

---

# NAVEGAÇÃO DE ROBÔS UTILIZANDO CURVAS IMPLÍCITAS

**Vinicius Mariano Gonçalves\***

mariano@cpdee.ufmg.br

**Luciano Cunha de Araújo Pimenta†**

lucpim@ufs.br

**Carlos Andrey Maia\***

maia@cpdee.ufmg.br

**Guilherme Augusto Silva Pereira\***

gpereira@cpdee.ufmg.br

\*Laboratório de Sistemas de Computação e Robótica (CORO), Departamento de Engenharia Elétrica, Escola de Engenharia da Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil

†Universidade Federal de São João Del-Rei, Campus Alto Paraopeba, Ouro Branco, MG, Brasil

---

## ABSTRACT

### Robot Navigation Using Implicit Curves

In several applications in robotics, including monitoring, mapping, and surveillance, robots must follow closed curves with several different shapes determined by the application. In this paper, it is proposed a methodology for robot navigation in tasks where the robot path can be specified by the intersection of level sets of given functions. Besides the path, the control law is also defined by a vector field created using these functions. The main contribution of this work is the computation of  $n$ -dimensional vector fields and the proofs that a holonomic robot controlled by this vector field converges and circulates the specified curve. The paper also shows that the resulting field is continuous and can be modified to maintain constant robot speed, an important characteristic for controlling airplane based aerial robots. Finally, it is shown a numerical technique, based on the weighted sum of radial basis functions, for the construction of the functions that determine the robot path. Simulations with a non-holonomic mobile robot subject to localization errors illustrate the potential of the method to drive several types of robots.

**KEYWORDS:** Mobile robots, navigation, implicit functions, limit cycles

## RESUMO

Em diversas aplicações robóticas que incluem monitoramento, mapeamento e vigilância, robôs devem seguir caminhos fechados que podem ter formas diversas, determinadas pela aplicação. Neste artigo é proposta uma metodologia para navegação de robôs em tarefas como estas, onde o caminho do robô pode ser definido como a interseção de curvas de nível de determinadas funções. Além do caminho, a lei de controle do robô também é definida por meio de um campo vetorial criado a partir destas funções. A principal contribuição deste artigo é a construção de campos vetoriais em espaços  $n$ -dimensionais. Prova-se que um robô holonômico controlado por este campo converge e circula a curva especificada. O artigo mostra ainda que o campo resultante é contínuo e pode ser modificado para manter o módulo da velocidade do robô constante, característica importante para controle de alguns robôs aéreos, como é o caso de aviões não-tripulados. Finalmente, é mostrada uma técnica numérica, baseada na soma ponderada de funções de base radial, para criação das funções que determinam o caminho a ser seguido pelo robô. Simulações realizadas com um robô móvel não-holonômico sujeito a erros de localização ilustram o potencial da técnica para a navegação de diversos tipos de robôs.

**PALAVRAS-CHAVE:** Robôs móveis, navegação, funções implícitas, ciclos limite

---

Artigo submetido em 30/03/2009 (Id.: 00984)

Revisado em 22/06/2009, 23/09/2009

Aceito sob recomendação do Editor Associado Prof. Alexandre Bazanella

## 1 INTRODUÇÃO

Um problema clássico em robótica é guiar um robô de uma determinada configuração inicial  $q_i$  para uma configuração final  $q_f$  na presença de obstáculos. Diversas metodologias têm sido propostas para resolver este problema. Estas incluem métodos reativos, puramente baseados em sensores, que podem ou não ter provas formais de que a tarefa será cumprida, e métodos deliberativos baseados em modelos mais complexos do ambiente, que geralmente proveem garantias da completude da missão. Diversas técnicas de planejamento de movimento de robôs são resumidas em (Pereira e Chaimowicz, 2007). Uma referência mais abrangente sobre o assunto é (Choset et al., 2005).

Apesar de diversas metodologias já terem sido propostas, diversos pesquisadores ainda buscam soluções eficientes para problemas envolvendo ambientes complexos e robôs com muitos graus de liberdade e/ou restrições de movimento. No Brasil, vários grupos de pesquisa têm considerado o problema de planejamento e controle de robôs com a utilização de técnicas distintas. Métodos de planejamento de caminhos em ambientes desconhecidos foram propostos por exemplo em (de Lima Ottoni e Lages, 2005) e (Faria e Romero, 2002). Outros métodos baseados em sensores e inspirados pela biologia podem ser vistos em (Langer et al., 2006) e (Tavares Neto e dos Santos Coelho, 2005). Um método estocástico para planejamento de caminhos em ambientes complexos foi proposto em (Adorno e Borges, 2006). Todas estas metodologias encontram caminhos no ambiente para os robôs. Para que o caminho seja seguido pelo robô, os artigos (da Costa e Silva Franco et al., 2008) e (Celeste et al., 2008) propõem controladores baseados no modelo dinâmico e nas restrições cinemáticas do robô. Outras abordagens, como por exemplo aquelas propostas em (Brandão et al., 2006) e (Chaves e Lizarralde, 2006), consideram o planejamento do caminho e o controle do robô simultaneamente. A integração entre planejamento e controle na mesma técnica é uma das características dos métodos baseados em campos vetoriais, como é o caso da metodologia proposta no presente artigo.

A primeira técnica de planejamento de movimento baseada em campos vetoriais foi proposta por Khatib (1986). A metodologia de Khatib era baseada na composição de uma função de potencial global por meio da soma de funções de potencial atrativas, relacionadas ao alvo, e repulsivas, relacionadas aos obstáculos. O negativo do gradiente da função gerada era utilizada como uma entrada de velocidade para o robô, que, eventualmente, era guiado para o alvo. Mesmo sendo elegante e eficiente, a metodologia sofria com mínimos locais, que são pontos distantes do alvo onde o gradiente se anula e, portanto, onde a velocidade do robô também é nula. Este problema foi resolvido posteriormente por outros pesquisadores, que criaram e propuseram técnicas para o cál-

culo de funções de potencial sem mínimo local, conhecidas como Funções de Navegação (Connolly et al., 1990; Rimon e Koditschek, 1992), e (Pimenta et al., 2006).

Outro importante problema de robótica, que recentemente tem recebido atenção da comunidade científica, é o problema de se controlar o robô para convergir e circular sobre curvas fechadas em seu espaço de configurações. Ainda que se trate de um problema de planejamento de movimento, soluções para o problema clássico com apenas uma configuração alvo, não podem ser diretamente aplicadas. Diferentes tarefas, tais como inspeção, manipulação e monitoramento de fronteiras podem ser executadas por meio de soluções deste problema. Uma aplicação interessante, por exemplo, é mostrada em (Pisano et al., 2006). Neste trabalho, um time de veículos aéreos é usado para seguir e circular uma nuvem química liberada na atmosfera.

Em (Chaimowicz et al., 2005; Hsieh e Kumar, 2006) e (Pimenta, Mendes, Mesquita e Pereira, 2007), campos vetoriais são utilizados para resolver o problema de geração de padrões, no qual grandes grupos de robôs móveis devem convergir para uma curva especificada no plano. A principal diferença entre os trabalhos é a maneira como o campo é calculado. Chaimowicz et al. (2005) calculam um campo atrativo como o gradiente de uma função obtida pela interpolação de funções de base radial centradas em amostras do padrão desejado. Esta idéia foi utilizada e generalizada no presente artigo, como será mostrado nas próximas seções. Pimenta, Mendes, Mesquita e Pereira (2007) propuseram a utilização de um método numérico para calcular de forma eficiente um campo eletrostático capaz de atrair o grupo de robôs para a curva alvo. A principal característica desta metodologia é a possibilidade de se utilizar o método em ambientes com obstáculos.

Nos trabalhos anteriores, os robôs eram somente atraídos para a curva alvo, onde deveriam atingir uma configuração estática. Para alguns problemas no entanto, é também necessário um campo vetorial rotacional para fazer o robô circular a curva. A composição dos campos atrativo e rotacional cria um ciclo limite atrativo no espaço de configurações do robô. Para clarear esta ideia, a Figura 1 mostra um exemplo de campo vetorial gerado para guiar um robô para uma circunferência. Este campo foi gerado com a metodologia proposta neste artigo, a qual é detalhada nas próximas seções. Além do presente artigo, esta ideia foi usada em alguns outros trabalhos (Quigley et al., 2005; Hsieh et al., 2007; Frew et al., 2007; Zhang e Leonard, 2007; Ceccarelli et al., 2008; Lawrence et al., 2009), e (Pereira et al., 2008).

Vários destes artigos foram motivados pela natureza do robô utilizado na aplicação. No caso de robôs aéreos baseados em aviões, o robô deve manter uma velocidade mínima, nunca

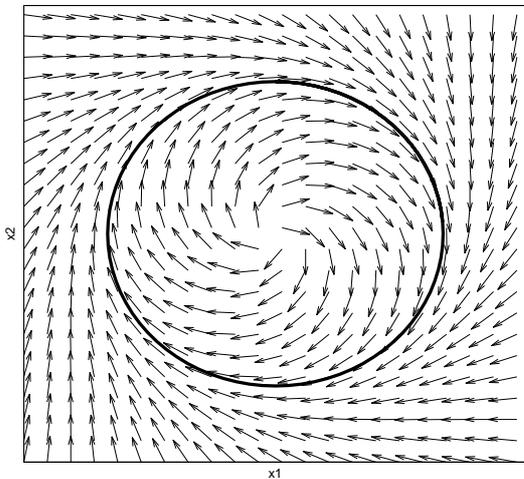


Figura 1: Campo vetorial gerado para guiar o robô para uma circunferência

podendo convergir para um ponto. Assim, a maioria das tarefas a serem realizadas por este tipo de robô devem ser modeladas como um problema de convergência e circulação de curvas. Por exemplo, Frew et al. (2007) controlaram um robô aéreo para rastrear um alvo móvel por meio de um campo vetorial que o guia para percorrer um círculo centrado no alvo.

