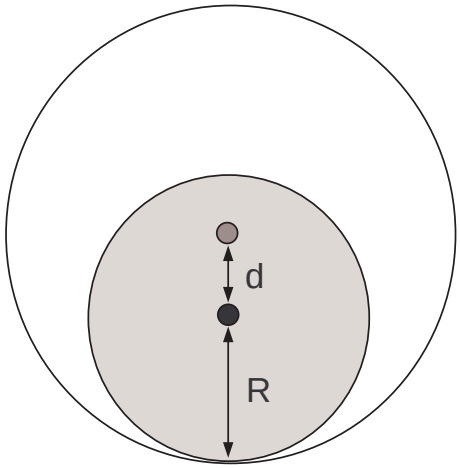
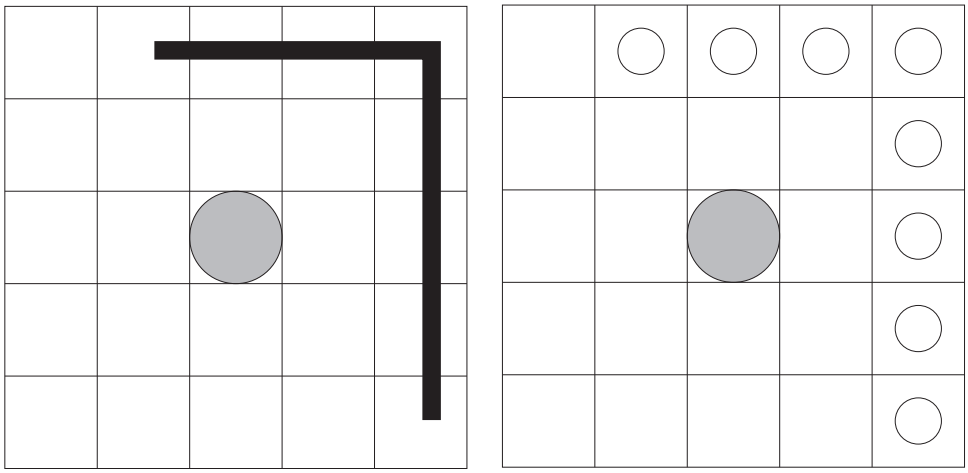


Figure 4.2: Feedback linearization point. The light grey circle represents a circular robot with radius R. Assuming a feedback linearization point at a distance d from the center, the white circle with radius $R' = R + d$ is used to guarantee collision avoidance.

Our approach is adapted such that the SPH particles are placed at the points $[x_{d_i}, y_{d_i}]^T$. Moreover, we replace the robot radius R in (4.8) by $R' = R + d$.

4.2.3 Virtual Particles

Besides driving the robots toward the goal, the external force in (4.5), $\mathbf{f}_i$, aims to avoid collisions between robots and static obstacles. When controlling multiple robots, due to the presence of inter-particle forces, $\mathbf{b}_i$, the external force, $\mathbf{f}_i$, may be not enough to avoid collisions. We add temporary virtual particles right at the boundaries of the configuration space such that we can guarantee collision avoidance. One option is to take advantage of the collision avoidance property provided by the artificial viscosity. There are several ways of implementing this virtual particle idea. A first idea is to create a temporary virtual particle at the closest boundary point, $\mathbf{p}$. Then



(a) (b)

Figure 4.3: Multiple virtual particles in an occupancy grid. (a) Robot and obstacles. (b) Virtual particles from occupied cells.

we adapt the term $\mathbf{b}_i$ in (4.6) such that:

$$\mathbf{b'}_i = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij}(h) - \lambda \Pi_{ip} \nabla_i W_{ip}(h') \,, \qquad (4.13)$$

where λ is a positive constant, j iterates only through the N particles that represent real robots, and p refers to the virtual particle. Due to the fact that the size of the robot is already taken into account in the configuration space, we use $R = 0$ in (4.8). Notice that, in this case, the virtual particle does not change the density ρ_i and also does not have its own density. The other terms necessary to compute Π_{ip} are $\overline{\rho}_{ip} = \rho_i$ and $\overline{c}_{ip} = c_i$.

Instead of using a single virtual particle, another option is to assign virtual particles to each cell with obstacle in a local occupancy grid. This option was found to be the most robust during experiments (see Figure 4.3).

Another possible implementation of virtual particles is to consider very dense virtual particles. These particles may be considered just like the other particles in the conservation of momentum equation (4.6). The key idea is

that particles which are denser than the others are repulsive particles. To understand this fact one should agree that particles which are denser than the reference density, ρ_0, produce positive values of pressure according to the equation of state (2.51):

$$P_i = B_i \left[\left(\frac{\rho_i}{\rho_0} \right)^{\gamma} - 1 \right] . \tag{4.14}$$

Since the gradient, $\nabla_i W_{ij}$, at particle i points toward particle j, it is clear that terms with positive values of pressure in (4.6) repel particle i.

To create very dense particles we propose to assign a high value of mass, m_d, to these particles. As we presented in Chapter 2, the density of a particle, i, is computed according to the conservation of mass:

$$\rho_i = \sum_j m_j W(\mathbf{x}_i - \mathbf{x}_j, h) , \tag{4.15}$$

where

$$W(\mathbf{r}, h) = \frac{10}{7\pi h^2} \begin{cases} 1 - \frac{3}{2}\kappa^2 + \frac{3}{4}\kappa^3 & \text{if} \quad 0 \leq \kappa \leq 1 , \\ \frac{1}{4}(2-\kappa)^3 & \text{if} \quad 1 \leq \kappa \leq 2 , \\ 0 & \text{otherwise} , \end{cases} \tag{4.16}$$

where $\kappa = \|\mathbf{r}\|/h$.

It is necessary that the density of the virtual particle $\rho_d \gg \rho_0$. This can be accomplished by doing $\rho_d = \sigma\rho_0$, where $\sigma \gg 1$. Therefore, we compute m_d using:

$$m_d = \left(\frac{7\pi h^2}{10} \right) \sigma\rho_0 . \tag{4.17}$$

The density of a particle, j, with mass given by (4.17) is guaranteed to be larger than or equal to $\sigma\rho_0$, since $m_j W_{ij}(0) = \sigma\rho_0$. Thus, the term

proportional to the pressure P_j in (4.6) will be positive, and consequently it will be a repulsive term.

Dynamic obstacles are those that are unknown a priori and/or are able to move. We can also use virtual particles to address collision avoidance in the presence of dynamic obstacles. It should be clear that addressing dynamic obstacles is not the main topic of this work and the proposed solution has limitations. The main issue is the fact that the robots may be trapped by the dynamic obstacles. Since the repulsive terms provide forces which are parallel to the segment which links the robot to the obstacle particle, it is possible to find local minima. One may propose new strategies based on forces that act perpendicularly to the referred segment to solve this issue. This is out of the scope of this work and may be a possible future direction of research.

4.2.4 Analysis

Our stability and convergence analysis is built upon the results in [Hsieh and Kumar, 2006] and [Hsieh, 2007] and follows a similar methodology. Similarly, we assume obstacle-free environments and $\mathbf{f}_i$ in (4.5) is given by $-\nabla\phi$, where ϕ is the shape function decribed in 4.2.1 for the case of smooth star-shaped patterns. We also assume that the robots are represented by identical SPH particles with mass m.

As in [Hsieh and Kumar, 2006], we define the function $\phi_S(\mathbf{q})$ as a measure of performance:

$$\phi_S(\mathbf{q}) = k \sum_i \phi(\mathbf{q}_i)\,, \tag{4.18}$$

with $k > 0$. The function ϕ_S provides a measure of how close the team is to Γ. The greater the value of this function, the greater the distance of the

team to the pattern.

The following Lemma concerning the Hessian of ϕ_S, $\mathbf{H}_{\phi_S}$, will be useful in this section.

Lemma 4.1 *Given a star-shaped boundary* Γ *as described in Subsection 4.2.1, the Hessian of the function* ϕ_S, $\mathbf{H}_{\phi_S}$, *is positive semi-definite on* Γ.

Proof:[Hsieh, 2007] According to (4.18) and (4.3), we have

$$\phi_S = k \sum_i^N s(\mathbf{q}_i)^2 = k \sum_i^N s_i^2 \,, \tag{4.19}$$

thus the Hessian is given by

$$\mathbf{H}_{\phi_S} = 2k \left(\mathbf{H}^I_{\phi_S} + \mathbf{H}^{II}_{\phi_S} \right) \,, \tag{4.20}$$

where

$$\begin{aligned} \mathbf{H}^I_{\phi_S} &= diag(\mathbf{H}^I_{\phi_1}, \ldots, \mathbf{H}^I_{\phi_i}, \ldots, \mathbf{H}^I_{\phi_N}) \,, \\ \mathbf{H}^{II}_{\phi_S} &= diag(\mathbf{H}^{II}_{\phi_1}, \ldots, \mathbf{H}^{II}_{\phi_i}, \ldots, \mathbf{H}^{II}_{\phi_N}) \end{aligned}$$

are $2N \times 2N$ block diagonal matrices. Each block is given by

$$\begin{aligned} \mathbf{H}^I_{\phi_S} &= (\nabla_i s_i)(\nabla_i s_i^T) \,, \\ \mathbf{H}^{II}_{\phi_S} &= s_i \mathbf{H}_{s_i} \,, \end{aligned}$$

where $\mathbf{H}_{s_i}$ is the Hessian of s_i. Since $s_i = 0$ at Γ, we conclude that $\mathbf{H}_{\phi_S}$ is positive semi-definite at Γ. ∎

The minimum value $\phi_S = 0$ is obtained when all the robots reach the desired boundary. Therefore, the primary objective of our controller should

be to minimize ϕ_S. The next Proposition assures that our system equilibrium points are at an extremum of ϕ_S.

Proposition 4.1 *Given a system of N point robots with dynamics $\ddot{\mathbf{q}}_i = \mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}, t)$ and a control law determined by (4.5), where $\mathbf{f}_i = -\nabla\phi$ and ϕ is a shape function, then the system equilibrium points are at an extremum of ϕ_S.*

Proof: Since the system is in equilibrium we have $\ddot{\mathbf{q}}_i = 0$ and $\dot{\mathbf{q}}_i = 0$, $i = 1, \ldots, N$. Consequently, for every robot $\Pi_{ij} = 0$ and $\mathbf{u}_i = 0$. Therefore,

$$\sum_i \mathbf{u}_i = \sum_i \left[-\sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} - k\nabla\phi_i \right] = 0.$$

Since $\nabla_i W_{ij} = -\nabla_j W_{ij}$,

$$\sum_i \mathbf{u}_i = k \sum_i \nabla\phi_i = 0. \tag{4.21}$$

However,

$$\nabla\phi_S = k \sum_i \nabla\phi_i = 0\,, \tag{4.22}$$

which is the necessary condition for ϕ_S to be at an extremum. ∎

Proposition 4.2 *Consider the positive semi-definite function:*

$$V = \phi_S + \sum_i e_i' + \frac{1}{2}\mathbf{v}^\mathbf{T}\mathbf{v}\,, \tag{4.23}$$

where $\mathbf{v} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_N^T]^T$ and e_i' is the part of the internal energy related to conservative forces such that (see eq. (2.47), page 38):

$$\frac{de_i'}{dt} = \frac{1}{2} \sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \mathbf{v}_{ij} \cdot \nabla_i W_{ij}. \tag{4.24}$$

Consider also the set $\Omega_c = \{\mathbf{x} \in X | V(\mathbf{q}, \mathbf{v}) \leq c\}$*, where* X *is the state space defined by* $\mathbf{x} = [\mathbf{q}_1^T, \mathbf{v}_1^T, \ldots, \mathbf{q}_N^T, \mathbf{v}_N^T]^T$*, and* $c \in \mathbb{R}^+$*. Given the system of robots defined in Proposition 4.1 with any initial condition* $\mathbf{x}_0 \in \Omega_c$*, the system converges to an invariant set,* $\Omega_I \subset \Omega_c$*, such that the points in* Ω_I *minimize the measure function* ϕ_S*.*

Proof: Since V is continuous, we conclude that Ω_c is closed for some $c > 0$. Also, due to the fact that $\phi_S + \sum_i e_i' \leq c$ and $\mathbf{v}^T\mathbf{v} \leq c$ we conclude that Ω_c is compact.

We have that

$$\dot{V} = \sum_i (k\nabla\phi_i^T \dot{\mathbf{q}}_i + \mathbf{v}_i^T \dot{\mathbf{v}}_i) + \sum_i \frac{de_i'}{dt}. \tag{4.25}$$

By using (4.5) and (4.24), and the fact that $\nabla_i W_{ij} = -\nabla_j W_{ji}$, $\Pi_{ij} = \Pi_{ji}$, $\dot{\mathbf{q}}_i = \mathbf{v}_i$, and $\dot{\mathbf{v}}_i = \ddot{\mathbf{q}}_i$, one has that

$$\begin{aligned}
\dot{V} =& \sum_i k\nabla\phi_i^T \mathbf{v}_i + \sum_i \mathbf{v}_i^T \left[-\sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} - \zeta\mathbf{v}_i - k\nabla\phi_i \right] \\
&+ \sum_i \frac{1}{2} \sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \mathbf{v}_{ij}^T \nabla_i W_{ij} \\
=& \sum_i \mathbf{v}_i^T \left[-\sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} - \zeta\mathbf{v}_i \right] + \\
&\sum_i \frac{1}{2} \mathbf{v}_i^T \sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} + \\
&\sum_j \frac{1}{2} \mathbf{v}_j^T \sum_i m \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \nabla_j W_{ji} \\
=& -\sum_i \zeta\mathbf{v}_i^T \mathbf{v}_i - \sum_i \frac{1}{2} \sum_j m\Pi_{ij} \mathbf{v}_i^T \nabla_i W_{ij} - \sum_j \frac{1}{2} \sum_i m\Pi_{ji} \mathbf{v}_j^T \nabla_j W_{ji} \\
=& -\sum_i \zeta\mathbf{v}_i^T \mathbf{v}_i - \sum_i \frac{1}{2} \sum_j m\Pi_{ij} \mathbf{v}_{ij}^T \nabla_i W_{ij}.
\end{aligned}$$

Our kernel is such that

$$\nabla_i W_{ij} = -\|\nabla_i W_{ij}\| \frac{\mathbf{q}_{ij}}{\|\mathbf{q}_{ij}\|} \tag{4.26}$$

By using the fact that $\Pi_{ij} > 0$ when $\mathbf{v}_{ij}^T \mathbf{q}_{ij} < 0$ and $\Pi_{ij} = 0$ otherwise, we conclude that

$$\dot{V} = -\sum_i \zeta \mathbf{v}_i^T \mathbf{v}_i - \sum_i \frac{1}{2} \sum_j m \Pi_{ij} \mathbf{v}_{ij}^T \nabla_i W_{ij} \leq 0. \tag{4.27}$$

By using the LaSalle's Invariance Principle (see Section 2.4), we conclude that for any $\mathbf{x}_0 \in \Omega_c$ the system converges asymptotically to the largest invariant set Ω_I contained in $\Omega_o = \{\mathbf{x} \in X | \dot{V} = 0\}$, which corresponds to $\mathbf{v}_i = 0$, $\forall i$, and $\Omega_o \subset \Omega_c$. Since Ω_I contains all equilibrium points in Ω_c and based on the Proposition 4.1 we conclude that all points in Ω_I satisfy the necessary condition for ϕ_S to be at an extremum. ∎

Proposition 4.3 *Consider the set Ω_S defined by*

$$\Omega_S = \{\mathbf{x} \in X \,|\phi(\mathbf{q}_i) = 0,\ \mathbf{v}_i = 0,\ \rho_i = \rho_0,\ i = 1, \ldots, N\}, \tag{4.28}$$

where ϕ is a shape function. Given the system of N robots defined in Proposition 4.1, the set Ω_S is a stable submanifold and $\Omega_S \subset \Omega_I$.

Proof: The potential energy of the system is given by $U = \phi_S + \sum_n e'_n$. Firstly, we show that the gradient of U is equal to zero at Ω_S. Let

$$\frac{de'_i}{dt} = \frac{\partial e'_i}{\partial t} + \frac{\partial e'_i}{\partial \mathbf{q}_i}^T \mathbf{v}_i + \sum_{j \neq i} \frac{\partial e'_i}{\partial \mathbf{q}_j}^T \mathbf{v}_j. \tag{4.29}$$

But the temporal partial derivative is zero and de'_i/dt is given by (4.24).

By using $\nabla_i W_{ij} = -\nabla_j W_{ji}$ in (4.24), we can write after some algebra

$$\frac{\partial e'_i}{\partial \mathbf{q}_i} = \frac{1}{2}\sum_j m\left(\frac{P_i}{\rho_i^2}+\frac{P_j}{\rho_j^2}\right)\nabla_i W_{ij}\,, \tag{4.30}$$

$$\frac{\partial e'_i}{\partial \mathbf{q}_j} = \frac{1}{2} m\left(\frac{P_i}{\rho_i^2}+\frac{P_j}{\rho_j^2}\right)\nabla_j W_{ji}. \tag{4.31}$$

Therefore

$$\frac{\partial \sum_n e'_n}{\partial \mathbf{q}_i} = \sum_j m\left(\frac{P_i}{\rho_i^2}+\frac{P_j}{\rho_j^2}\right)\nabla_i W_{ij}. \tag{4.32}$$

After using the state equation (2.51) and (2.54)

$$\frac{\partial \sum_n e'_n}{\partial \mathbf{q}_i} = \sum_j m\left[\sigma\frac{\rho_i^{\gamma-1}}{\rho_0^\gamma}-\frac{\sigma}{\rho_i}+\sigma\frac{\rho_j^{\gamma-1}}{\rho_0^\gamma}-\frac{\sigma}{\rho_j}\right]\nabla_i W_{ij}\,,$$

If $\rho_i = \rho_j = \rho_0$, which is a necessary condition for a point in Ω_S, then this gradient is null. For a shape function, it is shown in [Hsieh and Kumar, 2006] that $\nabla_i \phi_i = 0$ at Γ ($\phi(\mathbf{q}_i) = 0$). Therefore, $\nabla U = 0$ at Ω_S.

Now, we need to show that the Hessian of U, $\mathbf{H}_U = \mathbf{H}_{\phi_S} + \mathbf{H}_{\sum_i e'_i}$, is positive semi-definite when $\mathbf{q}_i \in \Gamma$ and $\rho_i = \rho_0\ \forall i$. It is proved in Lemma 4.1 that the $2N \times 2N$ matrix $\mathbf{H}_{\phi_S}$ is positive semi-definite when $\phi(\mathbf{q}_i) = 0$. Therefore, we need to prove that $\mathbf{H}_{\sum_i e'_i} \geq 0$ when $\rho_i = \rho_0$.