Em algumas situações, problemas tradicionais de navegação baseada em pontos de passagem (*waypoints*) podem também ser considerados como um problema de convergência para uma curva, como foi mostrado em (Pereira et al., 2008). Neste caso, o principal desafio é construir um campo vetorial que guia o robô para seguir sequências arbitrárias de pontos. A metodologia proposta por Lawrence et al. (2009), por exemplo, gera campos vetoriais contínuos para qualquer curva que pode ser transformada (por meio de transformações afins) em uma curva circular. Esta técnica não é genérica o suficiente e, portanto, os próprios autores propõem uma estratégia de chaveamento entre campos vetoriais para que o robô siga toda a sequência. Este chaveamento gera descontinuidades indesejadas no campo resultante. Pereira et al. (2008) calculam campos contínuos que permitem que o robô aéreo percorra qualquer sequência de *waypoints*. Os autores constroem um corredor poligonal que contém todos os pontos de passagem e utilizam a metodologia proposta em (Pimenta, Pereira e Mesquita, 2007) para gerar um campo dentro do corredor. O principal problema da metodologia é a descontinuidade do campo na borda do corredor e a dificuldade prática de se estender a metodologia para espaços tridimensionais. De fato, metodologias capazes de gerar campos vetoriais em espaços de dimensões maiores que dois não puderam ser encontradas na literatura até a presente data, e constituem a principal contribuição dos autores deste artigo. Exceções são feitas aos artigos dos próprios autores deste tra-

balho em uma conferência nacional (Gonçalves et al., 2008), onde espaços de três dimensões são considerados, e outra internacional (Gonçalves et al., 2009), onde já se considera espaços de dimensão  $n > 2$ .

Mais especificamente, neste artigo, será tratado o problema de se construir um campo vetorial contínuo que guie um robô para uma curva pré-definida. É importante ressaltar que o método proposto não é baseado no gradiente de uma função, uma vez de que a lei de controle aqui proposta não pode ser escrita como o gradiente de um campo escalar quando a curva a ser seguida é fechada, como será mostrado no artigo. O método é válido para qualquer curva simples em espaços  $n$ -dimensionais. São mostradas provas formais que garantem a convergência de um robô holonômico para a curva desejada. O artigo também apresenta técnicas para criação de curvas genéricas em duas e três dimensões. Estas técnicas foram inspiradas pelo artigo apresentado por Chaimowicz et al. (2005) que é focado em um método para guiar “enxames” de robôs na formação de padrões. Ainda que o problema de manter o robô em uma dada trajetória e o problema de controlá-lo para um determinado padrão não sejam o mesmo, o método proposto em (Chaimowicz et al., 2005), que utiliza uma função construída por meio de uma combinação linear de funções de base radial (Turk e Brien, 2002), foi diretamente utilizado neste trabalho para criar curvas fechadas genéricas em duas dimensões, e também, foi generalizado para curvas de três dimensões.

A próxima seção apresenta a metodologia de criação de campos vetoriais a partir de funções implícitas. Esta seção está dividida em duas subseções: a primeira, com caráter tutorial, apresenta a metodologia de forma intuitiva, e a segunda apresenta a formalização matemática, incluindo provas de convergência para campos em duas dimensões. A Seção 3 apresenta uma extensão importante da metodologia básica para duas dimensões, que é a geração de campos normalizados. Com estes campos, a metodologia de controle proposta pode ser usada como uma técnica de guiagem de robôs, na qual o módulo da velocidade dependerá do próprio robô ou da tarefa sendo executada. A extensão da metodologia para dimensões maiores que dois é mostrada na Seção 4. A Seção 5 apresenta uma técnica numérica para geração de curvas em duas e três dimensões a partir de amostras da curva desejada e a Seção 6 discute os passos necessários para que a metodologia teórica proposta possa ser aplicada a robôs reais. Nesta seção são também mostradas simulações mais realísticas que evidenciam o potencial prático da técnica. Finalmente, a Seção 7 apresenta as conclusões do artigo e as propostas de continuidade.

## 2 METODOLOGIA

Esta seção apresenta o principal resultado deste artigo, que consiste no desenvolvimento de uma metodologia para criação de um campo vetorial contínuo capaz de controlar um robô para que sua configuração aproxime e circule uma dada curva.

Considera-se, inicialmente, que o robô a ser controlado é holonômico com dinâmica dada por:

$$\dot{\mathbf{q}}(t) = \mathbf{u}(t), \quad (1)$$

sendo  $\mathbf{q}(t) = [x_1(t) \ x_2(t)]^T$  (todos os vetores neste texto são vetores coluna) a configuração do robô no instante  $t$  e  $\mathbf{u}$  a entrada de controle para o robô. Portanto, fazendo  $\mathbf{u}(t) = \mathbf{v}(x_1, x_2)$  tem-se um *campo vetorial*, ou seja, para cada ponto  $\mathbf{q}$  um vetor associado que representa a velocidade a ser seguida. A Figura 1 mostra um exemplo de campo vetorial gerado para guiar um robô para uma curva fechada em um espaço de duas dimensões. Este campo foi calculado com a metodologia mostrada nas próximas seções.

É importante observar que a consideração relativa ao Modelo (1) é uma simplificação do problema de controle de robôs reais. Tal modelo é válido apenas para robôs holonômicos com dinâmica muito simplificada. Provavelmente, esta simplificação só é válida na prática quando as velocidades ou acelerações envolvidas no movimento são muito pequenas e os efeitos inerciais do robô podem ser desprezados. Quando esses efeitos não puderem ser desprezados, pode-se dividir o problema de controle do robô em dois níveis de abstração, sendo que a metodologia proposta neste artigo estaria em um nível de abstração mais elevado, na qual o problema seria a geração de campos vetoriais para a aproximação e circulação da curva desejada. No nível de abstração mais baixo, o problema seria projetar um sistema de controle para assegurar que o robô siga esses campos vetoriais. Dessa forma, resolvendo os dois problemas de forma independente, teríamos a solução completa para controle de robôs. Neste artigo não pretende-se resolver o problema por completo, estando o foco na geração de campos vetoriais, como será visto a seguir.

### 2.1 Ideia Básica

Antes de apresentar a formulação matemática da metodologia, a mesma será explicada de maneira qualitativa e intuitiva. A Figura 2 serve como ilustração e abrange todos os passos da metodologia. Para facilitar o entendimento, assume-se que:

- a curva alvo está contida em um plano e é fechada;
- definido um sistema de eixos cartesianos ortogonais

em  $\mathbb{R}^2$  (coordenadas  $x_1$  e  $x_2$ ), a curva em questão pode ser descrita como uma curva implícita da forma  $\phi(x_1, x_2) = 0$  para uma função  $\phi$ ;

- a função  $\phi(x_1, x_2)$  é positiva fora da curva em questão, e negativa dentro. Ainda, para  $C_1 > C_2$ , sendo  $C_1$  e  $C_2$  constantes, as curvas de níveis  $\phi = C_1$  e  $\phi = C_2$  são fechadas e a curva de nível  $\phi = C_2$  está dentro da curva de nível  $\phi = C_1$ .

Como exemplo, se a curva alvo é uma circunferência de raio unitário centrada na origem, uma possível função é  $\phi(x_1, x_2) = x_1^2 + x_2^2 - 1$ . Deve-se então, alcançar dois objetivos: guiar o robô para a curva alvo e mantê-lo circulando nesta curva. Os dois problemas podem ser resolvidos separadamente e as suas soluções combinadas.

Para guiar o robô para a curva, observa-se as curvas de níveis da função  $\phi(x_1, x_2)$  (ou seja, as curvas formadas pela equação  $\phi(x_1, x_2) = C$  quando  $C$  varia). Devido à terceira hipótese acima, se  $C > 0$  a curva em questão está fora da curva alvo, se  $C < 0$  a curva está dentro, e se  $C = 0$  a curva em questão é a curva alvo. Em uma dada posição  $[x_1 \ x_2]^T$ , o robô está em uma curva de nível da função  $\phi(x_1, x_2)$ . Para guiá-lo para a curva alvo pode-se usar uma velocidade que é dada por um múltiplo escalar do gradiente da função  $\phi$  naquele ponto, uma vez que o gradiente é normal à curva de nível. Devido a terceira hipótese em  $\phi$  o gradiente aponta normalmente para fora. Assim, se o robô está em uma curva de nível exterior ( $\phi > 0$ ) deve-se utilizar uma constante negativa multiplicando o gradiente, fazendo assim com que a velocidade aponte para dentro (em direção à curva alvo). Se o robô estiver em uma curva interna ( $\phi < 0$ ), multiplica-se o gradiente por uma constante positiva (para que ele mantenha sua direção e portanto aponte para fora). Se o robô está na curva alvo ( $\phi = 0$ ), a constante deve ser zero (uma vez que não desejamos que o robô saia da curva).