The second derivatives are given by:

$$\begin{aligned}\frac{\partial^2 \sum_n e'_n}{\partial \mathbf{q}_i \partial \mathbf{q}_i} =& \sum_j \frac{\partial^2 \sum_n e'_n}{\partial \rho_j \partial \mathbf{q}_i}\frac{\partial \rho_j}{\partial \mathbf{q}_i} + \frac{\partial^2 \sum_n e'_n}{\partial \rho_i \partial \mathbf{q}_i}\frac{\partial \rho_i}{\partial \mathbf{q}_i} + \\ & \sum_j m\left[\sigma\frac{\rho_i^{\gamma-1}}{\rho_0^\gamma}-\frac{\sigma}{\rho_i}+\sigma\frac{\rho_j^{\gamma-1}}{\rho_0^\gamma}-\frac{\sigma}{\rho_j}\right]\frac{\partial \nabla_i W_{ij}}{\partial \mathbf{q}_i}\,,\end{aligned}$$

$$\frac{\partial^2 \sum_n e'_n}{\partial \mathbf{q}_k \partial \mathbf{q}_i} = \sum_j \frac{\partial^2 \sum_n e'_n}{\partial \rho_j \partial \mathbf{q}_i} \frac{\partial \rho_j}{\partial \mathbf{q}_k} + \frac{\partial^2 \sum_n e'_n}{\partial \rho_i \partial \mathbf{q}_i} \frac{\partial \rho_i}{\partial \mathbf{q}_k} + \sum_j m \left[\sigma \frac{\rho_i^{\gamma-1}}{\rho_0^{\gamma}} - \frac{\sigma}{\rho_i} + \sigma \frac{\rho_j^{\gamma-1}}{\rho_0^{\gamma}} - \frac{\sigma}{\rho_j} \right] \frac{\partial \nabla_i W_{ij}}{\partial \mathbf{q}_k} ,$$

where σ is a positive constant.

After manipulations and using the fact that $\rho_i = \rho_0$

$$\mathbf{H}_{\sum_i e'_i} = \frac{m^2 \gamma \sigma}{\rho_0^2} \mathbf{A}\mathbf{A}^T \geq 0 , \tag{4.33}$$

where

$$\mathbf{A} = \begin{pmatrix} \sum_k \partial \frac{W_{1k}}{\partial x_1} & \partial \frac{W_{12}}{\partial x_1} & \dots & \partial \frac{W_{1N}}{\partial x_1} \\ \sum_k \partial \frac{W_{1k}}{\partial y_1} & \partial \frac{W_{12}}{\partial y_1} & \dots & \partial \frac{W_{1N}}{\partial y_1} \\ \vdots & \vdots & \vdots & \vdots \\ \partial \frac{W_{N1}}{\partial x_N} & \partial \frac{W_{N2}}{\partial x_N} & \dots & \sum_k \partial \frac{W_{Nk}}{\partial x_N} \\ \partial \frac{W_{N1}}{\partial y_N} & \partial \frac{W_{N2}}{\partial y_N} & \dots & \sum_k \partial \frac{W_{Nk}}{\partial y_N} \end{pmatrix} .$$

Therefore, Ω_S is a stable submanifold and since $\mathbf{v}_i = 0$ for all i, $\Omega_S \subset \Omega_I$.

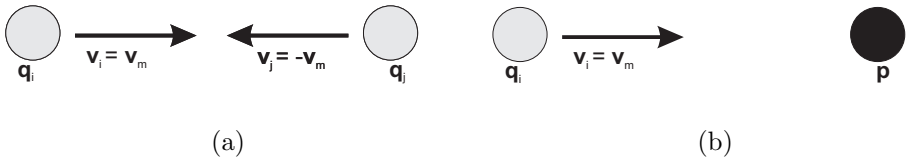
■

Proposition 4.4 *For any smooth star shape, Γ, the system of N robots defined in Proposition 4.1 with fixed ρ_i such that $\rho_i = \rho_0$, $i = 1, \dots, N$, converges to the desired boundary for any $\mathbf{x}_0 \in \Omega_c$.*

Proof: If $\rho_i = \rho_0$, $\forall i$, for all time t, then the equilibrium of the system is given by $\nabla_i \phi(\mathbf{q}_i) = 0$, $\forall i$. Proposition 4.2 guarantees the system converges to Ω_I. The function ϕ is a shape function and according to [Hsieh and Kumar, 2006] $\nabla_i \phi(\mathbf{q}_i) = 0$ if and only if $\mathbf{q}_i \in \Gamma$. Therefore $\Omega_I \equiv \Omega_S \equiv \Gamma$. ■

Next, we present results concerning collision avoidance.

Proposition 4.5 *Given a pair of robots, i and j, with circular shape defined by the radius R, dynamics $\ddot{\mathbf{q}}_i = \mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}, t)$, and the control law determined*



(a) (b)

Figure 4.4: Worst scenarios for collision avoidance. Fig. 4.4(a) worst scenario for a pair of robots. Fig. 4.4(b) worst scenario for a robot (grey circle) and a virtual particle (black circle).

by (4.5), where μ_{ij} is defined by (4.8), the robots will never collide with each other.

Proof: The pressure terms of $\mathbf{b}_i$ are bounded and the considered external forces are also bounded. The worst scenario for collision avoidance happens when the robots i and j drive with opposite maximum finite velocities $\mathbf{v}_m$ and $-\mathbf{v}_m$ (see Figure 4.4(a)), respectively, and the pressure terms of $\mathbf{b}_i$ and $\mathbf{f}_i$ assume constant maximum finite values in the direction $\mathbf{q}_{ji}$, and the pressure terms of $\mathbf{b}_j$ and $\mathbf{f}_j$ assume constant maximum finite values in the direction $\mathbf{q}_{ij}$. Since $-\nabla_i W_{ij}$ points in the direction of $\mathbf{q}_{ij}$ and $\Pi_{ij} \geq 0$ we conclude that the term given by the artificial viscosity is a repulsive term. Due to the symmetry we have in this case $\mathbf{v}_i = -\mathbf{v}_j$. Further, by checking the expression of μ_{ij} in (4.8) one should notice that the artificial viscosity term corresponds to a nonlinear damping $-F(\mathbf{q}_i, \mathbf{q}_j)v_i - G(\mathbf{q}_i, \mathbf{q}_j)v_i^2$, where $F \geq 0$ and $G \geq 0$. The artificial viscosity is active in the interval $\|\mathbf{q}_{ij}\| < 2h$, and since $F \to +\infty$ and $G \to +\infty$ when $\|\mathbf{q}_{ij}\| \to (2R + \varepsilon)$ we can guarantee that $\mathbf{v}_i \to 0$ somewhere in the interval $2R + \varepsilon \leq \|\mathbf{q}_{ij}\| < 2h$. Due to symmetry we can guarantee that $\mathbf{v}_{ij} \cdot \mathbf{q}_{ij} \geq 0$ when $\|\mathbf{q}_{ij}\| \to 2R + \varepsilon$. ∎

Proposition 4.6 *Given a single robot, i, represented in its configuration space, with dynamics $\ddot{\mathbf{q}}_i = \mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}, t)$, and the control law determined by (4.5)*

with the adapted term in (4.13) presented in the Subsection 4.2.3 to consider a single virtual particle, where μ_{ij} is defined by (4.8) with $R = 0$, the robot will never collide with the virtual particle.

Proof: Since the robot size is taken into account when building its configuration space we just need to guarantee that $\mathbf{v}_{ip} \cdot (\mathbf{q}_i - \mathbf{p}) \geq 0$ when $\|\mathbf{q}_i - \mathbf{p}\| - \varepsilon \rightarrow 0$, where $\mathbf{p}$ is the virtual particle position in the configuration space. The velocity $\mathbf{v}_p = 0$ and the worst scenario is shown in Figure 4.4(b). We have that $\mathbf{b'}_i$ is determined by (4.13). The arguments of the proof of Proposition 4.5 complete the proof. ∎

Remark 4.4 Basically, we were able to show that the system tends to move toward the desired pattern in an obstacle free environment. However, in general it will converge to an invariant set which may be different from the desired pattern. If the number of robots is too large, for example, it should be expected that some of these robots will not converge. Obtaining proofs for the case of external forces derived from harmonic functions is out of the scope of this work. In this case, the gradients have discontinuities exactly at the desired pattern. For a pair of robots and for a pair robot/virtual particle we were able to prove collision avoidance. Although we do not present a formal proof, we believe that in the presence of more robots and more virtual particles the robots will also avoid collisions. However, it may exist cases where the whole system stops before accomplishing the task in order to avoid collisions.

4.3 Numerical Simulations

In this section we illustrate our approach with ideal simulations developed in C++ computer language [Stroustrup, 2000] and OpenGL [Shreiner et al., 2005]. They are ideal in the sense that no practical issues such as saturation, noise, kinematic constraints, finite size, etc. were implemented. The holonomic point robot abstraction in (4.4) was used. Therefore, we applied the controller in (4.5). Harmonic functions were used as global potential functions. We used the open source software *Triangle* [Shewchuk, 1996] to generate triangular meshes and the software FEMM (Finite Element Method Magnetics [Meeker, 2004]), version 4.0, to proceed the finite element computation.

In Subsection 4.3.1 we present 3 examples where only static obstacles are considered. On the other hand, in Subsection 4.3.2 we present 2 examples where only dynamic obstacles are considered. Videos are available in: http://www.cpdee.ufmg.br/∼lucpim/thesis.

In the next examples, if nothing is said, we used the parameters shown in Table 4.1. The values of ξ_1, ξ_2, and η^2 are recommended values described in [Monaghan, 1992]. Since g is the gravity acceleration its value is a standard value. In these simulations we did not need to use virtual particles and the value of k was tuned to balance the external force term ($\mathbf{f}_i$) and the SPH term ($\mathbf{b}_i$) in the controller (4.5). The damping, ζ, was tuned to avoid intense oscillatory behavior close to the target region. The values of γ and H were experimentally tuned by observing the system behavior during simulations. We verified that these parameters influenced the proximity of the robots during the task execution. It is important to mention that the values of γ and H were not decisive in the accomplishment of the task. The most important parameters to tune are the mass, m_i, and the parameter h. These

Table 4.1: Simulation parameters

Parameter	Value
h	0.05 m
ξ_1	1
ξ_2	2
η^2	$0.01h^2$
γ	1
ζ	50
k	300
m_i	$1000h^2$ Kg
H	1/98 m
g	9.8 m/s^2

parameters are used in the computation of the density and they affect directly how close to each other the robots will be in the final configuration. In our simulations and experiments, the value of mass was chosen to be such that each robot could have a maximum of 4 neighbor robots at distances smaller than or equal to h when the system is in equilibrium, *i.e.*, $\rho_i = \rho_0$.

4.3.1 Static Obstacles - Examples

This subsection is dedicated to present examples where only static obstacles are taken into account.

Example 1

Figure 4.5 presents the first step of our approach for an environment with a rectangular static obstacle. The desired pattern is a circle. The arrows in the figure represent the computed vector field. As it was expected, the field converges to the pattern and diverges from the obstacle and from the external boundaries.

A simulation of 81 robots applying our approach in the environment of

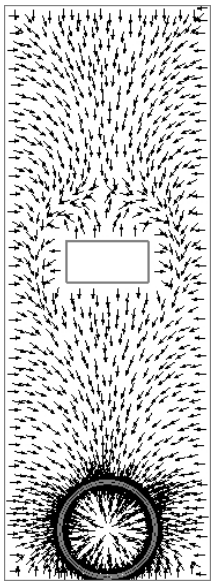


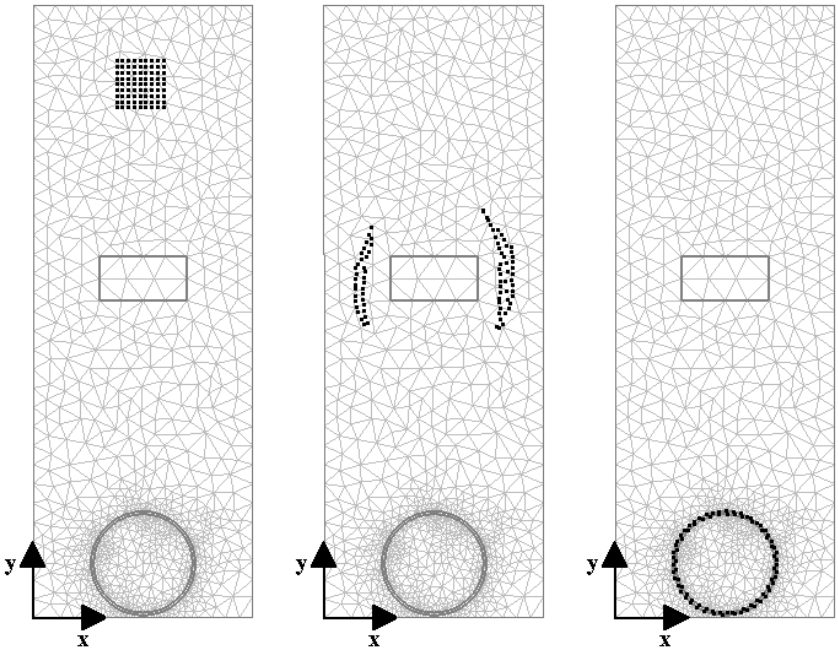
Figure 4.5: Vector field in an environment with a rectangular obstacle.

Figure 4.5 is shown in Figure 4.6. The robots are initially distributed in a 9×9 matrix where the distance between two cells is $d = h/1.2$. When the group comes close to the obstacle it splits into two subgroups due to the presence of a saddle line. Later on, the robots regroup, reach the circle, and distribute themselves along the pattern. The split of the group is due to the symmetry of the problem. Other simulations considering an initial distribution close to y axis presented a different behavior, where all the robots moved together along the left side of the obstacle.

Example 2

Our methodology is also able to treat more complex geometric patterns. Figure 4.7 presents 324 robots, initially distributed in an 18×18 matrix, generating a star shape. This example also shows the possibility of having an initial distribution enclosed by the desired pattern.

Figure 4.8 (a) shows the evolution of the density of a given particle along

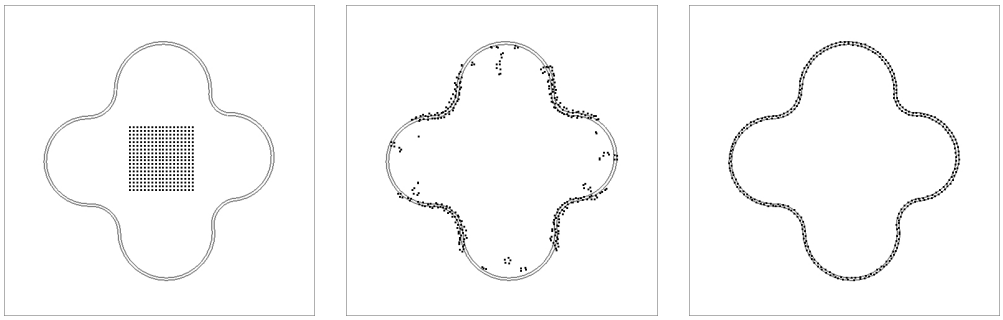


(a) (b) (c)

Figure 4.6: 81 robots generating a circle pattern in the presence of a rectangular static obstacle. (a) Initial distribution. (b) Split. (c) Final distribution.

the simulation iterations. It is interesting to observe that the particle tries to control its density according to the reference density, ρ_0, which is equal to 1000 Kg/m^3 in this case. Such behavior is expected since we are trying to mimic an incompressible fluid. The state equation (2.51), page 39, is responsible for this control by computing positive pressures when the density is larger than the reference and negative pressures otherwise. Clearly, if the density is equal to the reference the pressure is forced to zero.

Figure 4.8 (b) presents a histogram of the density of all particles along all the iterations. One could expect that the histogram would be actually centered at 1000. This did not occur due to the iterations in which the robots were located inside the pattern region. The pattern region is not thick enough which causes the swarm to be compressed inside it. It is important



(a) (b) (c)

Figure 4.7: 324 robots generating a star shape. (a) Start. (b) Intermediate instant. (c) End.

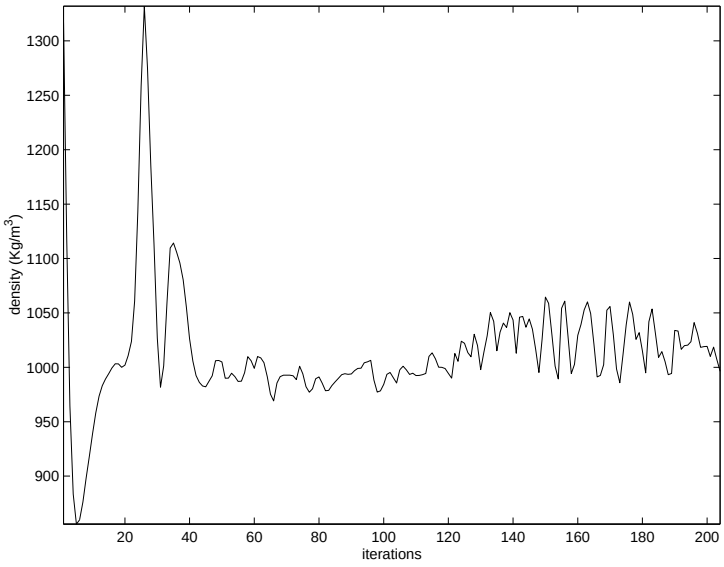
to mention that this compression is responsible for distributing the robots along the curve after reaching it. Inside the target region the vector field is equal to zero and the term (4.6) causes the distribution of the robots in this region mainly influenced by the values of pressure. Since the density values are larger in this region, the pressure values are positive and the robots tend to repel each other, which causes the swarm to be homogeneously distributed along the pattern.