Assim, a componente que aproxima da curva (que será chamada de *componente normal*) tem a forma:

$$\mathbf{N}(x_1, x_2) = \alpha(x_1, x_2) \nabla \phi(x_1, x_2).$$

O escalar  $\alpha(x_1, x_2)$  que multiplica o gradiente deve ser tal que este seja negativo quando o robô está em curvas de níveis externas (ou seja, quando  $\phi(x_1, x_2) > 0$ ), positivo para curvas de níveis internas (ou seja, quando  $\phi(x_1, x_2) < 0$ ) e nulo quando estamos na curva alvo ( $\phi(x_1, x_2) = 0$ ). Tendo em vista que a posição atual do robô em relação à curva alvo pode ser dada pelo sinal da função  $\phi$ , uma escolha de  $\alpha(x_1, x_2)$  simples é portanto:

$$\alpha(x_1, x_2) = G(\phi(x_1, x_2)),$$

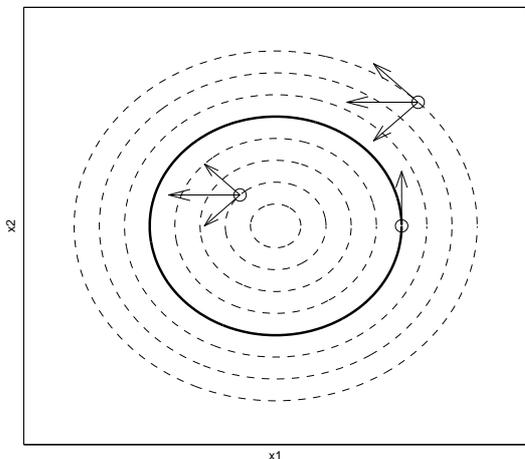


Figura 2: Robô em três pontos diferentes, para uma curva alvo circunferência (linha cheia). Em cada um desses pontos (um externo, um interno e um na curva) é demonstrado a componente normal, tangente e a soma de ambas. Quando o robô está na curva a componente normal é nula.

onde  $G$  é uma função com as propriedades  $G(y) > 0$  para  $y < 0$ ,  $G(y) < 0$  para  $y > 0$  e  $G(0) = 0$ . Como deseja-se que o movimento do robô seja o mais suave possível, escolhe-se uma função  $G$  que satisfaça essas restrições e ainda seja contínua. O exemplo mais simples de tal função é  $G(y) = -y$ .

Intuitivamente, esta componente normal guiará o robô para a curva. Para mantê-lo circulando, observa-se que se for utilizado um vetor ortogonal ao gradiente de  $\phi$  naquele ponto, o robô se manterá na curva de nível atual, circulando-a. A expressão de tal componente (que será chamada de *componente tangente*) é:

$$\mathbf{T}(x_1, x_2) = \beta(x_1, x_2) \left[ -\frac{\partial \phi}{\partial x_2} \quad \frac{\partial \phi}{\partial x_1} \right]^T.$$

O campo vetorial gerado pelo vetor que é multiplicado por  $\beta$  é chamado *campo Hamiltoniano* e portanto o símbolo  $\nabla_H \phi$  será utilizado para representá-lo. O múltiplo escalar  $\beta$  deve manter o sinal e ser não nulo em uma dada curva  $\phi = C$ , de modo que o robô mantenha-se circulando na curva de nível atual em um sentido fixo. De fato, escolhe-se então:

$$\beta(x_1, x_2) = H(\phi(x_1, x_2)),$$

sendo  $H(y)$  uma função que é não nula (isso garante que  $\beta$  é não nulo e tem o sinal mantido em uma dada curva de nível  $\phi(x_1, x_2) = C$ ) e cujo sinal de  $H(0)$  determina se a curva será percorrida no sentido negativo ou positivo. Um exemplo simples é  $H(y) = 1$ .

Somando os dois componentes em cada ponto, tem-se um campo vetorial dado por:

$$\mathbf{v}(x_1, x_2) = \mathbf{N}(x_1, x_2) + \mathbf{T}(x_1, x_2),$$

que se for usado como um vetor velocidade de um robô, o guiará para a curva alvo  $\phi = 0$  e, enquanto isso, o manterá circulando nas curvas de níveis em que passa. Portanto, foi gerado assim um ciclo limite para a curva fechada  $\phi(x_1, x_2) = 0$ . Quando o robô está na curva de nível desejada a componente normal se anula e tem-se apenas a tangencial (não nula) que o manterá circulando. A formalização matemática desta metodologia, que inclui provas de convergência e o relaxamento de algumas suposições iniciais é apresentada a seguir.

## 2.2 Formalização Matemática

Antes da apresentação do formalismo matemático para a metodologia apresentada intuitivamente anteriormente, faz-se necessário a seguinte definição:

**Definição 1** Uma função  $G : \mathbb{R} \mapsto \mathbb{R}$  neste texto é qualquer função que pode ser obtida como a derivada de uma função negativa definida diferenciável tal que a derivada se anule somente na origem.

Voltando ao exemplo da subseção anterior, onde  $G(y) = -y$ , note que  $G(y) = \frac{d}{dy}(-\frac{1}{2}y^2)$  e  $-\frac{1}{2}y^2$  é uma função negativa definida cuja derivada só se anula na origem. Essa definição garante que  $G(y)$  tenha as propriedades que foram apresentadas anteriormente:  $G(y) < 0$  para  $y > 0$ ,  $G(y) > 0$  para  $y < 0$  e  $G(0) = 0$ .

Com essa definição, é possível provar o resultado matemático principal dessa seção.

**Teorema 1** Seja  $\phi(x_1, x_2)$  uma função com as seguintes propriedades:

- A curva de nível  $\phi(x_1, x_2) = 0$  é conexa;
- As derivadas parciais de  $\phi$  são contínuas;
- $\nabla \phi$  é não nulo na curva  $\phi = 0$ .

Considere o seguinte sistema:

$$\dot{\mathbf{q}} = g(x_1, x_2)G(\phi)\nabla\phi + h(x_1, x_2)H(\phi)\nabla_H\phi, \quad (2)$$

com  $G(y)$  de acordo com a Definição 1,  $H(y)$  uma função contínua tal que  $H(0)$  é não nulo e  $g$  e  $h$  funções positivas e contínuas para qualquer  $x_1, x_2$ . Se todos os pontos de equilíbrio do sistema em questão, que são os pontos no qual

$\nabla\phi = \mathbf{0}$ , são instáveis, então a variável  $\mathbf{q}$  é conduzida para a curva  $\phi = 0$  e a mantém percorrendo em um sentido fixo para todas as condições iniciais  $\mathbf{q}_0$  que não são pontos de equilíbrio.

**Prova:** A prova vem diretamente do princípio de Krasovskii-LaSalle. Considere a função:

$$V(x_1, x_2) = - \int_0^{\phi(x_1, x_2)} G(y) dy,$$

que é positiva definida devido a própria definição da função  $G$ . Derivando com relação ao tempo tem-se:

$$\dot{V} = -G(\phi)\nabla\phi^T \dot{\mathbf{q}},$$

substituindo a expressão de  $\dot{\mathbf{q}}$  e usando o fato que  $\nabla\phi$  e  $\nabla_H\phi$  são ortogonais:

$$\dot{V} = -g(x_1, x_2)G(\phi)^2\|\nabla\phi\|^2.$$

Assim, devido as hipóteses em  $g$  e  $G$ ,  $\dot{V}$  é negativa semidefinida e nula apenas na curva alvo ( $\phi = 0$ ) ou nos pontos de gradiente nulo (que são pontos de equilíbrio). Como os pontos de equilíbrio são instáveis, por hipótese, o sistema se aproxima da curva  $\phi(x_1, x_2) = 0$ .

Uma vez na curva alvo, só restará a componente tangencial (já que  $G(0) = 0$ ) que manterá a variável  $\mathbf{q}$  percorrendo a curva em um dado sentido. A curva deve ser conexa pois caso contrário acontecem curvas degeneradas como, por exemplo, duas curvas disjuntas.  $\square$

Há algumas diferenças entre as hipóteses do Teorema 1 e a apresentação qualitativa da metodologia na subseção anterior. Na apresentação assumiu-se que a curva alvo era fechada, negativa dentro da curva e positiva fora dela (o que não faz nem sentido se a curva é aberta). Essas hipóteses não são necessárias e foram somente usadas para facilitar a explicação. Para a função  $H$ , que foi considerada inicialmente ser não nula, o Teorema 1 requer que ela seja não nula somente na origem, pois a circulação só é absolutamente necessária na curva desejada (o robô pode passar por outras curvas de níveis sem percorrê-las). Também foram introduzidas no Teorema 1, funções  $g$  e  $h$  acompanhando as componentes normal e tangencial, respectivamente. Elas não alteram o sentido e direção das componentes, uma vez que são estritamente positivas, mas existem para que seja possível que, por exemplo, o módulo da velocidade  $\dot{\mathbf{q}}$  seja constante (com escolhas adequadas de  $G, H, g$  e  $h$ ).

Para ilustrar o teorema foi escolhido um exemplo simples. Considere a curva descrita pela equação implícita  $x_1^4 + x_2^4 - 1 = 0$ . A escolha mais direta de  $\phi$  é  $\phi(x_1, x_2) = x_1^4 + x_2^4 - 1$ .

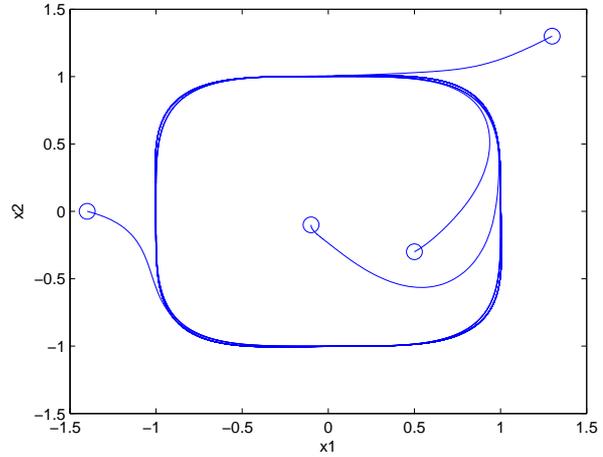


Figura 3: Simulações para a curva  $x_1^4 + x_2^4 - 1 = 0$  no sentido positivo, para quatro condições iniciais distintas.

Deseja-se convergir para essa curva e circulá-la no sentido positivo. Assim, escolhendo  $G(y) = -y$ ,  $g = h = H = 1$  temos o sistema:

$$\begin{aligned} \dot{x}_1 &= -(x_1^4 + x_2^4 - 1)4x_1^3 - 4x_2^3 \\ \dot{x}_2 &= -(x_1^4 + x_2^4 - 1)4x_2^3 + 4x_1^3, \end{aligned} \quad (3)$$

O único ponto em que  $\nabla\phi = \mathbf{0}$  é a origem  $x_1 = x_2 = 0$ . Para verificar que esse ponto de equilíbrio é instável, basta analisar a componente normal (que é responsável pela estabilidade)  $g(x_1, x_2)G(\phi(x_1, x_2))\nabla\phi(x_1, x_2) = [-(x_1^4 + x_2^4 - 1)4x_1^3 \quad -(x_1^4 + x_2^4 - 1)4x_2^3]^T$  perto do ponto  $x_1 = x_2 = 0$ . Observe que os termos de ordem maior que três ( $x_1^7, x_2^7, x_1^3x_2^4$  e  $x_1^4x_2^3$ ) são desprezíveis comparados com os termos de ordem três ( $x_1^3, x_2^3$ ) perto de  $x_1 = x_2 = 0$ , e, portanto, tem-se que o campo vetorial é aproximadamente igual a  $[4x_1^3 \quad 4x_2^3]^T$  próximo ao ponto. Um esboço rápido mostra que esse campo aponta para fora do ponto em questão. A Figura 3 mostra simulações para quatro condições iniciais distintas.

As próximas seções apresentam extensões do resultado teórico apresentado nesta seção. A Seção 3 apresenta um campo vetorial normalizado, que pode ser útil quando a velocidade do robô deve ser mantida constante durante a realização da tarefa, e a Seção 4 apresenta uma extensão da metodologia para curvas em espaços de  $n$  dimensões. Com esta extensão é possível o controle de robôs móveis em espaços de trabalhos tridimensionais ou o controle de robôs com  $n$  graus de liberdade.

### 3 CAMPO NORMALIZADO

Em aplicações práticas é muitas vezes desejável manter o módulo da velocidade  $\dot{\mathbf{q}}$  do robô constante ou, em outras palavras, é desejável um campo vetorial normalizado. Para esse fim, primeiro relaxa-se as hipóteses do Teorema 1 que exigem que  $g$  e  $h$  sejam contínuas e escolhe-se:

$$g(x_1, x_2) = \frac{1}{\|\nabla\phi\|},$$

$$h(x_1, x_2) = \frac{1}{\|\nabla_H\phi\|} = \frac{1}{\|\nabla\phi\|},$$

$$H(y) = \pm\sqrt{1 - G(y)^2}.$$

Além disso, adiciona-se a hipótese de que  $|G(y)| \leq 1$  (é escolhido um sinal para  $H$ ). O denominador das funções  $g$  e  $h$  (que podem causar descontinuidade) não será problema, uma vez que os pontos em que  $\nabla\phi = \mathbf{0}$  são, por hipótese, instáveis (entretanto eles não serão agora pontos de equilíbrio, mas sim pontos de descontinuidade do campo vetorial). Pela ortogonalidade das componentes normal e tangencial, observa-se que essa escolha garante de fato que  $\|\dot{\mathbf{q}}\| = 1$ .

Com o módulo da velocidade constante, o sistema está atrelado apenas à escolha de  $\phi$  e  $G(y)$ . De fato, existe um compromisso interessante entre a componente normal e a tangencial nessa situação: uma só cresce (em módulo) se a outra decresce. Assim, se  $G(y)$  decresce rapidamente para zero quando  $y$  se aproxima de zero,  $H(y)$  aumenta bruscamente. Qualitativamente, uma escolha de  $G$  pode regular se a trajetória é mais suave ou mais brusca, como será ilustrado a seguir.

Considere a curva descrita pela equação implícita  $\phi(x_1, x_2) = x_1^4/16 - 2x_1^2x_2^2/5 + x_2^4 - 1 = 0$ . O único ponto em que  $\nabla\phi = \mathbf{0}$  é a origem  $x_1 = x_2 = 0$ . Este ponto é instável como pode ser mostrado analisando-se a componente normal perto do ponto (não exatamente neste, uma vez que a componente normal é descontínua em  $x_1 = x_2 = 0$ ).

Foram escolhidas duas funções  $G(y)$  distintas:  $-\frac{2}{\pi} \arctan(10y)$  e  $-\frac{2}{\pi} \arctan(\frac{1}{2}y)$ , que estão ilustradas nas Figuras 4 e 5 respectivamente. Note que a primeira decresce bem mais bruscamente para zero em comparação com a segunda.

Nas Figuras 6 e 7, são apresentados dois grupos de simulações com as mesmas condições iniciais, mas com  $G(y)$  distintos. Note que com  $G(y) = -\frac{2}{\pi} \arctan(10y)$  a aproximação da curva ocorre mais bruscamente. Isso acontece pois a componente normal decresce substancialmente somente perto da curva alvo, e isso acontece rapidamente. Observe que a possibilidade de se ajustar a forma como a configuração converge para a curva alvo por meio da escolha de

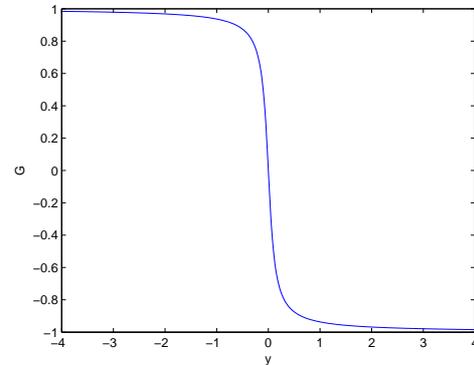


Figura 4: Função  $G(y) = -\frac{2}{\pi} \arctan(10y)$  no intervalo  $[-4, 4]$

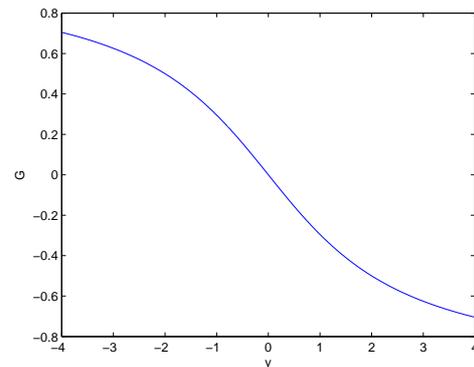


Figura 5: Função  $G(y) = -\frac{2}{\pi} \arctan(\frac{1}{2}y)$  no intervalo  $[-4, 4]$

$G(y)$  é muito interessante quando estamos trabalhando com robôs que possuem restrições que limitam o raio mínimo de curvatura a ser desenvolvido ao longo de seu caminho (por exemplo em um automóvel). Nestes casos  $G(y)$  pode ser escolhido de forma a contemplar o raio de curvatura mínimo do robô.

A próxima seção apresenta uma extensão da metodologia para dimensões superiores ( $\mathbb{R}^n$ ).

### 4 EXTENSÃO PARA DIMENSÕES SUPERIORES

A metodologia apresentada na seção anterior tem uma interpretação geométrica clara. É possível estender o Teorema 1 para curvas em  $\mathbb{R}^n$ , embora não seja mais possível valer-se tão fortemente da intuição geométrica.