Example 3

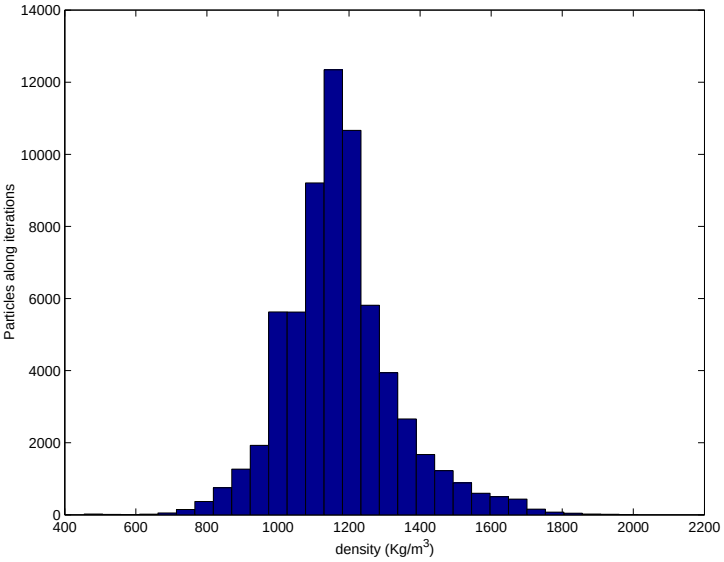
The proposed approach can also be used in geometric complex environments such as a maze. A simulation with 121 robots in a maze like environment is presented in Figure 4.9.

4.3.2 Dynamic Obstacles - Examples

In this subsection we present examples of environments with dynamic obstacles.



(a)



(b)

Figure 4.8: Convergence to the reference density. (a) Density of one particle along all the iterations. (b) Histogram of the density of all particles along all the iterations.

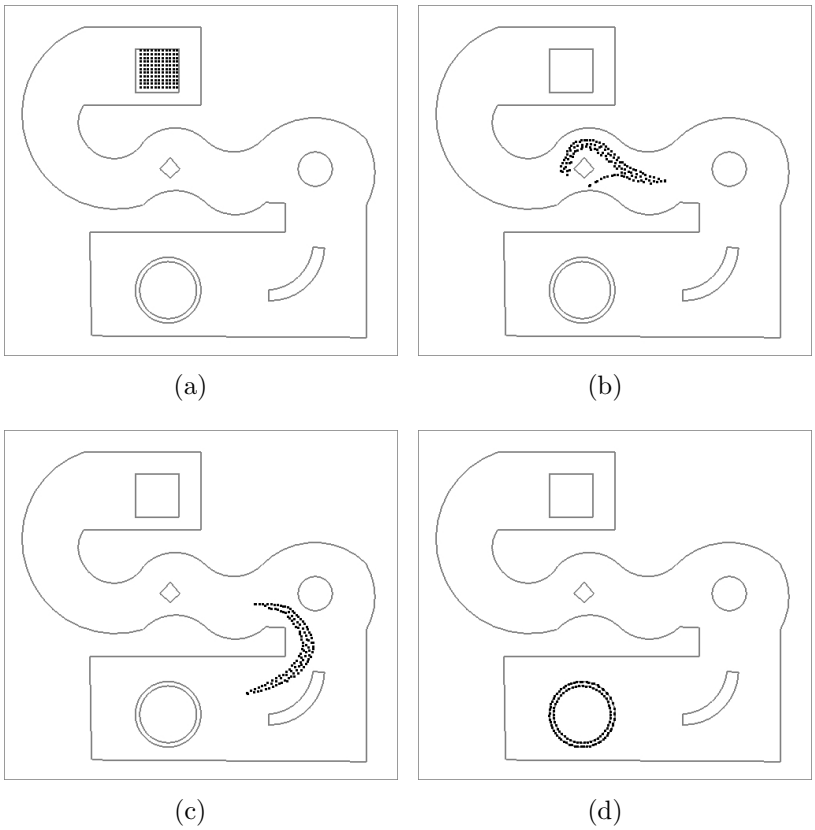


Figure 4.9: Simulation with 121 point robots from a starting configuration (a) to the goal (d), with intermediate configurations (b) and (c).

Example 4

Figures 4.10 and 4.11 present different time samples of a simulation that considers 3 dynamic particle obstacles. These obstacles are represented by stars (*) and are both unknown a priori and mobile. Two of these obstacles have periodic horizontal movement and one of them has a periodic vertical movement. In this case we used the very dense particles idea (see Subsection 4.2.3). We used $\sigma = 10$ to compute the mass of the particle obstacles. The robots are initially distributed in a 14×14 matrix.

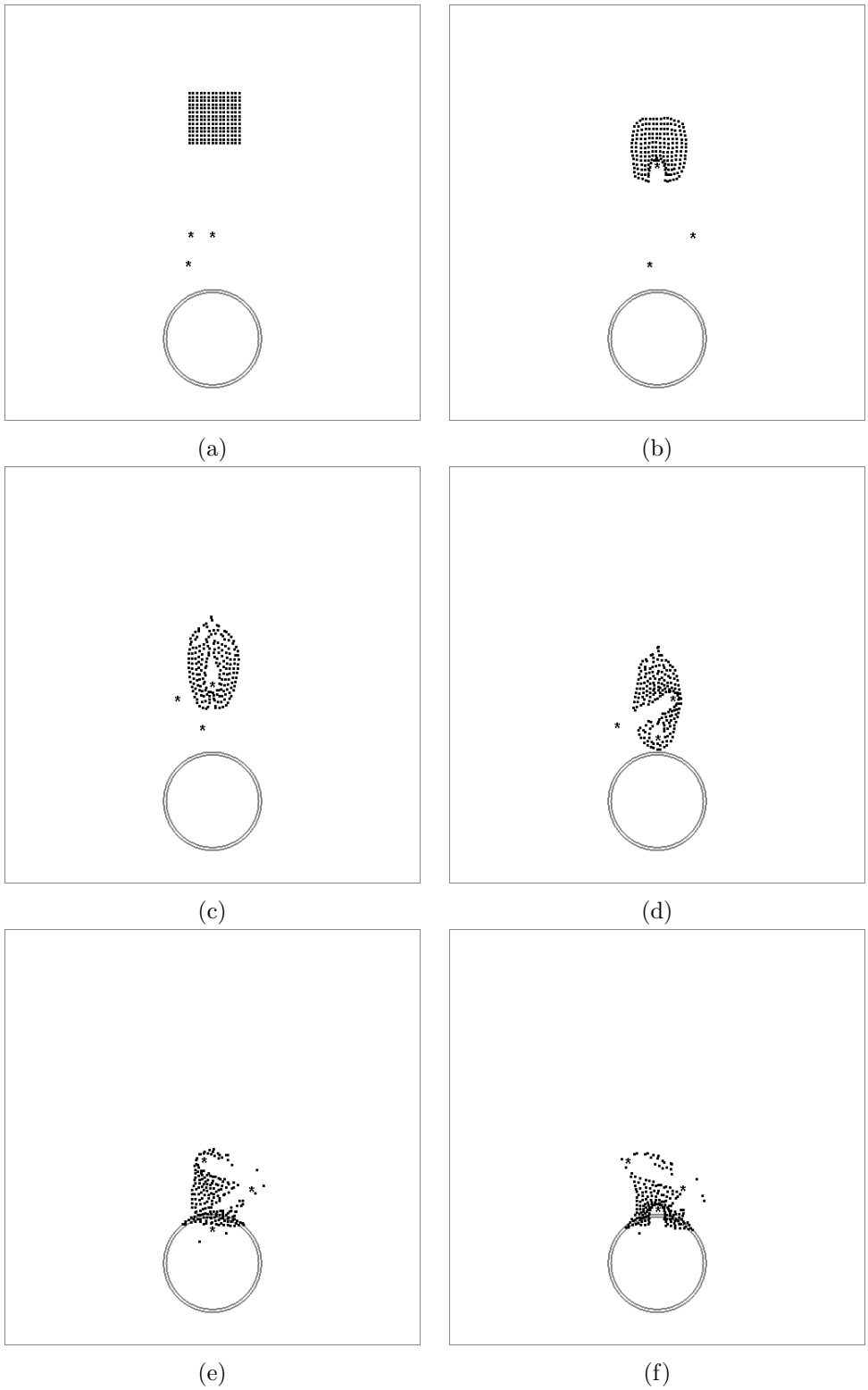
From Figures 4.10 and 4.11 one can conclude that the idea of considering

dynamic obstacles as very dense particles is feasible. When a particle obstacle reaches the range of action of a swarm particle, *i.e.*, when the distance between these two particles is smaller than $2h$, the swarm particle suffers the influence of a virtual repulsive force and the obstacle is properly avoided. It is interesting to note that even in the target region the particles are still able to avoid the dynamic obstacles.

Example 5

Figure 4.12 shows the evolution of 169 robots generating a circle pattern in the presence of a dynamic square obstacle. The obstacle boundary is discretized into dense particles where each edge is represented by 10 particles equally spaced. The distance between neighbor obstacle particles in the same edge is equal to $h/2 = 0.025$. The mass of these particles was computed using $\sigma = 10$. Besides, in this simulation we used $k = 500$.

It can be observed that the robots properly avoided the square obstacle without any attempt to penetrate the obstacle. It can also be observed that two particles were trapped by the dynamic obstacle and were not able to reach the target region. Although this may be an issue, it is important to note that the task was accomplished independently of the two missing robots. Therefore, we can conclude that the methodology has a certain level of fault-tolerance. Obviously, depending on the geometry and the velocity of a given dynamic obstacle, all the robots could be trapped preventing the task accomplishment. However, this fact is inherent to most methodologies that use only local information to take decisions. In this example, we stopped the simulation before the robots generate the complete circle pattern because our simulation environment was not prepared to have robots going out of the simulation domain.



(a) (b) (c) (d) (e) (f)

Figure 4.10: 196 robots generating a circle pattern in the presence of dynamic particle obstacles - Part I.

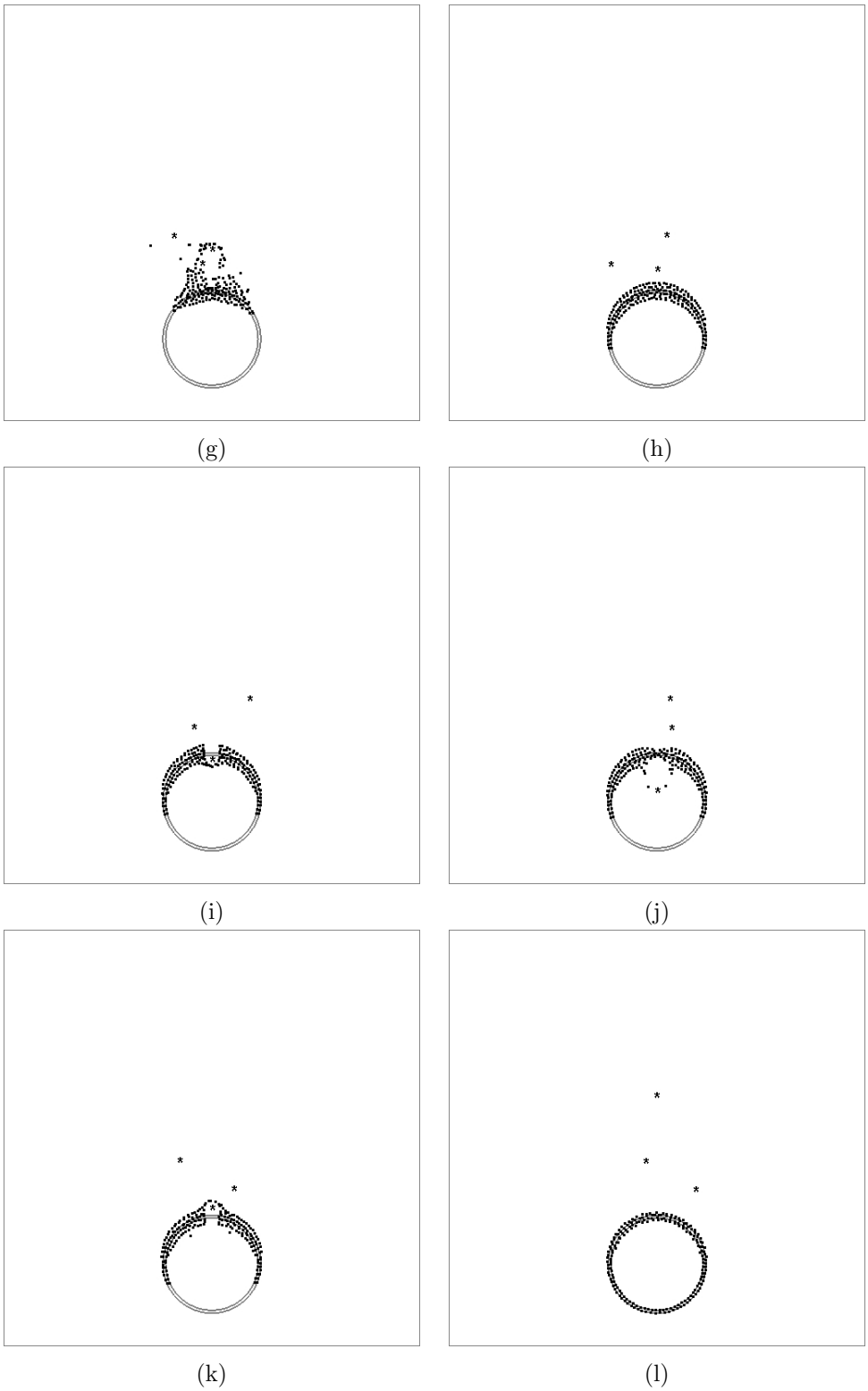


Figure 4.11: 196 robots generating a circle pattern in the presence of dynamic particle obstacles - Part II.

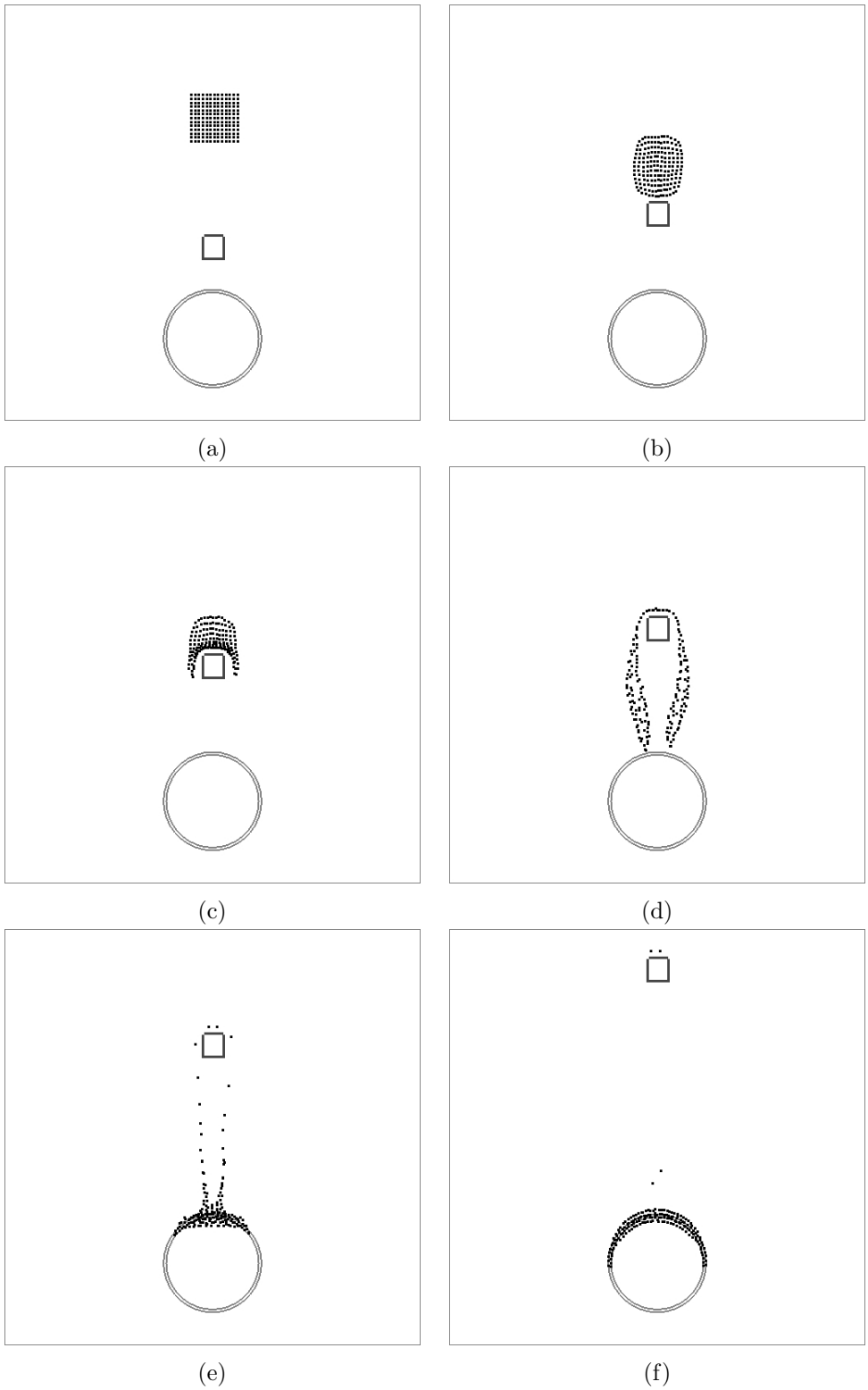


Figure 4.12: 169 robots generating a circle pattern in the presence of a dynamic square obstacle. The simulation was stopped before the particles generate the complete pattern.

4.4 Experimental Results

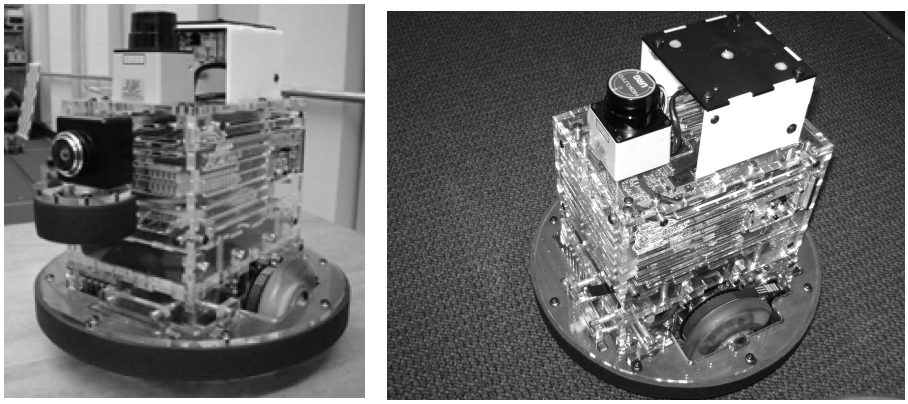
In this section, we present experimental results that verify the proposed approach for finite size and nonholonomic robots. Movies can be seen on the web page http://www.cpdee.ufmg.br/∼lucpim/thesis.

During the experimental phase we used the testbed for multi-robot experiments available in the GRASP (General Robotics, Automation, Sensing, and Perception) Laboratory at the University of Pennsylvania, USA. This testbed is composed by differential drive, kinematically controlled robots called *Scarabs* (see Figs. 4.13(a) and 4.13(b)) and an overhead tracking system that provides pose information in a global reference frame or to be used as ground truth.