Na seção anterior, a curva alvo era definida como uma curva implícita  $\phi(x_1, x_2) = 0$ . Na extensão da metodologia, para definir uma curva unidimensional em  $\mathbb{R}^n$  usa-

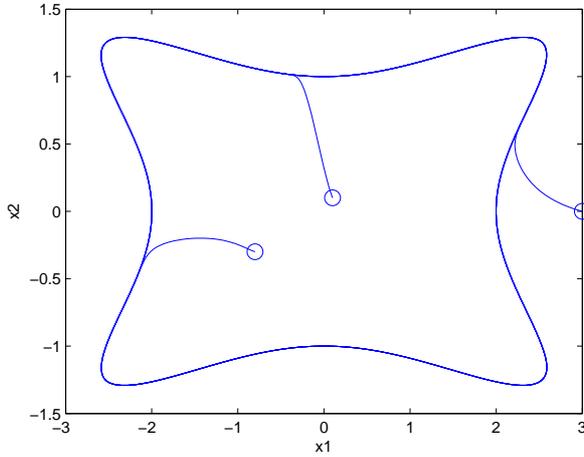


Figura 6: Simulação com três condições iniciais para a curva  $x_1^4/16 - 2x_1^2x_2^2/5 + x_2^4 - 1 = 0$  com  $G(y) = -\frac{2}{\pi} \arctan(10y)$

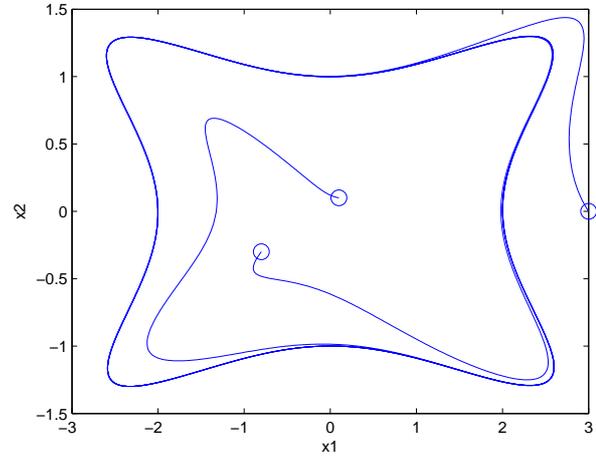


Figura 7: Simulação com três condições iniciais para a curva  $x_1^4/16 - 2x_1^2x_2^2/5 + x_2^4 - 1 = 0$  com  $G(y) = -\frac{2}{\pi} \arctan(\frac{1}{2}y)$

se a interseção de  $n - 1$  conjuntos de níveis de funções  $\phi_i(x_1, x_2, \dots, x_n)$ ,  $i = 1, 2, \dots, n - 1$ . Assim, a curva desejada é formada por todos pontos  $\mathbf{q} = [x_1 \ x_2 \ \dots \ x_n]^T$  em que  $\phi_i(x_1, x_2, \dots, x_n) = 0$  para  $i = 1, 2, \dots, n - 1$ .

Como exemplo de uma curva em  $\mathbb{R}^3$ , uma elipse no plano  $x_1x_2$  rotacionada em 45 graus no eixo  $x_2$  pode ser dada pela interseção dos conjuntos de níveis:

$$\begin{aligned}\phi_1(x_1, x_2, x_3) &= x_1^2 + x_2^2 - 1 = 0 \\ \phi_2(x_1, x_2, x_3) &= x_3 - x_1 = 0,\end{aligned}$$

sendo o primeiro um cilindro e o segundo, um plano. Então, para guiar o ponto  $\mathbf{q}$  para a curva, deve-se fazer com que  $n - 1$  funções  $\phi_i$  se anulam. Uma componente análoga à componente normal apresentada na seção anterior terá este objetivo. Para isso, usa-se o gradiente de uma função negativa definida cujos argumentos são as funções  $\phi_i$ . A Definição a seguir faz a apresentação formal.

**Definição 2** Uma função  $E : \mathbb{R}^{n-1} \mapsto \mathbb{R}$ , neste artigo, é qualquer função que é negativa definida, diferenciável em todos os seus argumentos e cujo gradiente se anula apenas na origem.

Como exemplo de  $E(y_1, y_2, \dots, y_{n-1})$ , a função  $E = -(y_1^2 + y_2^2 + \dots + y_{n-1}^2)$  é negativa definida e o seu gradiente  $\nabla E = [-2y_1 \ -2y_2 \ \dots \ -2y_{n-1}]^T$  só se anula quando  $y_1 = y_2 = \dots = y_{n-1} = 0$ .

Para manter um ponto  $\mathbf{q}$  percorrendo a curva em um sentido, observa-se que  $\frac{d}{dt}\phi_i(x_1, x_2, \dots, x_n) = \nabla\phi_i^T \dot{\mathbf{q}}$ . Para que uma vez na curva alvo ( $\phi_i = 0$  para  $i = 1, 2, \dots, n - 1$ ) o ponto não saia dessa curva, deve-se garantir que

$\frac{d}{dt}\phi_i(x_1, x_2, \dots, x_n) = \nabla\phi_i^T \dot{\mathbf{q}} = 0$  para  $i = 1, 2, \dots, n - 1$ . Isso implica que a componente tangente (que será a única componente a existir na curva alvo) deve ser ortogonal a cada um dos  $\nabla\phi_i$ . Se  $n - 1$  vetores em  $\mathbb{R}^n$  são linearmente independentes, existe apenas um vetor (sem contar múltiplos escalares) em  $\mathbb{R}^n$  ortogonal a todos eles. Isso motiva uma definição que faz-se necessária para o Teorema 1 estendido:

**Definição 3** Sejam  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n-1}$  vetores em  $\mathbb{R}^n$ , e

$$W = \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \dots \\ \mathbf{v}_{n-1}^T \\ \mathbf{0}^T \end{bmatrix}.$$

O produto de cunha  $\wedge_{i=1}^{n-1} \mathbf{v}_i$  desses vetores é outro vetor em  $\mathbb{R}^n$  cujos componentes são dados pela equação

$$(\wedge_{i=1}^{n-1} \mathbf{v}_i)_j = \text{cof}_{nj}(W),$$

ou seja, o  $j$ -ésimo elemento do vetor é o cofator  $nj$  da matriz  $W$  ( $(-1)^{n+j}$  vezes o determinante da matriz obtida retirando-se a  $n$ -ésima linha e  $j$ -ésima coluna da matriz original).

O produto de cunha é uma extensão do produto vetorial para  $\mathbb{R}^n$ . Para  $n = 3$  temos  $\wedge_{i=1}^2 \mathbf{v}_i = \mathbf{v}_1 \times \mathbf{v}_2$ . As propriedades são mantidas: trocar dois vetores troca o sinal do vetor resultante, além de que este é ortogonal a cada um dos  $n - 1$  vetores no produto. Se os vetores são linearmente dependentes o produto de cunha é o vetor nulo.

Com estas duas definições, é possível apresentar uma extensão para o Teorema 1. Mas antes, é importante ressaltar que no teorema a seguir  $E$  é uma função das variáveis  $\phi_i$ , mas como essas são funções das variáveis  $x_1, x_2, \dots, x_n$  isso implica que  $E$  também é uma função dessas. Assim, todo gradiente que aparece nesse artigo, com a exceção daquele na Definição 2, que é tomado com relação as variáveis  $y_1, y_2, \dots, y_{n-1}$ , é com relação às variáveis mais internas  $x_1, x_2, \dots, x_n$ .

**Teorema 2** *Sejam  $\phi_i(x_1, x_2, \dots, x_n)$ ,  $i = 1, 2, \dots, n - 1$ , funções com as seguintes propriedades:*

- A intercessão dos conjuntos de níveis  $\phi_i(x_1, x_2, \dots, x_n) = 0$  é uma curva em  $\mathbb{R}^1$ , conexa;
- As derivadas parciais das funções  $\phi_i$  são contínuas;
- $\wedge_{i=1}^{n-1} \nabla \phi_i$  é não nulo na intercessão dos conjuntos de níveis  $\phi_i = 0$ .

Então, considere o seguinte sistema:

$$\dot{\mathbf{q}} = g(\mathbf{q}) \nabla E(\phi_1, \dots, \phi_{n-1}) + h(\mathbf{q}) H(\phi_1, \dots, \phi_{n-1}) \wedge_{i=1}^{n-1} \nabla \phi_i, \quad (4)$$

com  $E(y_1, y_2, \dots, y_{n-1})$  de acordo com a Definição 2,  $H(y_1, y_2, \dots, y_{n-1})$  uma função contínua tal que  $H(0, 0, \dots, 0)$  é não nulo e  $g, h$  funções positivas e contínuas para qualquer  $x_1, x_2, \dots, x_n$ . Se todos os pontos de equilíbrio do sistema em questão, que são os pontos no qual  $\nabla E = \mathbf{0}$  que não estão na curva alvo, são instáveis, então, a variável  $\mathbf{q}$  é conduzida para a intercessão dos conjuntos de níveis  $\phi_i = 0$ ,  $i = 1, 2, \dots, n - 1$  e a mantém percorrendo em um sentido fixo para todas as condições iniciais  $\mathbf{q}_0$  que não são pontos de equilíbrio.