Each Scarab is equipped with an onboard computer (nano-itx 1GHz), power management system, wireless communication, and is actuated by stepper motors. The system is plug and play in the sense that sensors and actuators can be added or removed. Typical sensors used in this platform are Hokuyo URG laser range finder and Point Grey Firefly IEEE 1394 camera. There is also a foam bumper for protection.

The tracking system consists of seven IEEE 1394 cameras, computers, and tracking targets at the top of the robots (see Fig. 4.13(b)). The tracking target is composed by three LEDs that flash an 8 B identification pattern which is then detected by the camera system. An extended Kalman filter (EKF) is used to fuse the information from multiple cameras. An EKF also runs in each robot to fuse the local odometry and the overhead tracking information. This system has been successfully used to track tens of robots simultaneously with a position error of approximately 2 cm and an orientation error of 5^o at 29 Hz in a $9 \times 6 \times 6m^3$ room.

At the software level, the GRASP platform uses the open-source software

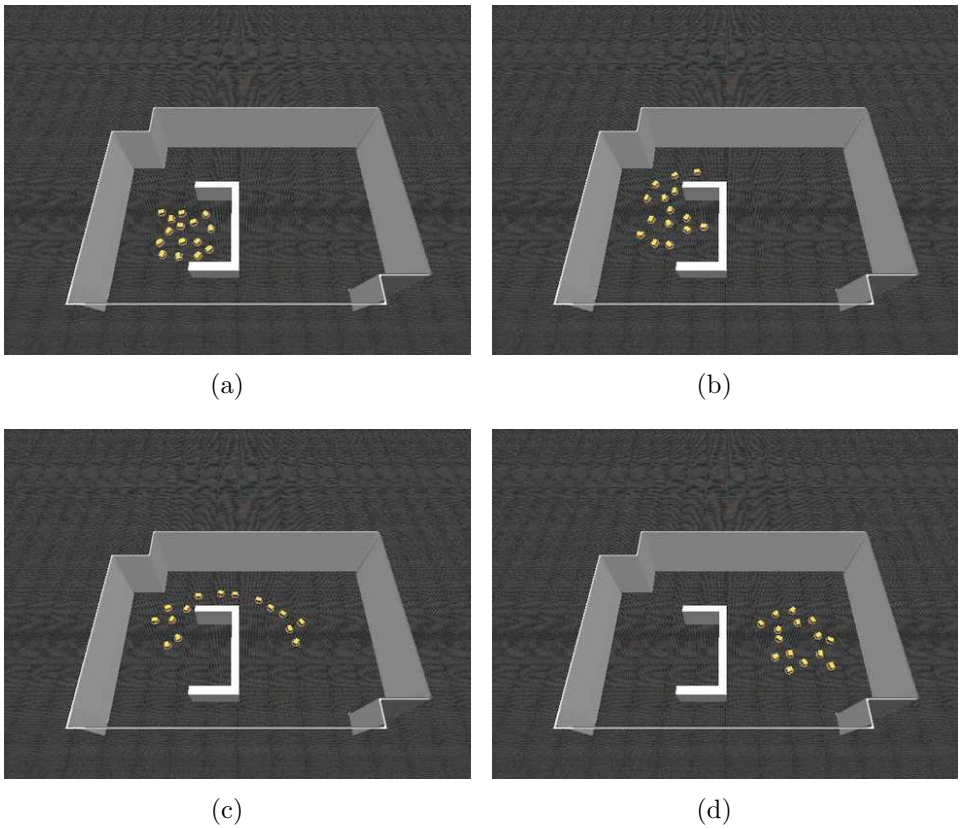


(a) (b)

Figure 4.13: The 20 × 13.5 × 22.2 cm^3 differential drive robotics platform (a) with camera (b) without camera.

developed by the Player/Stage/Gazebo (PSG) project [Gerkey et al., 2003]. Player is a network server for robot control. It provides the interfaces such that one can easily have access to the robot's sensors and actuators over an IP network. The fluid based approach was successfully implemented in the given testbed. The algorithms were coded in C++ in the form of a Player driver. A driver is a piece of software that talks to a robotic sensor, actuator, or algorithm, and translates its inputs and outputs to conform to one or more interfaces. Such module runs locally in each robot and is able to send and receive data to other drivers running locally or across the network. Further details of this infrastructure are described in [Michael et al., 2008].

Before running real experiments, simulations using the 3D environment GAZEBO [Gerkey et al., 2003] were implemented. Such environment allows for simulating conditions that are very close to the real world ones. We used robot models that capture the geometry, kinematics, and dynamics of the Scarab robots. Moreover, we considered a virtual world with the same



(a) (b)

(c) (d)

Figure 4.14: Simulation with 15 robots from a starting configuration (Fig. 4.14(a)) to the goal (Fig. 4.14(d)), with intermediate configurations (Figs. 4.14(b) and 4.14(c)).

dimensions of the area where the experiments were performed. The obstacles and the goal were also the same. The goal formation was a circular pattern located behind a U-shaped obstacle. A simulation trial with 15 robots is presented in Fig. 4.14.

In the experiments we used a team of seven *Scarabs*. The team of robots were provided with a map of the environment which was defined by the boundaries of the experimental area and static obstacles. A vector field based on harmonic functions was computed off-line. Each robot computed its location in the map based on localization information from the overhead

tracking system and current velocity from its motor controller. This information was broadcasted over the network to the other robots. To emulate the concept of neighborhoods in the smaller experimental area, each robot ignored messages from robots a distance greater than 2 m. At every update of the control algorithm, each robot computed its current SPH state based on its local information and the information received over the network. Additionally, each robot used virtual particles based on the map within a region $2h' \times 2h'$, with $h' = 0.3$ m. During the experimentation, the algorithm was found to be more robust when multiple virtual particles were defined. Each virtual particle was assigned to each cell with obstacle in a local occupancy grid (see Figure 4.3, page 62).

A vignette of one trial run of the implementation is shown in Fig. 4.15. The team of robots was started in an initial configuration (Fig. 4.15(a)) and provided a circular goal formation. The control law was executing at an update rate of 10 Hz while the inter-agent network communications were executing at a higher rate of 20 Hz to accommodate the asynchronous system design. One can see that two robots did not converge to the goal. This was due to limitations of the minimum allowed velocities of the actual robots. Since one of the robots that had already reached the goal was not moving, the two other ones could not converge.

4.5 Environment Coverage

In the previous sections we showed how our fluid based approach can be used to solve the pattern generation task. Due to all the interesting properties of the proposed solution, such as decentralization, density regulation, and collision avoidance, we believe that our approach can be used successfully

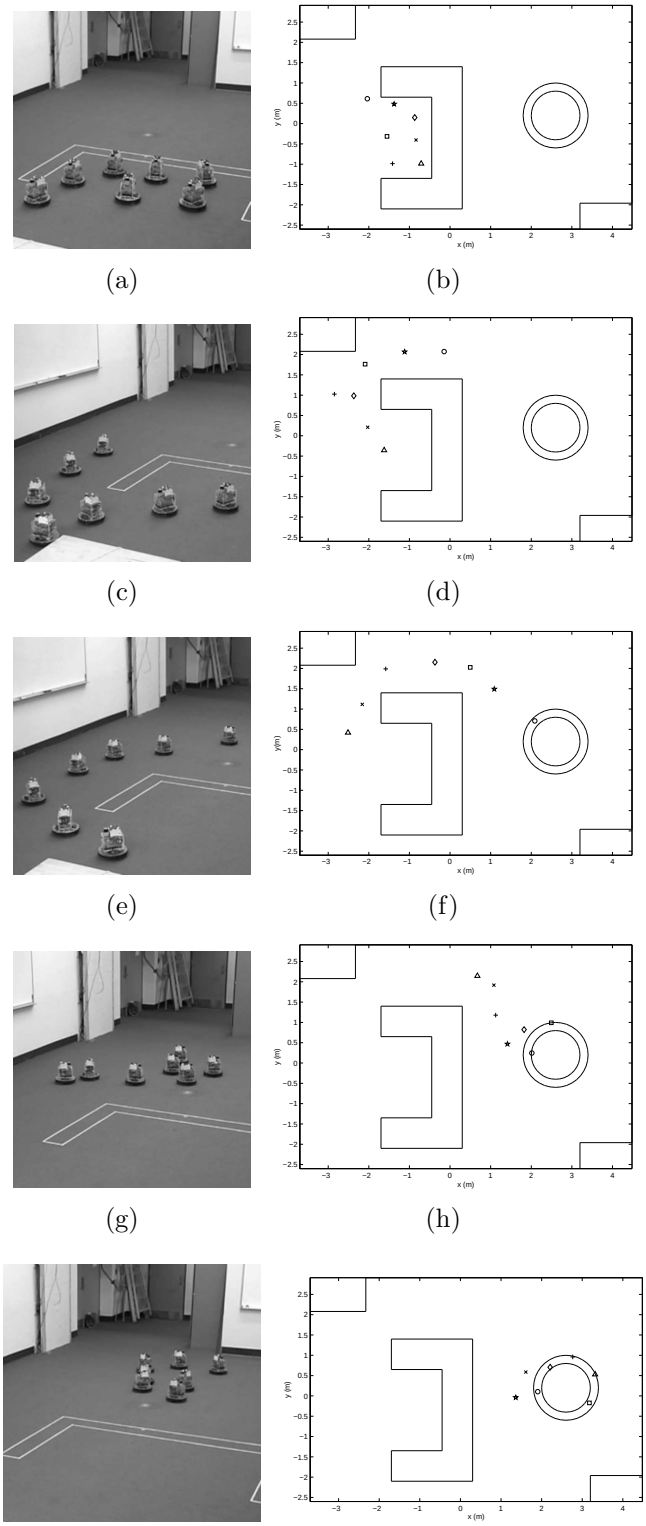


Figure 4.15: Experimental Results. A team of seven robots, starting from an initial configuration (Fig. 4.15(a)), control around an obstacle (Figs. 4.15(c), 4.15(e), and 4.15(g)) to a goal circular formation (Fig. 4.15(i)). The results in the configuration space are also shown in Figs. 4.15(b), 4.15(d), 4.15(f), 4.15(h), and 4.15(j).

in other tasks. In this section, we propose a solution to the problem of deploying a network of mobile sensors in an unknown environment. As defined in [Howard et al., 2002], a *mobile sensor network* is a distributed collection of *nodes* such that each node has sensing, computation, communication, and locomotion capabilities. In fact, we are considering teams of robots equipped with sensors. According to the nomenclature introduced in [Gage, 1992], we are interested in a *blanket coverage* where the main objective is to maximize the total detection area.

As mentioned in [Poduri and Sukhatme, 2004], there are certain applications where it is not only necessary to provide good sensor coverage, but also to satisfy certain local (e.g. node degree[2]) and global (e.g. network connectivity) constraints. One can define the *Constrained Coverage Problem* [Howard et al., 2002], informally, as the problem of deploying a sensor network in a configuration that maximizes the collective sensor coverage and at the same time satisfies one or more constraints. Solutions to this coverage problem have several applications such as environmental monitoring (e.g. for chemical and nuclear spills, forest fires, etc.), mapping spatial distributions (e.g. temperature, pressure, etc.), and surveillance. In [Poduri and Sukhatme, 2004], for example, a virtual potential field based approach is proposed to solve a constrained coverage problem. In this case, a repulsive force among agents is used to maximize coverage and an attractive force is used to impose a constraint of K-degree. This means that each node should have at least K neighbors. Two nodes are considered neighbors if the Euclidean distance between them is less than or equal to the communication range R_c. Our objective is to find decentralized controllers which solve a

[2]Given an agent i of the team, its node degree corresponds to the number of other agents that are able to interact with i.

different type of *Constrained Coverage Problem*.

Let Ω be an unknown environment where we wish to deploy our network. We are interested in 2-dimensional and 3-dimensional environments such that $\Omega \subset \mathbb{R}^2$ or $\Omega \subset \mathbb{R}^3$. In principle, we do not impose any geometric constraint in Ω.

Since we already showed how our approach can be extended to deal with practical robot issues, we will assume the holonomic point abstraction (see Subsection 4.2.2) in this section. The system configuration is defined by $\mathbf{q} = [\mathbf{q}_1^T, \ldots, \mathbf{q}_N^T]^T$, where $\mathbf{q}_i$ gives the cartesian coordinates $[x_i, y_i]^T$ or $[x_i, y_i, z_i]^T$, depending on the environment dimension, of agent i. As in the previous section, we consider mobile network nodes with second order dynamics:

$$\ddot{\mathbf{q}}_i = \mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}, t). \tag{4.34}$$

We assume that each agent is able to: (i) compute its own configuration $\mathbf{q}_i$ and velocity $\mathbf{v}_i = \dot{\mathbf{q}}_i$; (ii) communicate with other agents in a range R_c; and (iii) detect obstacles.

The incompressible fluid model presented in the last section has the ability to control the *swarm density*. Controlling the density may be seen as a loose way of controlling the network connectivity. Therefore, we define the *Density Constrained Coverage Problem*:

Problem 4.2 *Find decentralized controllers to deploy a mobile sensor network such that the resulting configuration maximizes the net sensor coverage and the net density is larger than or equal to a reference density, ρ_0.*

We use the same concept of density as before, given by equation (2.45):

$$\rho_i = \sum_{j \in \mathcal{N}_i} m_j W(\mathbf{q}_i - \mathbf{q}_j, h)\,, \tag{4.35}$$

where $\mathcal{N}_i$ is the neighborhood of agent i:

$$\mathcal{N}_i = \{j \neq i | \|\mathbf{q}_j - \mathbf{q}_i\| < R_c\}\,, \tag{4.36}$$

where R_c determines the parameter h. This is the parameter that defines the support size of W. Therefore, we have $R_c = 2h$ for the cubic spline in (2.36).

We propose to solve this coverage problem by using a control law which is similar to the one used before, with external forces equal to zero:

$$\mathbf{u}_i(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{b'}_i - \zeta \mathbf{v}_i\,, \tag{4.37}$$

where

$$\mathbf{b'}_i = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij}\right) \nabla_i W_{ij}(h) - \lambda \Pi_{ip} \nabla_i W_{ip}(h')\,, \tag{4.38}$$

ζ is a positive damping constant, λ is a positive constant, and p refers to the virtual particles (see Subsection 4.2.3).

Figure 4.16 presents a simulation where a team of twenty robots cover a building-like environment with a reference density $\rho_0 = 1000$ Kg/m^3. The start configuration is showed in Figure 4.16(a), intermediate configurations in Figures 4.16(b), 4.16(c), 4.16(d), 4.16(e), 4.16(f), 4.16(g), and the final configuration in Figure 4.16(h). Figures 4.17(a) and 4.17(b) present the evolution over iterations of the mean density value and the density standard deviation of the team, respectively. Therefore, we conclude that the system converges to the desired density.

Although the proposed approach solves an interesting coverage problem, in some applications we want to be able to say something about the optimality of the obtained coverage. Such property is hard to prove for the

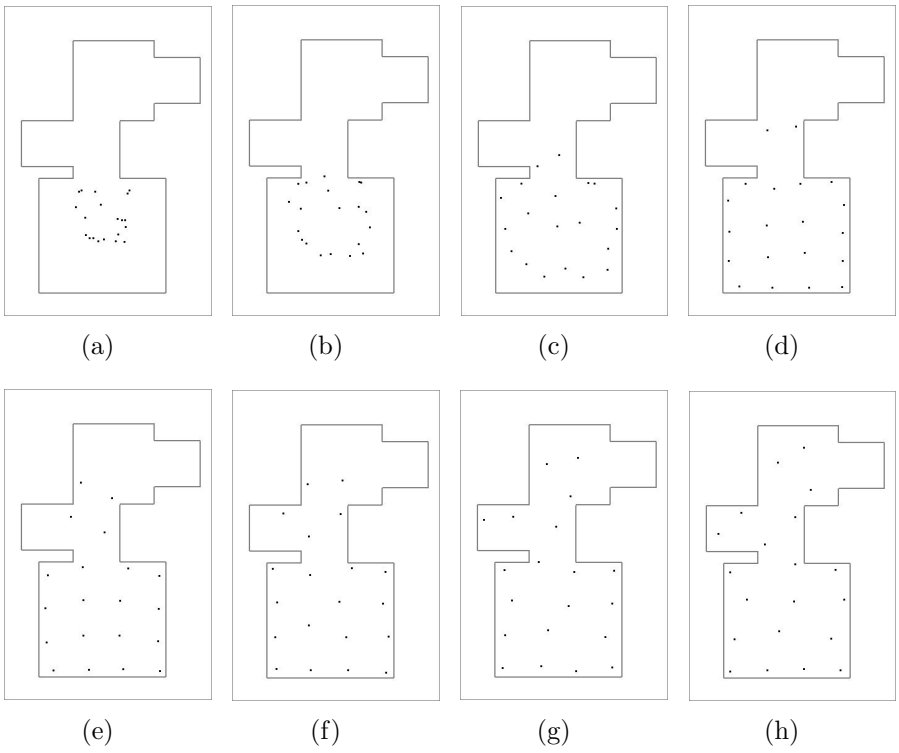
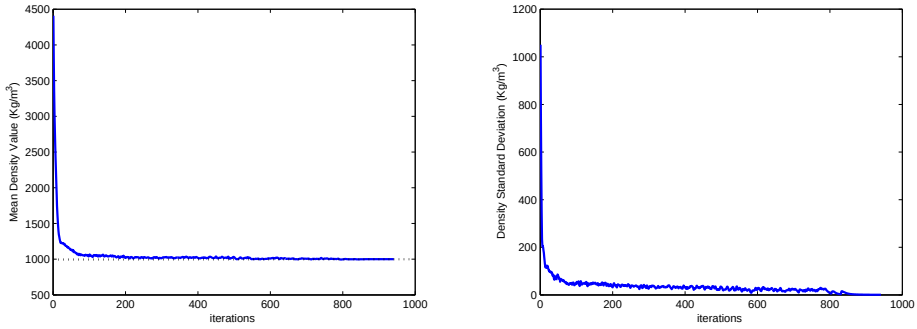


Figure 4.16: Coverage simulation result. A team of twenty robots cover a building-like environment from a starting configuration 4.16(a) to a final configuration 4.16(h), with intermediate configurations 4.16(b), 4.16(c), 4.16(d), 4.16(e), 4.16(f), and 4.16(g).

proposed SPH based approach. In the next chapter we present a different methodology, based on locational optimization tools, for optimal deployment of mobile sensor networks. This locational optimization based approach also keeps one of the most interesting functionalities of the SPH based approach which is the ability to control the density of agents over the environment. In fact, this approach defines a density function which allows, for example, to concentrate more agents in a given portion of the environment. Although, in the examples presented in the current chapter, we did not apply a fluid model with different values of density over the domain, we believe that our SPH controllers are also able to deal with such a model. This topic is left as



(a) (b)

Figure 4.17: Density of the team over iterations. (a) Mean density. (b) Density standard deviation.

a promising future direction of research.

Chapter 5

Coverage Control Based on Locational Optimization Tools

A distributed and asynchronous approach for optimal coverage of a domain with identical mobile sensing agents is proposed in [Cortez et al., 2004] based on a framework for optimized quantization derived in [Lloyd, 1982]. Each agent (robot) follows a control law, which is a gradient descent algorithm that minimizes a functional encoding the quality of the sensing coverage. Further, this control law depends only on the information of position of the robot and of its immediate neighbors. Neighbors are defined to be those robots that are located in neighboring Voronoi cells. Besides, these control laws are computed without the requirement of global synchronization. The functional also uses a *distribution density function* which weights points or areas in the environment that are more important than others. Thus, it is possible to specify areas where a higher density of agents is required. Furthermore, this technique is adaptive due to its ability to address changing environments, tasks, and network topology.

In this chapter we propose three important extensions to the work in [Cortez et al., 2004]. First, we address the problem of incorporating heterogeneity in the robot team by allowing the robots to have different types of sensors. This first extension is actually a minor contribution since we

use power diagrams, similarly to [Kwok and Martínez, 2008], with a different motivation. Second, we overcome the practical limitations of the point robot assumption in the original algorithm. Finally, we generalize the basic method to nonconvex environments. To the best of our knowledge, the last two extensions are not similar to any other extension found in the literature.

In the next section we present the main aspects of the basic method in [Cortez et al., 2004] derived from the *Locational Optimization Framework* [Okabe et al., 2000], using a distance function that is independent of the Euclidean metric.

5.1 Locational Optimization Framework

Let $\Omega \subset \mathbb{R}^N$ be a given representation of the environment, $\mathcal{P} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ be the configuration of n mobile sensors, where $\mathbf{q}_i \subset \Omega$, and $T = \{T_1, \ldots, T_n\}$ be a tessellation of Ω such that $I(T_i) \cap I(T_j) = \emptyset$, $\forall i \neq j$, where $I(\cdot)$ represents the interior of a given region, and $\cup_{i=1}^{n} T_i = \Omega$. The key idea is that each agent i is responsible for the coverage of the region T_i. As a measure of the system performance we define the *coverage functional*:

$$\mathcal{H}(\mathcal{P}, T) = \sum_{i=1}^{n} \mathcal{H}(\mathbf{q}_i, T_i) = \sum_{i=1}^{n} \int_{T_i} f(d(\mathbf{q}, \mathbf{q}_i))\varphi(\mathbf{q}) d\mathbf{q}\,, \tag{5.1}$$

where d corresponds to a function that measures distances between locations $\mathbf{q} \in \Omega$ and sensors. Note that we do not require that this function defines a metric in Ω. The function $\varphi : \Omega \to \mathbb{R}_+$ is a distribution density function which defines a weight for each point in Ω. The density function may reflect a knowledge of the probability of occurrence of events in different regions, or simply a measure of relative importance of different regions in Ω. Therefore, points with greater weight values should be better covered than points

with smaller values. The function $f : \mathbb{R} \to \mathbb{R}$ is a smooth strictly increasing function over the range of d that measures the degradation of sensing performance with distance. We assume that $\mathcal{H}(\mathcal{P}, T)$ is differentiable. The problem of covering the environment, Ω, is then translated to the problem of minimizing the functional in (5.1).

5.1.1 Centroidal Voronoi Tessellation

An important tool in the Locational Optimization theory is the *Voronoi tessellation*. According to Section 2.9, given the set of points $\mathcal{P} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$, often called *sites*, distributed over the bounded domain Ω, with boundary $\partial\Omega$, we define the Voronoi region, or Voronoi cell, V_i, associated to the point $\mathbf{q}_i$ according to a given distance function d as:

$$V_i = \{\mathbf{q} \in \Omega | d(\mathbf{q}, \mathbf{q}_i) \leq d(\mathbf{q}, \mathbf{q}_j), \forall j \neq i\}. \tag{5.2}$$

The generalized Voronoi tessellation of the set $\mathcal{P}$, $V(\mathcal{P})$, is the collection of such regions. The Voronoi boundary ∂V_i is defined as:

$$\partial V_i = \cup_{j=1}^{n} l_{ij} \cup \{\partial\Omega \cap V_i\}. \tag{5.3}$$

where l_{ij} is the bisector:

$$l_{ij} = \{\mathbf{q} \in \Omega | d(\mathbf{q}, \mathbf{q}_i) = d(\mathbf{q}, \mathbf{q}_j), j \neq i\}. \tag{5.4}$$

Given a robot i we define the neighborhood of i, $\mathcal{N}_i$, as the set of robots that share Voronoi boundaries with V_i:

$$\mathcal{N}_i = \{j \in \mathcal{P} | \partial V_i \cap \partial V_j \neq \emptyset\}. \tag{5.5}$$

The next two propositions relate Voronoi tessellations with the minimization of the objective functional in (5.1) for a general distance function, d. The proofs follow the same arguments, used by [Du et al., 1999] and [Schwager et al., 2006] in the case of the Euclidean distance.

Proposition 5.1 *A necessary condition for a minimizer of the objective functional in (5.1) is that the T tessellation corresponds to the Voronoi tessellation, $V(\mathcal{P})$, according to the distance function d.*

Proof: Let $\hat{V}$ be another tessellation than the Voronoi V. For a given point $\mathbf{q} \in V_i$, and $\mathbf{q} \in \hat{V}_j$, where $\mathbf{q}_k \in \hat{V}_j \Leftrightarrow k = j$, we can write:

$$f(d(\mathbf{q}, \mathbf{q}_i))\varphi(\mathbf{q}) \leq f(d(\mathbf{q}, \mathbf{q}_j))\varphi(\mathbf{q}). \tag{5.6}$$

Since $\hat{V}$ is not a Voronoi tessellation, the inequality in (5.6) will hold strictly over some measurable set of Ω. Therefore:

$$\mathcal{H}(\mathcal{P}, V) < \mathcal{H}(\mathcal{P}, \hat{V}).$$

■

Assuming that T is determined by the Voronoi tessellation of the points in $\mathcal{P}$ then $\mathcal{H}(\mathcal{P}, T) = \mathcal{H}(\mathcal{P}, V(\mathcal{P})) = \mathcal{H}(\mathcal{P})$ and we have the following result

Proposition 5.2 *A necessary condition for $\mathcal{H}(\mathcal{P})$ to be minimized is:*

$$\frac{\partial \mathcal{H}(\mathcal{P})}{\partial \mathbf{q}_i} = \frac{\partial \mathcal{H}(\mathbf{q}_i, V_i)}{\partial \mathbf{q}_i} = \int_{V_i} \frac{\partial}{\partial \mathbf{q}_i} f(d(\mathbf{q}, \mathbf{q}_i))\varphi(\mathbf{q}) d\mathbf{q} = 0. \tag{5.7}$$

Proof: By applying the differentiation under the integral sign (see [Flanders, 1973])

we can write

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial x_i} =& \frac{\partial}{\partial x_i}\int_{V_i} f(d(\mathbf{q},\mathbf{q}_{i_0}))\varphi(\mathbf{q})d\mathbf{q}|_{x_i=x_0} + \int_{V_{i_0}} \frac{\partial}{\partial x_i} f(d(\mathbf{q},\mathbf{q}_i))\varphi(\mathbf{q})d\mathbf{q} \\ &+ \sum_{j\in\mathcal{N}_i} \frac{\partial}{\partial x_i}\int_{V_j} f(d(\mathbf{q},\mathbf{q}_j))\varphi(\mathbf{q})d\mathbf{q}|_{x_i=x_0} \\ =& \int_{\partial V_i} f(d(\mathbf{q},\mathbf{q}_{i_0}))\varphi(\mathbf{q})\frac{\partial(\partial V_i)}{\partial x_i}\cdot\mathbf{n}_i ds + \int_{V_{i_0}} \frac{\partial}{\partial x_i} f(d(\mathbf{q},\mathbf{q}_i))\varphi(\mathbf{q})d\mathbf{q} \\ &+ \sum_{j\in\mathcal{N}_i}\int_{\partial V_j} f(d(\mathbf{q},\mathbf{q}_j))\varphi(\mathbf{q})\frac{\partial(\partial V_j)}{\partial x_i}\cdot\mathbf{n}_j ds\,,\end{aligned}$$

where $\mathbf{q}_i = [x_i, y_i]^T$ in two dimensions, $\mathbf{q}_{i_0}$ is a fixed configuration of agent i, V_{i_0} is the Voronoi region associated to $\mathbf{q}_i = \mathbf{q}_{i_0}$, $\mathbf{n}_i$ and $\mathbf{n}_j$ are the outward facing unit normals of ∂V_i and ∂V_j respectively, and ds is the element of arc length. At the bisector we have $\partial V_i = \partial V_j = l_{ij}$ and $\mathbf{n}_j = -\mathbf{n}_i$. Moreover, since $\partial V_i = \{\cup_{j\in\mathcal{N}_i} l_{ij}\} \cup \{\partial\Omega \cap V_i\}$, and $\frac{\partial(\partial V_j)}{\partial x_i} = 0$ at $\partial\Omega \cap V_i$, we have

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial x_i} =& \sum_{j\in\mathcal{N}_i}\int_{l_{ij}} [f(d(\mathbf{q},\mathbf{q}_{i_0})) - f(d(\mathbf{q},\mathbf{q}_j))]\varphi(\mathbf{q})\frac{\partial(\partial V_i)}{\partial x_i}\cdot\mathbf{n}_i ds \\ &+ \int_{V_{i_0}} \frac{\partial}{\partial x_i} f(d(\mathbf{q},\mathbf{q}_i))\varphi(\mathbf{q})d\mathbf{q}.\end{aligned}$$

Due to the property in (5.4), $d(\mathbf{q},\mathbf{q}_{i_0}) = d(\mathbf{q},\mathbf{q}_j)$ at l_{ij}. Clearly, $\frac{\partial \mathcal{H}}{\partial y_i}$ can be obtained similarly to $\frac{\partial \mathcal{H}}{\partial x_i}$. Therefore, we conclude that

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} = \int_{V_i} \frac{\partial}{\partial \mathbf{q}_i} f(d(\mathbf{q},\mathbf{q}_i))\varphi(\mathbf{q})d\mathbf{q}\,, \tag{5.8}$$

which must be equal to zero at a minimum point. ∎

In [Cortez et al., 2004] the Euclidean distance is used as d, and $f(d) = d^2$. Moreover, since it is assumed a convex environment it is easy to prove that all Voronoi cells are convex polytopes. In this case the necessary configuration

to be at a minimum is obtained when each agent is located exactly at the *centroid* of its own Voronoi cell. The centroid is given by:

$$\mathbf{q}_i^* = \frac{\int_{V_i} \mathbf{q}\varphi(\mathbf{q})d\mathbf{q}}{\int_{V_i} \varphi(\mathbf{q})d\mathbf{q}}. \tag{5.9}$$

Similarly, we can define a *generalized centroid* for general f and d functions, as follows:

$$\mathbf{q}_i^* = \arg\min_{\mathbf{q}_i \in V_i} \int_{V_i} f(d(\mathbf{q}, \mathbf{q}_i))\varphi(\mathbf{q})d\mathbf{q}. \tag{5.10}$$

According to Propositions 5.1 and 5.2, every robot must be driven to the generalized centroid of its Voronoi region to minimize the functional (5.1). The resulting partition of the environment is commonly called *Centroidal Voronoi Tessellation* (CVT).

5.1.2 Continuous-Time Lloyd Algorithm

A classic discrete-time method to compute CVT's is the Lloyd's algorithm [Lloyd, 1982]. In each iteration this method executes three steps: (i) compute the Voronoi regions; (ii) compute the centroids; (iii) move each point site to the corresponding centroid.

In [Cortez et al., 2004] a continuous-time version of this approach is proposed for kinematic models: $\dot{\mathbf{q}}_i = \mathbf{u}_i$. The following control law guarantees that the system converges to a CVT:

$$\mathbf{u}_i = -k(\mathbf{q}_i - \mathbf{q}_i^*)\,, \tag{5.11}$$

where k is a positive gain. The control law is a gradient-descent approach,

since if d is the Euclidean distance, $f(d) = d^2$, and considering (5.9):

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} = 2\left(\int_{V_i} \varphi(\mathbf{q}) d\mathbf{q}\right)(\mathbf{q}_i - \mathbf{q}_i^*).$$

It is important to mention that $\mathcal{H}$ is nonconvex, which implies that the system will in general converge to a CVT that corresponds to a local minimum. In the rest of the paper we present possible extensions of the method proposed in [Cortez et al., 2004].

5.2 Heterogeneous Robots

In this section we consider the problem of deploying a team of agents with heterogeneous sensing capabilities to cover an environment. We capture this heterogeneity by modelling each robot (sensor) as a circle (in $\mathbb{R}^2$) $B_i(\mathbf{q}_i, R_{\mathbf{q}_i})$, where $\mathbf{q}_i$ is the center position and $R_{\mathbf{q}_i}$ is the radius. These models address omnidirectional sensors that may have limited range and may have the quality of the measured information only acceptable in the region B_i. Our robots are heterogeneous in that the sensor footprints are different. Although the points inside B_i have acceptable quality, this does not necessarily mean that the quality is uniform. The performance may degrade with the distance from the center of B_i. Mathematically, we describe the team task as the minimization of the functional:

$$\mathcal{H}(\mathcal{P}, PV) = \sum_{i=1}^{n} \int_{PV_i} [\|\mathbf{q} - \mathbf{q}_i\|^2 - R_{\mathbf{q}_i}^2] \varphi(\mathbf{q}) d\mathbf{q}\,, \tag{5.12}$$

where $f(d) = d$ and $d(\mathbf{q}, \mathbf{q}_i) = \|\mathbf{q} - \mathbf{q}_i\|^2 - R_{\mathbf{q}_i}^2$ is the so-called *power distance* [Aurenhammer, 1987]. According to Proposition 5.1 the required tessellation must then be a Voronoi partition according to the power dis-

tance. The resulting tessellation is well-known in the literature and it is often called the *Voronoi Diagram in the Laguerre geometry* [Imai et al., 1985] or the *power diagram* [Aurenhammer, 1987].

The power diagram, PV, associates a power region, PV_i, with each circle $B_i(\mathbf{q}_i, R_{\mathbf{q}_i})$ in $\mathbb{R}^2$ defined by:

$$PV_i = \{\mathbf{q} \in \mathbb{R}^2 | d(\mathbf{q}, \mathbf{q}_i) \leq d(\mathbf{q}, \mathbf{q}_j), \forall j \neq i\}, \tag{5.13}$$

where the *power distance* $d(\mathbf{q}, \mathbf{q}_i) = \|\mathbf{q} - \mathbf{q}_i\|^2 - R^2_{\mathbf{q}_i}$.

The power diagram can be viewed as a generalized Voronoi diagram which is closely related to the original Voronoi diagram. Some of its properties are:

Property 5.1 *The bisector between neighbor power cells PV_i and PV_j is a hyperplane perpendicular to the segment that connects the centers of the circles, B_i and B_j. If the two circles intersect, the bisector passes through the points of intersection. The bisector is defined by the equation:*

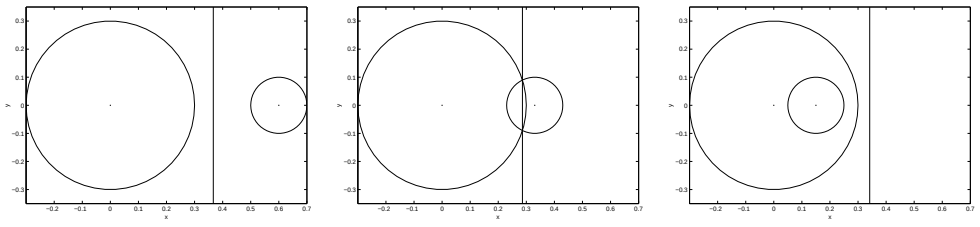
$$(\mathbf{q}_i - \mathbf{q}_j)^T \mathbf{q} = \frac{1}{2}(\|\mathbf{q}_i\|^2 - \|\mathbf{q}_j\|^2 - R^2_{\mathbf{q}_i} + R^2_{\mathbf{q}_j}). \tag{5.14}$$

Property 5.2 *Each power cell PV_i is convex or empty. A necessary condition for PV_i to be empty is that the center $\mathbf{q}_i$ is contained in the union of the other circles.*

Property 5.3 *If a circle, B_i, is not intersected by any other circle, then B_i is entirely contained in PV_i.*

Property 5.4 *If all circles B_i are identical, $PV_i = V_i$.*

Figure 5.1 presents three different cases for the power regions for two circles. Figure 5.1(c) presents the situation where an agent is not located



(a) (b) (c)

Figure 5.1: Power cells. Fig. 5.1(a) Two disjoint circles. Fig. 5.1(b) Intersecting circles with both centers inside their corresponding power cells. Fig. 5.1(c) Intersecting circles with one center outside its corresponding power cell.

inside its own power region. Thus it is also possible that a robot could have an empty power region. In Figure 5.1(b), if we have a third robot the same size of the smallest robot inside the power region of the large robot, this third robot contained in the power region of the largest robot would have an empty power region.

The next proposition presents the gradient that will be used by the distributed control laws.

Proposition 5.3 *The gradient of $\mathcal{H}(\mathcal{P}, PV)$ in (5.12) is given by:*

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} = 2(\mathbf{q}_i - \mathbf{q}_i^*) \int_{PV_i} \varphi(\mathbf{q}) d\mathbf{q}\,, \tag{5.15}$$

where $\mathbf{q}_i^*$ is the centroid of PV_i:

$$\mathbf{q}_i^* = \frac{\int_{PV_i} \mathbf{q}\varphi(\mathbf{q}) d\mathbf{q}}{\int_{PV_i} \varphi(\mathbf{q}) d\mathbf{q}}. \tag{5.16}$$

Proof: According to Proposition 5.2 we have

$$\begin{aligned}\frac{\partial \mathcal{H}(\mathcal{P}, PV_i)}{\partial \mathbf{q}_i} &= \int_{PV_i} \frac{\partial[\|\mathbf{q}-\mathbf{q}_i\|^2 - R_{\mathbf{q}_i}^2]}{\partial \mathbf{q}_i}\varphi(\mathbf{q})d\mathbf{q} \\ &= \int_{PV_i} -2(\mathbf{q}-\mathbf{q}_i)\varphi(\mathbf{q})d\mathbf{q} \\ &= 2(\mathbf{q}_i - \mathbf{q}_i^*)\int_{PV_i}\varphi(\mathbf{q})d\mathbf{q}.\end{aligned}$$

■

By observing the last proposition we conclude that we can use the same control law in (5.11). However, one must be careful of the special properties of power diagrams.