**Prova:** A prova é análoga a do Teorema 1. Considere a função:

$$V(x_1, x_2, \dots, x_n) = -E(\phi_1, \phi_2, \dots, \phi_{n-1}),$$

que é positiva definida devido a própria definição da função  $E$ . Derivando com relação ao tempo tem-se:

$$\dot{V} = -\nabla E^T \dot{\mathbf{q}}.$$

Agora, nota-se que:

$$\nabla E = \sum_{i=1}^{n-1} \frac{\partial E}{\partial \phi_i} \nabla \phi_i, \quad (5)$$

e portanto  $\nabla E$  é uma combinação linear dos gradientes  $\nabla \phi_i$ . Então, substituindo a expressão de  $\dot{\mathbf{q}}$  e usando o fato que

$\nabla E$  e  $\wedge_{i=1}^{n-1} \nabla \phi_i$  são ortogonais (pois  $\wedge_{i=1}^{n-1} \nabla \phi_i$  é ortogonal a cada um dos  $\nabla \phi_i$ ):

$$\dot{V} = -g(x_1, x_2, \dots, x_n) \|\nabla E\|^2.$$

Assim, devido as hipóteses em  $g$ ,  $\dot{V}$  é negativa semidefinida e se anula quando  $\nabla E = \mathbf{0}$ . Há dois tipos de pontos distintos em que  $\nabla E$  se anula. Como  $E$  é uma função negativa definida diferenciável então suas derivadas parciais (com relação as variáveis mais externas, no caso,  $\phi_i$ ) se anulam quando  $\phi_1 = \phi_2 = \dots = \phi_{n-1} = 0$  (na curva alvo). Outros pontos em que o gradiente se anula são pontos de equilíbrio do sistema (4), uma vez que  $\nabla E = \mathbf{0}$  sem estar na curva alvo implica que, em virtude da expressão (5), e do fato que devido a Definição 2 as derivadas parciais  $\frac{\partial E}{\partial \phi_i}$  não se anulam, ao menos que  $\phi_i = 0$  para  $i = 1, 2, \dots, n - 1$ , os vetores  $\nabla \phi_i$  são linearmente dependentes. Logo  $\wedge_{i=1}^{n-1} \nabla \phi_i$  também se anula. Desta forma, como os pontos de equilíbrio são instáveis, por hipótese, o sistema se aproxima da intercessão dos conjuntos de níveis  $\phi_i(x_1, x_2, \dots, x_n) = 0$ .

Uma vez na curva alvo, só restará a componente tangente que manterá a configuração  $\mathbf{q}$  percorrendo a curva em um dado sentido. Novamente, a curva deve ser conexa pois, caso contrário, acontecem curvas degeneradas como, por exemplo, duas curvas disjuntas.  $\square$

Para verificar que o Teorema 2 é uma generalização do Teorema 1, observe que para  $n = 2$ ,  $\nabla E(\phi) = \frac{dE}{d\phi} \nabla \phi$ . Note também que  $G$  na Definição 1 é exatamente  $\frac{dE}{d\phi}$  com  $E$  na Definição 2 (ou seja, para todo  $G$  de acordo com a Definição 1 existe um  $E$  de acordo com a Definição 2 tal que  $G = \frac{dE}{d\phi}$ ). Isso significa que a componente normal dos Teoremas 1 e 2 concordam quando  $n = 2$ . Para componente tangente, basta notar que, usando a Definição 3 com  $n = 2$ , temos que  $\wedge_{i=1}^1 \nabla \phi_i = \nabla_H \phi$ . Assim, o Teorema 2 se reduz ao Teorema 1.

O teorema será ilustrado com um exemplo: a curva alvo será a intercessão dos conjuntos de níveis  $\phi_1 = x_1^2 + x_2^2 - 1 = 0$  (cilindro) e  $\phi_2 = x_3 - x_1^2 = 0$  (parábola). Escolha-se  $E = -\frac{1}{2}(\phi_1^2 + \phi_2^2)$ ,  $g = h = H = 1$ . O único ponto de gradiente nulo de  $E$  que não está na curva alvo é a origem  $x_1 = x_2 = x_3 = 0$ . Esse ponto é de sela cuja região de atração é o eixo  $x_3$  (ou seja, para todas as condições iniciais do tipo  $[0 \ 0 \ x_3]^T$ , o sistema é atraído para a origem). Na Figura 8 está ilustrada a simulação para esta situação. Para facilitar a visualização de que de fato o ponto se aproxima da curva alvo, na Figura 9 é mostrada a função  $-E$  em função do tempo.

Para fins de curiosidade, provar-se-a um Teorema que mostra que o campo vetorial gerado pelo lado direito da expressão (4) não pode ser expresso como o gradiente de um campo escalar se a curva alvo for fechada.

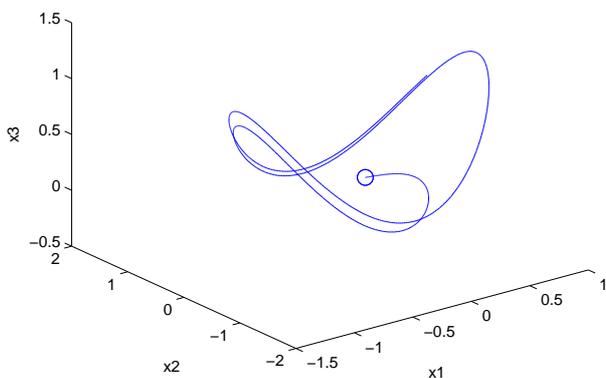


Figura 8: Simulação para a intercessão dos conjuntos de níveis  $\phi_1 = x_1^2 + x_2^2 - 1 = 0$  e  $x_3 - x_1 = 0$

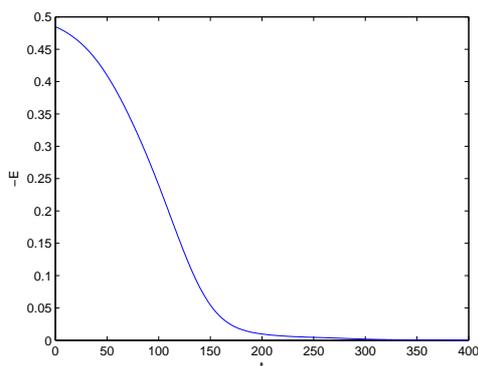


Figura 9: Função  $-E(t)$  para a simulação da Figura 8. Observe que, como esperado,  $-E(t) \rightarrow 0$  quando  $t \rightarrow \infty$

**Teorema 3** Se um campo vetorial induz um ciclo limite para uma curva fechada então este não pode ser um campo gradiente.

**Prova:** A prova vem por contradição. Considere que o campo  $\mathbf{v}$  é um campo gradiente e induz um ciclo limite para a curva  $\mathcal{C}$  em um dado sentido. Realizando uma integral de linha nessa curva e utilizando o mesmo sentido:

$$\oint_{\mathcal{C}} \mathbf{v}^T d\mathbf{q} = 0, \quad (6)$$

pois a curva  $\mathcal{C}$  é fechada e  $\mathbf{v}$  é um campo gradiente. Entretanto  $\mathbf{v}$  induz um ciclo limite para  $\mathcal{C}$ , ou seja, dirige o sistema para a curva e a mantém percorrendo em um sentido fixo. Assim tem-se que  $\mathbf{v}^T d\mathbf{q} > 0$  para todo ponto  $\mathbf{q}$  em  $\mathcal{C}$  (pois o vetor  $\mathbf{v}$  em  $\mathcal{C}$  tem o mesmo sentido e direção do vetor  $d\mathbf{q}$ , e é

não nulo, pois caso contrário não haveria o ciclo limite). Isso implica que a integral de linha deve ser positiva, o que é uma contradição.  $\square$

Se a curva alvo é aberta, o campo resultante pode ser um campo gradiente. Como exemplo, usando o Teorema 1 com a curva alvo  $x_2 = 0$  no sentido negativo direita para esquerda ( $H(y) = 1$ ) com  $g = h = 1$  e  $G(y) = -y$  tem-se  $\dot{\mathbf{q}} = [-1 \ -x_2]^T = \nabla(-x_1 - \frac{1}{2}x_2^2)$ .

Uma vez que foi proposta uma metodologia genérica para criação de campos vetoriais capazes de controlar robôs holonômicos no rastreamento de curvas em espaços de  $n$  dimensões, tornam-se necessárias metodologias para criação destas curvas e das funções que as geram. A próxima seção apresenta uma metodologia numérica capaz de gerar tais funções a partir de amostras do espaço, que em algumas aplicações podem ser consideradas como pontos de passagem para o robô.

## 5 GERAÇÃO DE CURVAS

Para construir o campo vetorial necessita-se obter as funções  $\phi_i$  (com as propriedades requeridas) tal que a intercessão dos conjuntos de níveis  $\phi_i = 0$  seja a curva desejada. Quando a curva não é suficientemente simples para as funções serem obtidas por inspeção, devem ser utilizados métodos numéricos. Nesta seção será apresentada a metodologia para construir curvas em  $\mathbb{R}^2$  proposta por (Turk e Brien, 2002), e posteriormente uma generalização desta metodologia para certos tipos de curvas em  $\mathbb{R}^3$ .

### 5.1 Curvas em $\mathbb{R}^2$

Como proposto em (Turk e Brien, 2002), funções implícitas podem ser construídas por meio de uma combinação linear de funções de bases radiais, definidas a seguir.

**Definição 4** Uma função de base radial é uma função  $f : \mathbb{R}^n \mapsto \mathbb{R}$  que depende apenas da distância euclidiana a um ponto  $\tilde{\mathbf{q}}_i$ :  $f_i(\mathbf{q}) = F(\|\mathbf{q} - \tilde{\mathbf{q}}_i\|)$ .