As stated in Property 5.2, a robot that is located in the union of other sensors footprint may have an empty power region. This is not surprising. Since we are designing strategies for heterogeneous teams of robots that act only on local information, robots that have better sensors than others have priority during the deployment. Thus, robots with the worst sensors may get trapped in configurations in which they do not contribute to the overall mission. Of course, it is possible to let the "trapped" robot perturb the system by executing a deterministic controller to a region outside the circle in which it is trapped. In practice, we didn't find empty power regions in our simulations.

5.3 Robots with Finite Size

A practical problem of the unconstrained minimization executed by the pure gradient-descent law in (5.11) is that actual robots are not point-robots. In this section, we extend the basic results to robots that can be modelled as circular disks, each one with radius $r_{\mathbf{q}_i}$. Also, as in [Cortez et al., 2004], we

first assume d is the Euclidean distance and $f(d) = d^2$. Let $\mathcal{F}_{V_i}$ be the *free Voronoi region* defined by the set of points:

$$\mathcal{F}_{V_i} = \{\mathbf{q} \in V_i | \|\mathbf{q} - \mathbf{q}_{\partial V_i}\| \geq r_{\mathbf{q}_i}, \forall \mathbf{q}_{\partial V_i}\}\,, \tag{5.17}$$

where $\|\cdot\|$ is the Euclidean norm and $\mathbf{q}_{\partial V_i}$ is a point at the boundary of the Voronoi region, ∂V_i. In fact, the boundaries of the free Voronoi regions, $\partial\mathcal{F}_{V_i}$, are hyperplanes parallel to the hyperplanes that define the boundaries of V_i, located at a distance $r_{\mathbf{q}_i}$ from ∂V_i. It is straight forward to check that such free region is convex.

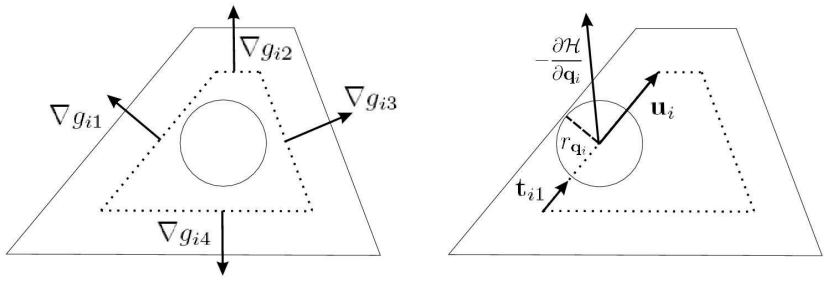
If the robots start from a safe configuration (*i. e.* the robots are not colliding in the start configuration), a sufficient condition to guarantee collision avoidance is that each robot disk lies in the interior of its own Voronoi region. Therefore, we define a *constrained, location optimization* problem as follows. If robot i is constrained to remain inside its own Voronoi region which is bounded by m facets, there are at most m linear constraints on the position of its center. Accordingly, we find the robot positions:

$$\begin{aligned} &\min_{\mathbf{q}_i} \mathcal{H}(\mathcal{P}, V) \\ &\text{s.t.} \\ &g_{i1}(\mathbf{q}_i) \leq 0\,, \ldots\,, g_{im}(\mathbf{q}_i) \leq 0 \end{aligned} \tag{5.18}$$

where $g_{il}(\mathbf{q}_i) = 0$ defines the lth facet of $\partial\mathcal{F}_{V_i}$.

Accordingly we choose the control law given by:

$$\mathbf{u}_i = \beta_i \mathbf{h}_i\,, \tag{5.19}$$



(a) (b)

Figure 5.2: Linear constraints for a disk-robot with radius $r_{\mathbf{q}_i}$. The dotted lines represent the facets of $\partial\mathcal{F}_{V_i}$, which are given by $g_{i1} = 0$, $g_{i2} = 0$, $g_{i3} = 0$, and $g_{i4} = 0$, with gradients as in Fig. 5.2(a). Given the active set $\mathcal{A} = \{g_{i1}\}$ in Fig. 5.2(b) we compute the control input $\mathbf{u}_i$ by projecting the negative gradient of $\mathcal{H}$ onto the unit vector $\mathbf{t}_{i1}$ which is tangent to the active facet.

where $\beta_i \propto \left\| \frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} \right\|$ and the vector $\mathbf{h}_i$ is determined by:

$$\begin{aligned} &\min_{\mathbf{h}_i} \left(\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} \right)^T \mathbf{h}_i \\ &\text{s.t.} \\ &\|\mathbf{h}_i\| = 1 \,, \nabla g_{ip}^T \mathbf{h}_i \leq 0 \,, \ldots \,, \nabla g_{it}^T \mathbf{h}_i \leq 0 \end{aligned} \tag{5.20}$$

If $\mathcal{A} = \{g_{ip}, \ldots, g_{it}\}$ is the set of active constraints $g_{ip} = \ldots = g_{it} = 0$, we can compute the controls in (5.19) as follows:

1. If $\mathcal{A}$ is empty then
$$\mathbf{u}_i = -k \frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} \,, \tag{5.21}$$

2. Otherwise
$$\mathbf{u}_i = k\pi \left(-\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}, \partial \mathcal{F}_{V_i} \right) \,, \tag{5.22}$$
where $\pi \left(-\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}, \partial \mathcal{F}_{V_i} \right)$ gives the projection of vector $-\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}$ along the vector $\mathbf{t}_{il}$ which is the unit vector tangent to the lth facet (see Fig. 5.2).

This tangent vector provides a feasible direction and is such that

$$-\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}^T \mathbf{t}_{il} > 0. \tag{5.23}$$

The presented approach can be considered as a type of *Gradient Projection Method* [Rosen, 1960].

Since the region $\mathcal{F}_{V_i}$ is convex and $\partial\mathcal{F}_{V_i}$ is given by plane faces, the projection π is guaranteed to return a feasible vector if one exists. Equilibrium is obtained when the first-order Karush-Kuhn-Tucker (KKT) conditions [Karush, 1939, Kuhn and Tucker, 1951] are satisfied.

Clearly the control law designed to solve the problem in (5.18) will not guarantee the convergence of robots to the centroids, $\mathbf{q}_i^*$. However, they converge to the *constrained centroids*, $\mathbf{q}_i^c$, which are determined by the projection of $\mathbf{q}_i^*$ on the boundary $\partial\mathcal{F}_{V_i}$. This solution is locally optimal in the sense that this is the best solution which also guarantees that the entire robot disk (and not just the center point) is confined to its own dominance region.

Remark 5.1 We can use the algorithm presented in this section with heterogeneous mobile sensors by using (5.15) and the power diagram, PV, to compute $\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}$ but (5.19, 5.20) computed using the facets from the Voronoi diagram, V.

5.4 Nonconvex Environments

In [Ganguli et al., 2006], nonconvex environments are addressed in the problem of deploying a team of robots to achieve full visibility of the environment. Differently, in the present work we consider the following problem

in a nonconvex environment: *minimize the coverage functional (5.1), such that d is the geodesic distance and $f(d) = d^2$*:

$$\mathcal{H}(\mathcal{P}, T) = \sum_{i=1}^{n} \mathcal{H}(\mathbf{q}_i, T_i) = \sum_{i=1}^{n} \int_{T_i} d(\mathbf{q}, \mathbf{q}_i)^2 \varphi(\mathbf{q}) d\mathbf{q}. \tag{5.24}$$

Let Ω be a compact region in $\mathbb{R}^2$ with boundary, $\partial\Omega$, determined by a simple polygon with m sides and set of vertices $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$. Also, assume that $\mathcal{P} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\} \subset \Omega$. By the geodesic distance, $d(\mathbf{o}, \mathbf{w})$, between two points $\mathbf{o}$ and $\mathbf{w}$, we mean the length of the shortest path, $s(\mathbf{o}, \mathbf{w})$, between $\mathbf{o}$ and $\mathbf{w}$, entirely contained in Ω. In fact, it is well known that such a path is formed by the sequence of segments $\{\overline{\mathbf{o}\mathbf{v}_{r1}}, \overline{\mathbf{v}_{r1}\mathbf{v}_{r2}}, \ldots, \overline{\mathbf{v}_{rl-1}\mathbf{v}_{rl}}, \overline{\mathbf{v}_{rl}\mathbf{w}}\}$, where $\mathbf{v}_{ri}$'s are reflex vertices of $\partial\Omega$. By reflex vertices, we mean the vertices with internal angle greater than 180 degrees. According to Proposition 5.1 we require a Voronoi tessellation computed using the geodesic metric. Such a geodesic Voronoi tessellation can be computed by means of the algorithm proposed in [Aronov, 1989]. Some properties of the geodesic distance in simple polygons are [Aronov, 1989]:

Property 5.5 *As Ω is closed and bounded by a simple polygon, $s(\mathbf{o}, \mathbf{w})$ exists and is unique for every pair of points in Ω. Moreover, $s(\mathbf{o}, \mathbf{w})$ is piecewise linear with "breakpoints" at reflex vertices of $\partial\Omega$.*

Property 5.6 *The function $d(\mathbf{o}, \mathbf{w})$ is continuous in both $\mathbf{o}$ and also $\mathbf{w}$. Furthermore, this function is continuously differentiable except at a set of measure zero Λ_0.*

The following proposition allows for devising distributed control laws.

Proposition 5.4 *If $\mathbf{w} \in \Omega \backslash \Lambda_0$ is a point where the gradient $\nabla_{\mathbf{w}} d(\mathbf{w}, \mathbf{q})$*

exists, then this gradient is given by:

$$\frac{\partial d}{\partial \mathbf{w}} = -\mathbf{z}_{\mathbf{w},\mathbf{q}}\,, \tag{5.25}$$

where $\mathbf{z}_{\mathbf{w},\mathbf{q}}$ *is a unit vector directed along the first segment of the shortest path* $s(\mathbf{w},\mathbf{q})$.

Proof: Let the shortest path, $s(\mathbf{w},\mathbf{q})$, between the points $\mathbf{w}$ and $\mathbf{q}$, be given by the sequence of segments $\{\overline{\mathbf{w}\mathbf{v}_{r1}}, \overline{\mathbf{v}_{r1}\mathbf{v}_{r2}}, \ldots, \overline{\mathbf{v}_{rl-1}\mathbf{v}_{rl}}, \overline{\mathbf{v}_{rl}\mathbf{q}}\}$. The path length is then determined by: $d(\mathbf{w},\mathbf{q}) = \|\mathbf{v}_{r1} - \mathbf{w}\| + \|\mathbf{v}_{r2} - \mathbf{v}_{r1}\| + \ldots + \|\mathbf{v}_{rl} - \mathbf{v}_{rl-1}\| + \|\mathbf{q} - \mathbf{v}_{rl}\|$, where $\mathbf{v}_{ri}$'s are reflex vertices of the polygon $\partial\Omega$. Therefore,

$$\frac{\partial d}{\partial \mathbf{w}} = -\frac{(\mathbf{v}_{r1} - \mathbf{w})}{\|\mathbf{v}_{r1} - \mathbf{w}\|}. \tag{5.26}$$

■

From this proposition and from (5.8), a distributed control law that leads to the minimization of $\mathcal{H}$ in (5.24) can be computed by:

$$\mathbf{u}_i = -k\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} = 2k\int_{V_i} d(\mathbf{q}_i,\mathbf{q})\varphi(\mathbf{q})\mathbf{z}_{\mathbf{q}_i,\mathbf{q}}d\mathbf{q}. \tag{5.27}$$

It is important to mention that at the points where we have discontinuities in the gradient, which are also reflex vertices of the polygon $\partial\Omega$, we must use generalized gradients [Clarke, 1983].

Definition 5.1 *(Clarke's Generalized Gradient): For a locally Lipschitz function* $g : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}$ *define the generalized gradient of* g *at* $(\mathbf{x},t)$ *by*

$$\partial g(\mathbf{x},t) = \overline{co}\{\lim \nabla g(\mathbf{x}_i,t_i) | (\mathbf{x}_i,t_i) \to (\mathbf{x},t)\}\,, \tag{5.28}$$

where $\overline{co}$ *is the convex closure,* $(\mathbf{x}_i,t_i) \notin \Lambda_0$ *which is the set of measure zero*

where the gradient is not defined.

Assume that agent $\mathbf{q}_i$ is located exactly at a reflex vertex $\mathbf{v}_{ri}$. Let $\partial_F\mathcal{H}(\mathbf{q}_i, V_i) \subset \partial\mathcal{H}(\mathbf{q}_i, V_i)$ be the subset of generalized gradients, ξ, at the reflex vertex, $\mathbf{v}_{ri}$, that have feasible directions, *i. e.*, $\exists\delta \in (0, \delta_{max}]$ such that $\mathbf{v}_{ri} - \delta\xi \in \Omega$. We can use the following control law at the reflex vertices:

$$\mathbf{u}_i = -k\xi\,, \tag{5.29}$$

where k is a positive gain, and ξ could be any vector belonging to $\partial_F\mathcal{H}(\mathbf{q}_i, V_i)$.

Remark 5.2 Heterogeneous robots in nonconvex environments may be considered by replacing both the Euclidean distance in (5.12) by the geodesic distance and the radius $R_{\mathbf{q}_i}$ by the length of the part of the geodesic path which lies inside the circle B_i. This is an interesting future direction of research since the properties of the resulting Voronoi tessellation still needs to be investigated. Disk-shaped robots can also be considered in nonconvex environments by projecting the computed control law onto the facets of the ordinary Voronoi diagram, V, when necessary.

5.5 Simulation Results

In this section we present simulations developed in MATLAB [1] to verify the three extensions to the work in [Cortez et al., 2004]. First we show in Figure 5.3 the result of applying the algorithm described in Section 5.2 to deal with heterogeneous, point robots. We show a team of nine agents with three possible footprints. Two agents have a large circular footprint of radius

[1] http://www.mathworks.com/products/matlab/. Accessed in 19 January 2009.

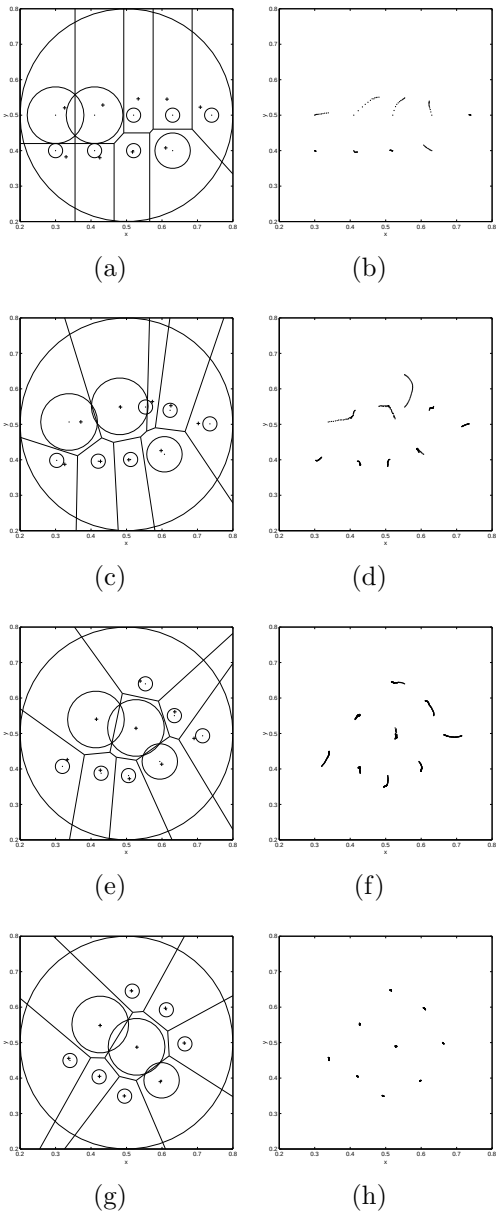


Figure 5.3: Simulation results for heterogeneous agents in a environment with density function defined as in (5.30). In the panels on the left, the large circle with diameter 0.6 units represents the density function support. A team of nine robots, starting from an initial configuration (Fig. 5.3(a)), with intermediate configurations (Figs. 5.3(c) and 5.3(e)), converge to a centroidal formation (Fig. 5.3(g)). Figs. 5.3(b), 5.3(d), 5.3(f), and 5.3(h) correspond to the trajectories followed by the robots starting from the configuration in the figure presented in the left. The crosses represent the centroids of the corresponding power regions. Small circles associated with robots represent the footprints of the sensors carried by the robots.