Uma função  $\phi_1$  pode ser então construída por meio de uma soma ponderada de funções de bases radiais com centros  $\tilde{\mathbf{q}}_i$  mais um termo constante  $-1$  (que impede que o sistema linear que será obtido para obter os pesos  $\omega_i$  seja homogêneo):

$$\phi_1(x_1, x_2) = \sum_{i=1}^n \omega_i f_i(x, y) - 1. \quad (7)$$

Em princípio, as únicas hipóteses a serem feitas acerca de  $F$  é que esta tenha como domínio  $\mathbb{R}^+$  (pois pela construção de

$\phi_1$  anterior, esta deve estar definida para valores não negativos) e seja diferenciável nesse mesmo intervalo (uma vez que usamos o gradiente de  $\phi_1$  no controlador e esse depende de  $F$ ). Aqui, a função escolhida é  $F(y) = py^2 \ln(py^2)$  com  $p > 0$ .

Suponha que a curva alvo seja amostrada em diversos pontos  $\tilde{\mathbf{q}}_i = [\tilde{x}_{1i} \ \tilde{x}_{2i}]^T$ ,  $i = 1, 2, 3, \dots, m$ . Deseja-se que  $\phi_1$  seja tal que a curva de nível  $\phi_1(x_1, x_2) = 0$  contenha estes pontos. Coloca-se então como  $m$  restrições que  $\phi_1(\tilde{x}_{1i}, \tilde{x}_{2i}) = \sum_{i=1}^n \omega_i f_i(\tilde{x}_{1i}, \tilde{x}_{2i}) - 1 = 0$  para  $i = 1, 2, 3, \dots, m$ . As restrições impostas por  $\tilde{\mathbf{q}}_i$  não são (na maioria das vezes) suficientes para construir uma curva de nível adequada. Mais restrições são necessárias devido a um ou mais destes fatores, a saber: (i) a curva de nível gerada pode ser inadequada (os pontos podem formar, por exemplo, duas curvas fechadas não conectadas), ou (ii) a função  $\phi_1$  pode não conter as propriedades desejadas (pontos de gradiente nulo estáveis, por exemplo).

Então, são necessárias *restrições internas*, ou seja, restrições do tipo  $\phi_1(x_j, y_j) = I_j < 0$  para  $[\tilde{x}_{1j} \ \tilde{x}_{2j}]^T$  contido dentro da curva desejada e *restrições externas*, que são restrições do tipo  $\phi_1(x_j, y_j) = E_j > 0$  para  $[\tilde{x}_{1j} \ \tilde{x}_{2j}]^T$  fora da curva fechada em questão. Essas restrições são utilizadas para corrigir os problemas da função original. Assim, tem-se  $n - m$  restrições adicionais ( $n > m$ ) de modo que o total seja de  $n$  restrições. Forma-se então um sistema linear com  $n$  equações e  $n$  incógnitas que pode ser facilmente resolvido por métodos numéricos tradicionais. A seguir será proposta uma extensão desta metodologia para criação de curvas em  $\mathbb{R}^3$ .

## 5.2 Curvas em $\mathbb{R}^3$

Para definir curvas em  $\mathbb{R}^3$ , são necessárias duas funções,  $\phi_1$  e  $\phi_2$ , cuja interseção dos conjuntos de níveis,  $\phi_1 = 0$  e  $\phi_2 = 0$ , seja a curva desejada. Ainda,  $\nabla\phi_1$  e  $\nabla\phi_2$  devem ser linearmente independentes nessa curva para que a componente tangente seja não nula. Esse problema será resolvido neste artigo para uma classe específica de curvas.

Assuma que a curva alvo é tal que sua projeção em um plano (que, por simplificação e sem perda de generalidade, assume-se que seja o plano  $x_1x_2$ ) seja unívoca, ou seja, para cada ponto da projeção existe apenas um ponto na curva associado (cuja projeção é o ponto em questão). Então, uma das curvas  $\phi_1(x_1, x_2, x_3)$  pode ser obtida projetando a curva no plano  $x_1x_2$  e se aplicando o método descrito na subseção anterior (a função resultante não dependerá de  $x_3$ ). Devido a propriedade de projeção unívoca, a segunda função pode ser escrita como  $x_3 = k(x_2, x_1)$  para uma função  $k$ . Amostrando pontos da curva usamos um método de interpolação para encontrar a função  $k$ . Assim, procedemos com  $\phi_2 = x_3 - k(x_1, x_2)$ .

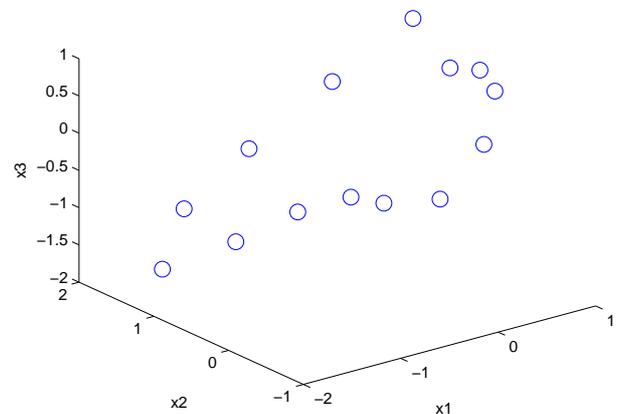


Figura 10: Pontos amostrados de uma curva fechada.

Os gradientes  $\nabla\phi_1$  e  $\nabla\phi_2$  sempre serão linearmente independentes na curva alvo se  $\nabla\phi_1$  não for nulo nessa curva. Para que esta propriedade fique clara, observe que  $\frac{\partial\phi_1}{\partial x_3} = 0$  enquanto  $\frac{\partial\phi_2}{\partial x_3} = 1$ . A dependência linear então implicaria que  $\nabla\phi_1 = \mathbf{0}$ . A seguir serão mostrados exemplos numéricos da aplicação desta metodologia.

## 5.3 Exemplo

Para ilustrar a metodologia de criação de curvas, a Figura 10 mostra 14 amostras de uma curva (fechada) tridimensional, que em uma aplicação de robótica aérea, por exemplo, podem ser considerados *waypoints* para o robô. A Figura 11 mostra a projeção dos pontos no plano  $x_1x_2$  para mostrar que ela é de fato unívoca. Como os pontos estão relativamente espaçados, pode-se usar um valor pequeno de  $p$  (neste exemplo,  $p = 1/30$ ) na função  $F(y) = py^2 \ln(py^2)$  de modo que a curva  $\phi_1 = 0$  seja fechada. Nesse caso, não são necessárias restrições internas ou externas, e portanto, após a resolução do sistema linear foram obtidos os pesos  $\omega_i$ . Para obter a função  $\phi_2$ , utilizou-se uma interpolação polinomial de duas variáveis de grau 4. Como são 14 pontos e um polinômio completo de quarta ordem em duas variáveis, tem-se 15 coeficientes associados. Assumiu-se neste exemplo que o coeficiente relativo ao termo  $x_1^2x_2^2$  é 0. As superfícies  $\phi_1 = 0$  e  $\phi_2 = 0$  e sua interseção, que forma a curva alvo em  $\mathbb{R}^3$ , podem ser vistas na Figura 12.

Após a criação das funções implícitas, estas foram utilizadas para controlar um robô holonômico. Para tanto, as seguintes

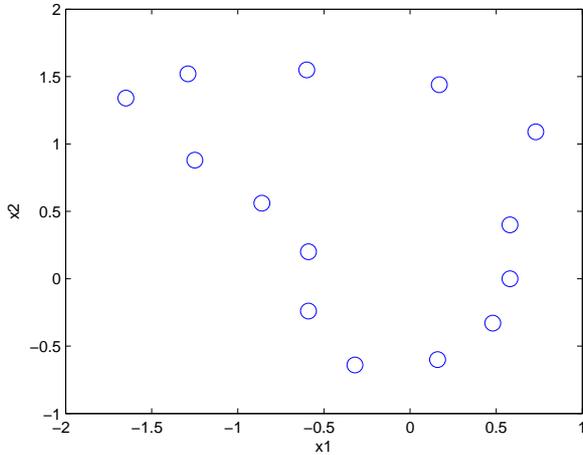


Figura 11: *Projeção dos pontos da Figura 10 no plano  $x_1x_2$ .*

funções na Equação (4) foram escolhidas:

$$E = - \int_0^{(\phi_1^2 + \phi_2^2)^{1/10}} \frac{2}{\pi} \arctan(y) dy,$$

$$g = 1 / \|\nabla(\phi_1^2 + \phi_2^2)^{1/10}\|,$$

$$h = 1 / \|\wedge_{i=1}^2 \nabla \phi_i\|,$$

$$H = \sqrt{1 - \left(\frac{2}{\pi} \arctan((\phi_1^2 + \phi_2^2)^{1/10})\right)^2},$$

de modo que  $\|\dot{\mathbf{q}}\| = 1$ . A condição inicial para o sistema foi  $x_1 = 1$ ,  $x_2 = 0.1$  e  $x_3 = 0.1$ . A Figura 13 mostra o resultado da simulação, onde os *waypoints* são representados pelos círculos menores e a configuração inicial do robô pelo círculo maior.