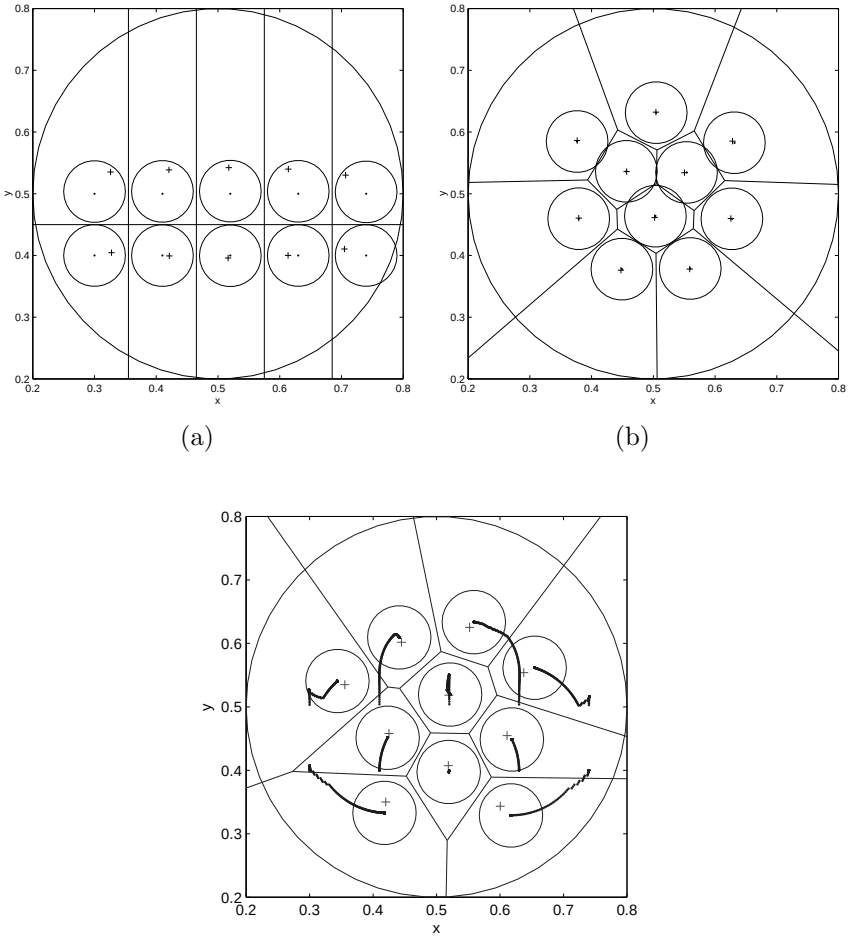
0.08 units, one agent has a medium-size circular footprint of radius 0.05 units, and the remaining robots have the smallest circular footprint of radius 0.02 units. In the left figures we present configurations of the group and in the right we present trajectories that show how the robots evolved from the configuration shown in the panel immediately to the left to the next configuration in the panel below. The trajectories in Figure 5.3(h) end at the optimal configuration. The corresponding power diagrams are also presented. The density function is a smooth cubic spline centered at the point $\mathbf{q}_c = [0.5, 0.5]^T$:

$$\varphi(\mathbf{q}-\mathbf{q}_c, h) = \begin{cases} 1 - \frac{3}{2}\kappa^2 + \frac{3}{4}\kappa^3 & \text{if} \quad 0 \leq \kappa \leq 1\,, \\ \frac{1}{4}(2-\kappa)^3 & \text{if} \quad 1 \leq \kappa \leq 2\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{5.30}$$

where $\kappa = \|\mathbf{q} - \mathbf{q}_c\|/h$. It can be observed that the function support is determined by $2h$. In the simulation $h = 0.15$.

We can verify that the final configuration is obtained when a centroidal tessellation is obtained, as expected. It is interesting to note that the best sensors converged to the area where better coverage is required, *i.e.*, the area where we find the higher values of the density function.

Figure 5.4 shows the result obtained when solving the constrained minimization proposed in Section 5.3 for a team of ten disk-shaped robots. The density function is the same as before. In this example, collisions are observed when the unconstrained control law (5.11) is used (see Figure 5.4(b)). However, the trajectories shown in Figure 5.4(c) which are obtained by using the control law (5.19) are free of collisions. Note that, in this case, not all agents converge to their centroids (see Figure 5.4(c)). This is the price the robots have to pay to guarantee safety (no collisions).



(a) (b)

(c)

Figure 5.4: Simulation results for finite size agents in an environment with density function defined as in (5.30). A team of ten finite size agents start from an initial configuration (Fig. 5.4(a)) and have final configuration that implies collisions as in Fig. 5.4(b) when executing the unconstrained minimization as in [Cortez et al., 2004]. By using the technique proposed in Section 5.3 the agents reach a final configuration (Fig. 5.4(c)) without colliding (see trajectories). The crosses represent the centroids of the corresponding Voronoi regions. The largest circle represents the density function support while the other circles represent the shape of the robots.

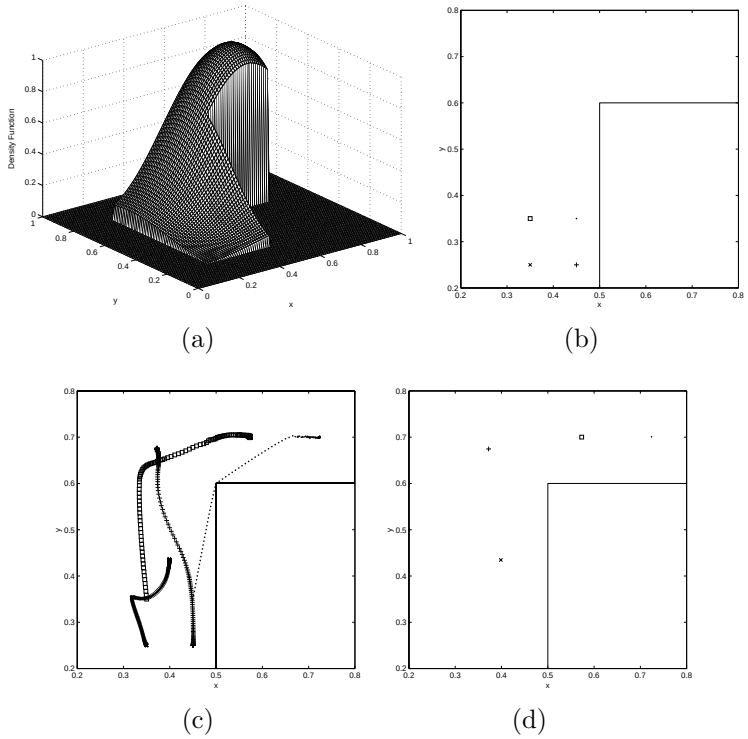


Figure 5.5: Simulation results for a L-shape environment. Given a density function defined in a nonconvex domain (Fig. 5.5(a)), a group of four robots with the initial configuration shown in Fig. 5.5(b) executes the approach proposed in Section 5.4 and moves according to the trajectories in Fig. 5.5(c). The final configuration is presented in Fig. 5.5(d).

The extension to address nonconvex environments is verified in Figure 5.5. A group of four agents with the initial configuration shown in Figure 5.5(b) cover a L-shape environment according to the density function in Figure 5.5(a). The peak of the function is located at the point $[0.7, 0.7]^T$. This density function is a cubic spline inside the nonconvex domain and zero outside. One can conclude from Figure 5.5(c) that the robots follow trajectories that do not leave the nonconvex domain as desired. As expected, the robots moved to the region where better coverage is required (see the final configuration in Figure 5.5(d)).

Chapter 6

Conclusions

6.1 Summary

In this work we proposed novel approaches to control swarms of robots. Our approaches are scalable and there is no need for labelling the robots. Therefore, all the robots run the same software and the success in task execution is not dependent on specific members of the group. Besides, our approaches are robust to dynamic deletion and addition of new agents.

We propose to use the Smoothed Particle Hydrodynamics simulation technique to model the swarm as a fluid. In fact, we use incompressible fluid models which provide density control. This density control provides a loose way of controlling the swarm connectivity. We use the Finite Element Method (FEM) to compute harmonic functions that determine external forces to the fluid. These forces are mainly responsible for driving the swarm to a desired region of the workspace. Since obstacles may have generic geometries, the use of FEM allows for efficiency in the function computation. By means of a weak coupling between FEM and SPH, the derived controllers are decentralized in the sense that only local information is used by a robot of the group: the gradient of the harmonic function at the location of the robot and position and velocity of neighbor robots. For the first time, mathematical

guarantees such as stability and convergence of controllers derived from the SPH equations were established. The proposed approach was successfully instantiated in a pattern generation task and also in a density constrained coverage task. Techniques to accommodate actual robot features such as finite size and nonholonomic constraints (more specifically non-sideslipping constraint) are proposed. Basically, this is accomplished by using feedback linearization and also by adapting the artificial viscosity term. The vector field computed from the designed harmonic function helps to avoid collisions between robots and static obstacles. However, such vector field may be not enough and collisions may occur. An strategy that places virtual particles at the boundaries of the obstacles is proposed to guarantee collision avoidance. The artificial viscosity terms provided by these virtual particles are responsible for guaranteeing such property. Ideal computer simulations, realistic simulations and actual robots experiments were performed showing the efficacy of the method.

In the case of sensing coverage tasks, a very important property is optimality. However, it is difficult to prove that the proposed SPH based controllers drive the robots to an optimal configuration for coverage. Therefore, we investigated novel tools that guarantee optimality. In this work, we use a locational optimization based framework in order to devise optimal distributed control laws. This strategy also keeps one of the main functionalities of the previously proposed approach which was the ability to control the density of robots over a certain region. We incorporate three novel extensions into the previous works [Lloyd, 1982, Cortez et al., 2004] to address: (a) sensors with circular footprints of different radii, (b) disk-shaped robots, and (c) nonconvex polygonal environments. The extensions are based on the use of different distance functions, power distance and geodesic distance, and

the incorporation of constraints to allow collision avoidance.

Both approaches SPH and locational optimization are decentralized since only local information is used. This is important in order to guarantee scalability. In the first approach, each agent requires information from other agents which are located within a pre-specified range. In the latter one each agent requires information from its Voronoi neighbors.

It is important to mention that the SPH approach requires less computation. For each agent, only simple sums that iterates over the neighbor agents are executed. In the locational optimization based controllers, each agent needs to compute the Voronoi diagram and afterwards perform numerical integrations over its Voronoi cell. On the other hand, the optimality property is inherent to the optimization based approach. Nevertheless, it should be clear that we can only guarantee local optimality. In general, the objective function is nonlinear and nonconvex. Therefore, we can conclude that each approach has its advantages and disadvantages and the choice of one of them for a given application will depend on the context.

6.2 Publications

In this section we present the list of papers related to the theme of this thesis, which were published during the doctoral period.

- Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., Campos, M. F. M. (2005). On Computing Complex Navigation Functions. In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2005)*, p. 3463-3468, (Barcelona, Spain).

- Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C.,

Silva, E. J., Caminhas, W. M., Campos, M. F. M. (2005). Robots navigation based on electromagnetic fields. In: *Record of the 15th COMPUMAG Conference on the Computation of Electromagnetic Fields*, v. 4, p. 154-155, (Shenyang, China).

- Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C., Pereira, G. A. S. (2006). Fluids, Particles, and Multiple Robots in Electrostatic Fields. In: *Proceedings of The Twelfth Biennial IEEE Conference on Electromagnetic Field Computation (CEFC 2006)*, p. 309, (Miami, USA). (This paper won the Poster Presentation Award at the Twelfth Biennial IEEE Conference on Electromagnetic Field Computation)

- Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., Campos, M. F. M. (2006). *Robot navigation based on electrostatic field computation.* In: IEEE Transactions on Magnetics, v. 42, n. 4, p. 1459-1462.

- Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C., Pereira, G. A. S. (2006). Uma Abordagem via Simulação de Fluidos em Campos Eletrostáticos para Geração de Padrões por Múltiplos Robôs. In: *Anais do MOMAG 2006 - 12º SBMO - Simpósio Brasileiro de Microondas e Optoeletrônica - 7º CBMAG Congresso Brasileiro de Eletromagnetismo*, p. 1-5 ,(Belo Horizonte, Brazil).

- Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C., Pereira, G. A. S. (April, 2007). *Fluids in Electrostatic Fields: An Analogy for Multi-Robot Control.* IEEE Transactions on Magnetics, v. 43, n. 4, p. 1765-1768.

- Pimenta, L. C. A., Pereira, G. A. S., Mesquita, R. C. (2007). Fully

continuous vector fields for mobile robot navigation on sequences of discrete triangular regions. In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2007)*, p. 1992-1997, (Rome, Italy).

- Pereira, G. A. S., Pimenta, L. C. A., Chaimowicz, L., Fonseca, A. R., Almeida, D. S. C., Correa, L. Q., Mesquita, R. C., Campos, M. F. M. (2008). *Robot navigation in multi-terrain outdoor environments.* In: Experimental Robotics: The 10th International Symposium on Experimental Robotics ed.Berlin / Heidelberg : Springer, v.39, p. 331-342.

- Pimenta, L. C. A., Michael, N., Mesquita, R. C., Pereira, G. A. S., Kumar, V. (2008) Control of Swarms Based on Hydrodynamic Models. In: *Proceedings of the IEEE International Conference on Robotics and Automation 2008*, p. 1948-1953, (Pasadena, CA, EUA).

- Mendes, M. L., Pimenta, L. C. A., Mesquita, R. C., Silva, E. J., Santana, T. C. (2008). Smoothed Particle Electromagnetics with Boundary Absorbing Condition Using Perfectly Matched Layers. In: *Proceedings of the IET 7th International Conference on Computation in Electromagnetics*, p. 164-165, (Brighton, UK).

- Pimenta, L. C. A., Kumar, V., Mesquita, R. C., Pereira, G. A. S. (2008). Sensing and Coverage for a Network of Heterogeneous Robots. In: *Proceedings of the 47th IEEE Conference on Decision and Control*, p. 3947-3952, (Cancún, México).

- Pimenta, L. C. A., Schwager, M., Lindsey, Q., Kumar, V., Rus, D., Mesquita, R. C., Pereira, G. A. S. (2008). Simultaneous Coverage and Tracking (SCAT) of Moving Targets with Robot Networks. In:

Proceedings of the Eighth International Workshop on the Algorithmic Foundations of Robotics, p. 1-16, (Guanajuato, México).

- Gonçalves, V. M., Pimenta, L. C. A., Maia, C. A., Pereira, G. A. S. (2009). Artificial vector fields for robot convergence and circulation of time-varying curves in n-dimensional spaces. (accepted for publication) *American Control Conference*, p. 1-6. (This paper was selected as one of four final nominees for the Best Student Paper Award at the 2009 American Control Conference)

- Pereira, G. A. S., Pimenta, L. C. A., Chaimowicz, L., Fonseca, A. R., Almeida, D. S. C., Correa, L. Q., Mesquita, R. C., Campos, M. F. M. (2009). *Robot navigation in multi-terrain outdoor environments* (to appear). The International Journal of Robotics Research, p. 1-17.

6.3 Future Work

So far, we have instantiated our fluid based approach in a pattern generation task and in an environment coverage task. A possible future work is to consider other cooperative tasks. We are mainly interested in the problem of object manipulation, where the robots must transport an object from a determined part of the workspace to another. Although some solutions have already been devised [Song and Kumar, 2002, Pereira et al., 2003, Fink et al., 2008], such task is still a challenge in the robotics community. We intend to address this task by considering a phase transition in the fluid. If the fluid becomes a solid after the robots have surrounded the object, then it is possible to transport it by properly driving such solid. We also intend to study the effects of defining different values of density over the environment.

Applications that require different concentrations of agents in different portions of the environment may directly benefit of such study. It can be also interesting to investigate novel vector field computation techniques which may be useful in other tasks. We believe that the vector fields obtained in [Pimenta et al., 2007b, Pereira et al., 2008, Pereira et al., 2009] for single robot navigation may be extended for multi-robot coordination. These vector fields are free of local minima and vanish only at the goal.

We presented a solution to the problem of static pattern generation in two-dimensional environments. By static, we mean patterns that do not change over time. An interesting future work is the development of novel techniques to generalize the obtained results to time-varying curves in n-dimensional spaces. This would allow, for example, teams of Unmanned Aerial Vehicles (UAV) to converge to time-varying curves. In the SPH point of view, the consideration of n-dimensional environments is straight forward since the SPH approach is not restricted to the number of dimensions. On the other hand, the computation of harmonic functions is harder in this case. This is because the Finite Element Method (FEM) depends on meshes with good quality which are difficult to compute in higher dimensions. One option is to investigate some mesh-free methods that can be used in this case. Another option is to consider novel artificial vector fields that generalize to n-dimensional spaces and also to time-varying curves. We already have preliminary results in this last direction of research. We had a paper accepted for publication in the American Control Conference (ACC) 2009 [Gonçalves et al., 2009] that considers the convergence of a single robot to time-varying curves in n-dimensional spaces.

In this work we established a weak coupling between SPH and FEM, where FEM is used to compute static external forces to SPH equations.

Due to the number of advantages of using SPH in fluid dynamics problems, it would be also interesting to establish a strong coupling between SPH and FEM to solve real physical problems. A possible future work could be an iterative method for solving problems of fluids and charged particles in electrostatic fields. Possible applications are: electrophoresis, aerosol particle deposition, electrostatic precipitation, etc. We already have preliminary results in the application of SPH ideas to address electromagnetic problems [Mendes et al., 2008].

We addressed the problem of deriving optimal distributed control laws for networks of robots by using a locational optimization based framework. We considered heterogeneous mobile sensor, agents with finite sizes, and nonconvex polygonal domains. As a future work, we intend to derive novel extensions of this framework. The first extension may be the consideration of nonconvex environments with holes. The second extension is the consideration of time-varying density functions. In fact, we already have preliminary results on time-varying density functions which were published in the Workshop on the Algorithmic Foundations of Robotics (WAFR) 2008 [Pimenta et al., 2008c]. In this case, we proposed a technique that allows groups of robots to track intruders while also keeping coverage of the environment. We model the intruders by means of radial basis functions which are then used to compute the required density functions. Since the intruders are allowed to move, the density functions change over time. We are planning to further investigate the issues of the proposed approach such as the presence of singularities in the control law. Besides, we aim to consider different dynamic cases such as the case of moving environment boundaries.

Bibliography

[Aronov, 1989] Aronov, B. (1989). On the geodesic Voronoi diagram of point sites in a simple polygon. *Algorithmica*, 4(1):109–140.

[Aurenhammer, 1987] Aurenhammer, F. (1987). Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96.

[Barnes et al., 2006] Barnes, L., Alvis, W., Fields, M., Valavanis, K., and Moreno, W. (2006). Heterogeneous swarm formation control using bivariate normal functions to generate potential fields. In *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS 2006)*, pages 85–94.

[Belta and Kumar, 2004] Belta, C. and Kumar, V. (2004). Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875.

[Beni and Wang, 1989] Beni, G. and Wang, J. (1989). Swarm intelligence. In *Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan*, pages 425–428.

[Berman et al., 2007] Berman, S., Ádám Halász, Kumar, V., and Pratt, S. (2007). Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 2318–2323.

[Bonabeau et al., 1999] Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, New York, NY.

[Chaimowicz and Kumar, 2004] Chaimowicz, L. and Kumar, V. (2004). Aerial shepherds: Coordination among uavs and swarms of robots. In *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS 2004)*, pages 231–240, Toulouse, France.

[Chaimowicz et al., 2005] Chaimowicz, L., Michael, N., and Kumar, V. (2005). Controlling swarms of robots using interpolated implicit functions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2498–2503, Barcelona, Spain.

[Charifa and Masoud, 2005] Charifa, S. M. and Masoud, A. A. (2005). Solid mechanics-inspired sensor-based motion planner. In *Proc. of the IEEE Conference on Control Applications*, pages 221–226.

[Clarke, 1983] Clarke, F. (1983). *Optimization and Nonsmooth Analysis*. New York: Wiley.

[Connolly, 1992] Connolly, C. I. (1992). Applications of harmonic functions to robotics. In *Proceedings of the IEEE International Symposium on Intelligent Control*.

[Connolly et al., 1990] Connolly, C. I., Burns, J. B., and Weiss, R. (1990). Path planning using Laplace's equation. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 2102–2106.

[Correll, 2007] Correll, N. (2007). *Coordination schemes for distributed boundary coverage with a swarm of miniature robots: synthesis, analysis and experimental validation*. PhD thesis, Ecole Polytechnique Federale de Lausanne.

[Cortez et al., 2005] Cortez, J., Martinez, S., and Bullo, F. (2005). Spatially-distributed coverage optimization and control with limited-range interactions. *ESIAM: Control, Optimisation and Calculus of Variations*, 11:691–719.

[Cortez et al., 2004] Cortez, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255.

[de Berg et al., 2000] de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O. (2000). *Computational Geometry Algorithms and Applications*. Springer-Verlag, second edition.

[Deneubourg et al., 1991] Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1991). The dynamics of collective sorting: robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior*, pages 356–365, Cambridge, MA, USA. MIT Press.

[Dorigo et al., 2004] Dorigo, M., Trianni, V., Sahin, E., Gro, R., Labella, T., Baldassarre, G., Nolfi, S., Deneubourg, J., Mondada, F., Floreano, D., and Gambardella, L. (2004). Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2–3):223–245.

[Drezner, 1995] Drezner, Z. (1995). *Facility Location: A Survey of Applications and Methods*. Springer Series in Operations Research. Springer-Verlag, New York.

[Du et al., 1999] Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676.

[Dudek and Jenkin, 2000] Dudek, G. and Jenkin, M. (2000). *Computational Principles of Mobile Robotics*. Cambridge University Press, 1 edition.

[Fink et al., 2008] Fink, J., Hsieh, M. A., and Kumar, V. (2008). Multi-robot manipulation via caging in environments with obstacles. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 1471–1476.

[Flanders, 1973] Flanders, H. (1973). Differentiation under the integral sign. *American Mathematical Monthly*, 80(6):615–627.

[Gage, 1992] Gage, D. W. (1992). Command control for many-robot systems. In *Nineteenth Annual AUVS Technical Symposium*, pages 22–24.

[Ganguli et al., 2006] Ganguli, A., Cortez, J., and Bullo, F. (2006). Distributed deployment of asynchronous guards in art galleries. In *Proceedings of the American Control Conference*, pages 1416–1421, Minneapolis, USA.

[Gaudiano et al., 2005] Gaudiano, P., Bonabeau, E., and Shargel, B. (2005). Evolving behaviors for a swarm of unmanned air vehicles. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2005)*, pages 317–324.

[Gerkey et al., 2003] Gerkey, B., Vaughan, R. T., and Howard, A. (2003). The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics*, pages 317–323.

[Gingold and Monaghan, 1977] Gingold, R. A. and Monaghan, J. J. (1977). Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389.

[Gonçalves et al., 2009] Gonçalves, V. M., Pimenta, L. C. A., Maia, C. A., and Pereira, G. A. S. (2009). Artificial vector fields for robot convergence and circulation of time-varying curves in n-dimensional spaces (accepted for publication). In *Proceedings of the American Control Conference*, pages 1–6.

[Howard et al., 2002] Howard, A., Matarić, M. J., and Sukhame, G. S. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, pages 299–308.

[Hsieh, 2007] Hsieh, M. A. (2007). *Motion Control Strategies for Networked Robot Teams in Environment with Obstacles.* PhD thesis, University of Pennsylvania.

[Hsieh and Kumar, 2006] Hsieh, M. A. and Kumar, V. (2006). Pattern generation with multiple robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2442–2447, Orlando, USA.

[Hsieh et al., 2007] Hsieh, M. A., Loizou, S., and Kumar, V. (2007). Stabilization of multiple robots on stable orbits via local sensing. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 2312–2317.

[Hughes, 2000] Hughes, T. J. R. (2000). *The Finite Element Method - Linear Static and Dynamic Finite Element Analysis.* Dover Publications,Inc.

[Ida and Bastos, 1992] Ida, N. and Bastos, J. P. A. (1992). *Electromagnetics and Calculation of Fields.* Springer-Verlab.

[Imai et al., 1985] Imai, H., Iri, M., and Murota, K. (1985). Voronoi diagram in the Laguerre geometry and its applications. *SIAM Journal on Computing*, 14(1):93–105.

[Jun et al., 1999] Jun, J.-H., Lee, D.-W., and Sim, K.-B. (1999). Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 6, pages 614–619.

[Karush, 1939] Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. Master's thesis, Department of Mathematics. University of Chicago.

[Kerr and Spears, 2005] Kerr, W. and Spears, D. F. (2005). Robotic simulation of gases for a surveillance task. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2905–2910.

[Kerr et al., 2004] Kerr, W., Spears, D. F., Spears, W. M., and Thayer, D. R. (2004). Two formal fluids models for multiagent sweeping and obstacle avoidance. Lecture Notes in Artificial Intelligence, vol. 3228, Springer-Verlag.

[Khatib, 1986] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98.

[Khepera, 2007] Khepera (2007). Kteam corporation. `http://www.k-team.com`. Visited in February, 2007.

[Koditschek, 1987] Koditschek, D. E. (1987). Exact robot navigation by means of potential functions: Some topological considerations. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 1–6.

[Koenig and Liu, 2001] Koenig, S. and Liu, Y. (2001). Terrain coverage with ant robots: a simulation study. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 600–607, New York, NY, USA. ACM Press.

[Konolige, 2000] Konolige, K. (2000). A gradient method for realtime robot control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 639–646.

[Krishnanand et al., 2006] Krishnanand, K. N., Amruth, P., Guruprasad, M. H., Bidargaddi, S. V., and Ghose, D. (2006). Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 958–963, Orlando, USA.

[Kuhn and Tucker, 1951] Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492.

[Kumar and Sahin, 2003] Kumar, V. and Sahin, F. (2003). Cognitive maps in swarm robots for the mine detection application. In *Proceedings of the*

IEEE International Conference on Systems, Man and Cybernetics, volume 4, pages 3364–3369.

[Kwok and Martínez, 2008] Kwok, A. and Martínez, S. (2008). Deployment algorithms for a power constrained mobile sensor network. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 140–145, Pasadena, USA.

[Latombe, 1991] Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA.

[Liu and Liu, 2003] Liu, G. R. and Liu, M. B. (2003). *Smoothed Particle Hydrodynamics - a meshfree particle method*. World Scientific Publishing Co. Pte. Ltd.

[Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28(2):129–137.

[Luca and Oriolo, 1995] Luca, A. D. and Oriolo, G. (1995). *Modelling and Control of Nonholonomic Mechanical Systems*, chapter 7, pages 277–342. Number 360 in CISM Courses and Lectures. Springer-Verlag.

[Lucy, 1977] Lucy, L. B. (1977). Numerical approach to testing the fission hypothesis. *Astronomical Journal*, 82:1013–1024.

[Meeker, 2004] Meeker, D. (2004). *Finite Element Method Magnetics - User's Manual*. version 4.0.

[Mendes et al., 2008] Mendes, M. L., Pimenta, L. C. A., Mesquita, R. C., Silva, E. J., and Santana, T. C. (2008). Smoothed particle electromagnetics with boundary absorbing condition using perfectly matched layers. In *Proceedings of the IET 7th International Conference on Computation in Electromagnetics*, pages 164–165.

[Michael et al., 2006] Michael, N., Belta, C., and Kumar, V. (2006). Controlling three dimensional swarms of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 964–969, Orlando, USA.

[Michael et al., 2007] Michael, N., Fink, J., and Kumar, V. (2007). Controlling a team of ground robots via an aerial robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 965–970.

[Michael et al., 2008] Michael, N., Fink, J., and Kumar, V. (2008). Experimental testbed for large multi-robot teams: Verification and validation. *IEEE Robotics and Automation Magazine*, 15(1):53–61.

[Michael and Kumar, 2008] Michael, N. and Kumar, V. (2008). Controlling shapes of ensembles of robots of finite size with nonholonomic constraints. In *Proceedings of the Robotics Science and Systems Conference*, pages 1–8.

[Milnor, 1963] Milnor, J. (1963). *Morse Theory*. Princeton University Press.

[Monaghan, 1992] Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574.

[Monaghan, 1994] Monaghan, J. J. (1994). Simulating free surface flow with SPH. *Journal of Computational Physics*, 110:399–406.

[Murray et al., 1994] Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

[Okabe et al., 2000] Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics. New york: Wiley, 2nd edition.

[Pac et al., 2006] Pac, M. R., Erkmen, A. M., and Erkmen, I. (2006). Scalable self-deployment of mobile sensor networks: A fluid dynamics approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1446–1451.

[Pac et al., 2007] Pac, M. R., Erkmen, A. M., and Erkmen, I. (2007). Control of robotic swarm behaviors based on smoothed particle hydrodynamics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4194–4200.

[Pereira, 2003] Pereira, G. A. S. (2003). *Motion Planning and Control of Cooperating Mobile Robots: A Graph Connectivity Approach*. PhD thesis, Universidade Federal de Minas Gerais.

[Pereira et al., 2003] Pereira, G. A. S., Kumar, V., and Campos, M. F. M. (2003). Decentralized algorithms for multi-robot manipulation via caging. *International Journal of Robotics Research*, 23(7):783–795.

[Pereira et al., 2008] Pereira, G. A. S., Pimenta, L. C. A., Chaimowicz, L., Fonseca, A. R., Almeida, D. S. C., Correa, L. Q., Mesquita, R. C., and

Campos, M. F. M. (2008). Robot navigation in multi-terrain outdoor environments. In *Experimental Robotics: The 10th International Symposium on Experimental Robotics*, volume 39, pages 331–342. Springer.

[Pereira et al., 2009] Pereira, G. A. S., Pimenta, L. C. A., Chaimowicz, L., Fonseca, A. R., Almeida, D. S. C., Correa, L. Q., Mesquita, R. C., and Campos, M. F. M. (2009). Robot navigation in multi-terrain outdoor environments (to appear). *International Journal of Robotics Research*, pages 1–17.

[Perkinson and Shafai, 2005] Perkinson, J. R. and Shafai, B. (2005). A decentralized control algorithm for scalable robotic swarms based on mesh-free particle hydrodynamics. In *Proceedings of the IASTED International Conference on Robotics and Applications*, pages 1–6.

[Pimenta, 2005] Pimenta, L. C. A. (2005). Navegação de robôs móveis baseada na equação de Laplace: uma nova abordagem utilizando elementos finitos. Master's thesis, Universidade Federal de Minas Gerais.

[Pimenta et al., 2005a] Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., and Campos, M. F. M. (2005a). On computing complex navigation functions. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 3463–3468.

[Pimenta et al., 2005b] Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., and Campos, M. F. M. (2005b). Robots navigation based on electromagnetic fields. In *Record of the 15th COMPUMAG Conference on the Computation of Electromagnetic Fields*, volume 4, pages 154–155.

[Pimenta et al., 2006a] Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., and Campos, M. F. M. (2006a). Robot navigation based on electrostatic field computation. *IEEE Transactions on Magnetics*, 42(4):1459–1462.

[Pimenta et al., 2008a] Pimenta, L. C. A., Kumar, V., Mesquita, R. C., and Pereira, G. A. S. (2008a). Sensing and coverage for a network of heterogeneous robots. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3947–3952.

[Pimenta et al., 2006b] Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C., and Pereira, G. A. S. (2006b). Fluids, particles, and multiple robots in

electrostatic fields. In *Proceedings of The Twelfth Biennial IEEE Conference on Electromagnetic Field Computation*, page 309.

[Pimenta et al., 2006c] Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C., and Pereira, G. A. S. (2006c). Uma abordagem via simulação de fluidos em campos eletrostáticos para geração de padrões por múltiplos robôs. In *Anais do MOMAG 2006 12o SBMO - Simpósio Brasileiro de Microondas e Optoeletrônica - 7o CBMAG Congresso Brasileiro de Eletromagnetismo*, pages 1–5.

[Pimenta et al., 2007a] Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C., and Pereira, G. A. S. (2007a). Fluids in electrostatic fields: An analogy for multi-robot control. *IEEE Transactions on Magnetics*, 43(4):1765 – 1768.

[Pimenta et al., 2008b] Pimenta, L. C. A., Michael, N., Mesquita, R. C., Pereira, G. A. S., and Kumar, V. (2008b). Control of swarms based on hydrodynamic models. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 1948–1953.

[Pimenta et al., 2007b] Pimenta, L. C. A., Pereira, G. A. S., and Mesquita, R. C. (2007b). Fully continuous vector fields for mobile robot navigation on sequences of discrete triangular regions. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 1992–1997.

[Pimenta et al., 2008c] Pimenta, L. C. A., Schwager, M., Lindsey, Q., Kumar, V., Rus, D., Mesquita, R. C., and Pereira, G. A. S. (2008c). Simultaneous coverage and tracking (scat) of moving targets with robot networks. In *Proceedings of the International Workshop on Algorithmic Foundations of Robotics*, pages 1–16.

[Pioneer, 2009] Pioneer (2009). Activmedia robotics. `http://www.activrobots.com`. Visited in January, 2009.

[Poduri and Sukhatme, 2004] Poduri, S. and Sukhatme, G. S. (2004). Constrained coverage for mobile sensor networks. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 165–172.

[Reynolds, 1993] Reynolds, C. W. (1993). An evolved, vision-based behavioral model of coordinated group motion. In Meyer, J. A., Roitblat, H. L., and Wilson, S. W., editors, *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92)*, pages 384–392, Cambridge, MA. MIT Press.

[Rimon and Koditschek, 1988] Rimon, E. and Koditschek, D. E. (1988). Exact robot navigation using cost functions: The case of distinct spherical boundaries in E^n. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 1791–1796.

[Rimon and Koditschek, 1992] Rimon, E. and Koditschek, D. E. (1992). Exact robot navigation using artificial potencial functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518.

[Rosen, 1960] Rosen, J. B. (1960). The gradient projection method for nonlinear programming. part 1: Linear constraints. *J. Soc. Indust. Appl. Math.*, 8(1):181–217.

[Roy, 1995] Roy, T. M. (1995). Physically based fluid modeling using smoothed particle hydrodynamics. Master's thesis, Graduate College of the University of Illinois.

[Salapaka et al., 2003] Salapaka, S., Khalak, A., and Dahleh, M. A. (2003). Constraints on locational optimization problems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1744–1746, Maui, Hawaii, USA.

[Sato, 1987] Sato, K. (1987). Collision avoidance in multi-dimensional space using laplace potential. In *Proc. of the 15th Conf. Robotics Soc. Jpn.*, pages 155–156.

[Schwager et al., 2008a] Schwager, M., Bullo, F., Skelly, D., and Rus, D. (2008a). A ladybug exploration strategy for distributed adaptive coverage control. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 2346–2353.

[Schwager et al., 2006] Schwager, M., McLurkin, J., and Rus, D. (2006). Distributed coverage control with sensory feedback for networked robots. In *Proceedings of Robotics: Science and Systems*, pages 49–56, Philadelphia, USA.

[Schwager et al., 2008b] Schwager, M., Slotine, J.-J. E., and Rus, D. (2008b). Consensus learning for distributed coverage control. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 1042–1048, Pasadena, USA.

[Shewchuk, 1994] Shewchuk, J. R. (1994). An introduction to the conjugate gradient method without the agonizing pain. Technical report, School of Computer Science, Carnegie Mellon University.

[Shewchuk, 1996] Shewchuk, J. R. (1996). Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Lin, M. C. and Manocha, D., editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag. From the First ACM Workshop on Applied Computational Geometry.

[Shewchuk, 1998] Shewchuk, J. R. (1998). Tetrahedral mesh generation by delaunay refinement. In *Proc. Symp. on Comput. Geometry*, pages 86–95.

[Shimizu et al., 2003a] Shimizu, M., Ishiguro, A., Kawakatsu, T., Masubuchi, Y., and Doi, M. (2003a). Adaptative swarming by exploiting hydrodynamic interaction based on stokesian dynamics method. In *Proceedings of The Society of Instrument and Control Engineers Annual Conference*, volume 2, pages 1546 – 1551.

[Shimizu et al., 2003b] Shimizu, M., Ishiguro, A., Kawakatsu, T., Masubuchi, Y., and Doi, M. (2003b). Coherent swarming from local interaction by exploiting molecular dynamics and stokesian dynamics methods. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1614 – 1619.

[Shreiner et al., 2005] Shreiner, D., Woo, M., Neider, J., and Davis, T. (2005). *OpenGL(R) Programming Guide : The Official Guide to Learning OpenGL(R), Version 2 (5th Edition)*. Addison-Wesley Professional.

[Slotine and Li, 1991] Slotine, J.-J. and Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.

[Song and Kumar, 2002] Song, P. and Kumar, V. (2002). A potential field based approach to multi-robot manipulation. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 1217–1222.

[Soto and Lin, 2005] Soto, J. and Lin, K. C. (2005). Using genetic algorithms to evolve the control rules of a swarm of uavs. In *Proceedings of the International Symposium on Collaborative Technologies and Systems*, pages 359–365.

[Spector et al., 2003] Spector, L., Klein, J., Perry, C., and Feinstein, M. D. (2003). Emergence of collective behavior in evolving populations of flying agents. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2723 of Lecture Notes in Computer Science, pages 61–73, Berlin, Germany. Springer-Verlag.

[Stroustrup, 2000] Stroustrup, B. (2000). *The C++ Programming Language (Special 3rd Edition)*. Addison-Wesley Professional.

[Sun et al., 2001] Sun, S.-J., Lee, D.-W., and Sim, K.-B. (2001). Artificial immune-based swarm behaviors of distributed autonomous robotic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3993–3998.

[Valavanis et al., 2000] Valavanis, K. P., Hebert, T., Kolluru, R., and Tsourveloudis, N. (2000). Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field. *IEEE Transaction on Systems, Man, and Cybernetics—Part A*, 30(2):187–196.

[Wang and Chirikjian, 2000] Wang, Y. and Chirikjian, G. S. (2000). A new potential field method for robot path planning. In *Proceedings of the IEEE International Conference on Robotics Automation*, pages 977–982.

[Ward et al., 2001] Ward, C. R., Gobet, F., and Kendall, G. (2001). Evolving collective behavior in an artificial ecology. *Artificial Life*, 7(2):191–209.

[Weber, 1929] Weber, A. (1929). *Theory of the Location of Industries*. The University of Chicago Press, Chicago, IL. Translated by Carl. J. Friedrich.

[Wilson et al., 2004] Wilson, M., Melhuish, C., Sendova-Franks, A. B., and Scholes, S. (2004). Algorithms for building annular structures with minimalist robots inspired by brood sorting in ant colonies. *Autonomous Robots*, 17(2–3):115–136.