## 6 CONSIDERAÇÕES SOBRE APLICAÇÕES A ROBÔS REAIS

A metodologia teórica como explicada até agora, assume um robô holonômico representado por sua configuração exata  $\mathbf{q}$ . Com esta teoria, é possível, por exemplo, o controle de um manipulador robótico holonômico com  $n$  graus de liberdade para o qual, devido a presença de sensores absolutos de posição nas juntas,  $\mathbf{q}$  pode ser conhecido com grande exatidão. Talvez a maior dificuldade neste caso seja a definição das funções implícitas no espaço  $\mathbb{R}^n$ . Também, é importante lembrar que a metodologia proposta neste artigo não considera a dinâmica do robô e, portanto, é muitas vezes necessário um controlador de mais baixo nível responsável por garantir que o robô siga o campo vetorial. Robôs holonômicos com orientação são também facilmente controlados com o campo proposto se sua configuração puder ser facilmente medida. No caso de um robô se movendo em um plano, por

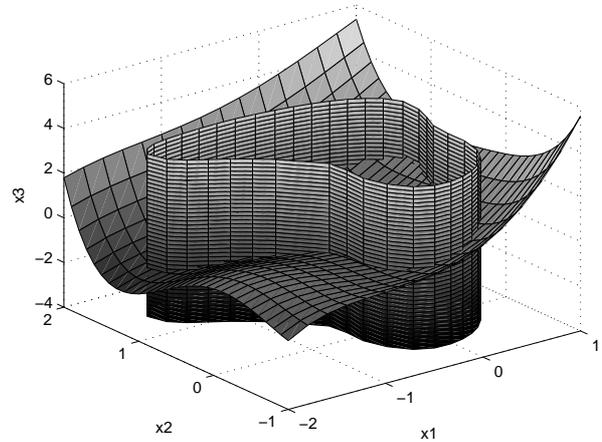


Figura 12: *Superfícies cuja intercessão gera a curva alvo relativa aos pontos amostrados da Figura 10.*

exemplo, a configuração  $\mathbf{q}$  teria dimensão três (posição em  $x$ , posição em  $y$  e orientação) e portanto, o robô poderia ser controlado para rastrear um ciclo limite em um espaço não-Euclidiano de três dimensões.

No entanto, grande parte dos robôs possui restrições cinemáticas e utilizam sensores ruidosos para se localizar. Torna-se então natural questionar quando a metodologia proposta poderia ser aplicada a um robô não-holonômico com configuração estimada  $\hat{\mathbf{q}}$ . Apesar de não ser o foco deste artigo a implementação da metodologia teórica e sua experimentação, a literatura da área de robótica apresenta algumas ideias que podem responder a este questionamento.

Para levar em consideração as restrições não-holonômicas, um controlador baseado em linearização por realimentação de estados poderia ser utilizado em vários casos para fazer o robô seguir o campo vetorial (Slotine e Li, 1991). Suponha, por exemplo, um robô diferencial se movendo no plano. Deseja-se que este robô convirja e circule uma curva fechada em seu espaço de trabalho. Por meio da definição do ponto de controle do robô a uma distância  $d$  à frente do seu centro de massa, é possível escrever:

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{\sin \theta}{d} & \frac{\cos \theta}{d} \end{bmatrix} \mathbf{u}(\mathbf{q}), \quad (8)$$

onde  $V$  e  $\omega$  são as velocidades linear e angular do robô, respectivamente,  $\theta$  é a orientação do robô e  $\mathbf{u}(\mathbf{q})$  é o campo vetorial dado pelo lado direito da Equação (2). Para diferentes tarefas e robôs com modelos dinâmicos e cinemáticos diversos, outros controladores devem ser desenvolvidos. Se, por exemplo, for necessário, além do controle da posição do robô planar também o controle de sua orientação, além da geração de um campo em seu espaço de configurações tridimensio-

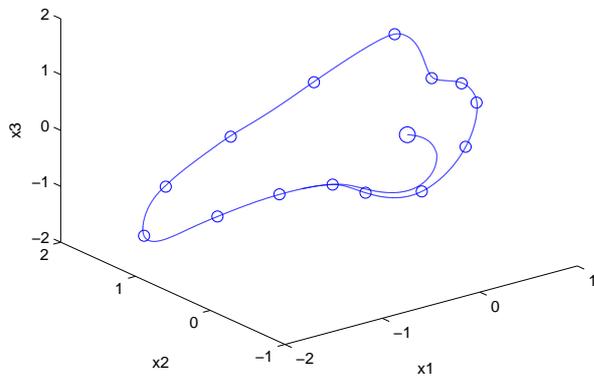


Figura 13: Trajetória seguida por um robô holonômico quando são usadas as funções da Figura 12.

nal, torna-se necessária a utilização de um controlador baseado na linearização dinâmica do modelo, ao contrário do que é feito na Equação (8), que considera apenas uma linearização estática.

A estimativa de  $\mathbf{q}$  é um dos problemas de mais difícil solução em robótica móvel. Várias boas soluções já foram propostas na literatura para localização em ambientes internos, principalmente quando existe alguma estrutura conhecida no ambiente (Thrun et al., 2005). No entanto, é ainda muito difícil a localização com grande precisão em ambientes externos ou não estruturados. Uma solução conhecida é a utilização de filtros recursivos para combinar a informação proveniente de diversas fontes de dados. O filtro de Kalman Estendido, por exemplo, é extensivamente utilizado para combinar odometria e informação inercial com dados de GPS (*Global Positioning System*) (Thrun et al., 2005). No entanto, uma importante característica de metodologias baseadas em campos de potencial contínuos é que elas são robustas a pequenos erros de localização. Assim, mesmo que as estimativas de localização possuam pequenos erros e desvios, o robô ainda continuará se movendo na direção correta.

Para mostrar o potencial da metodologia teórica para o controle de robôs reais, a Figura 14 apresenta uma simulação realizada no ambiente Player/Stage (Gerkey et al., 2003), onde a cinemática do robô pode ser facilmente simulada. Além de uma restrição não-holonômica, foram adicionados ruídos gaussianos de média nula e desvios padrões iguais a 10cm e 4,5 graus à posição e à orientação do robô, respectivamente. Este desvios são iguais ou maiores do que aqueles obtidos por sistemas de localização visual em ambiente interno (Garcia et al., 2007). Nesta simulação, o robô Pioneer II-DX simulado foi controlado para rastrear a curva mos-

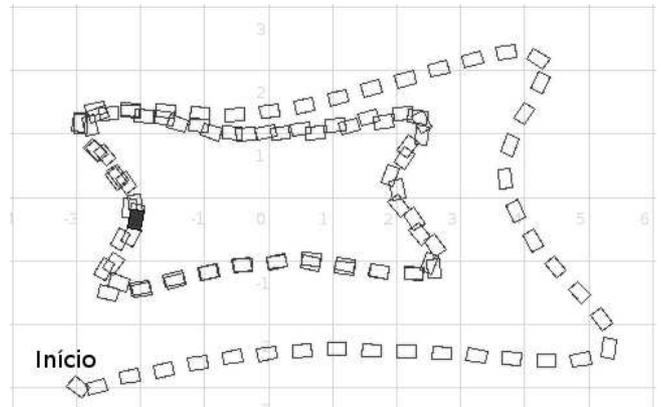


Figura 14: Simulação realizada na plataforma Player/Stage de um robô não-holonômico sujeito a erros de localização rastreado a curva da Figura 7.

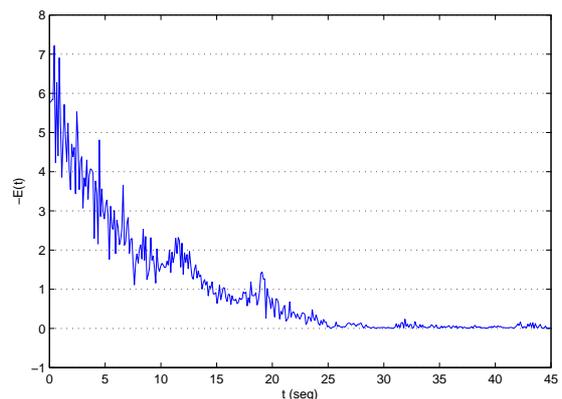


Figura 15: Função  $-E(t)$  para a simulação da Figura 14. Observe que, mesmo com a presença de ruído na localização e a restrição não-holonômica no robô  $-E(t) \rightarrow 0$  quando  $t \rightarrow \infty$ .

trada na Figura 7. O campo foi seguido com a utilização do controlador mostrado na Equação (8). A Figura 15 mostra o comportamento da função  $-E(t)$  para esta simulação.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Este artigo apresentou uma metodologia para construção de campos vetoriais contínuos capazes de controlar um robô holonômico para aproximar e circular uma curva definida por funções implícitas. A ideia básica da metodologia é a construção de um campo vetorial que possua um ciclo limite sobre a curva desejada. A principal contribuição do artigo é a construção de ciclos limites no espaço  $n$ -dimensional, o que até então não tinha sido encontrado na literatura. São apresentadas provas de convergência, inclusive para campos

normalizados, úteis quando se deseja manter a velocidade do robô constante.

O artigo ainda mostra como as funções implícitas podem ser numericamente calculadas a partir de amostras da curva alvo desejada. É mostrada uma extensão para ambiente tridimensionais da metodologia bidimensional proposta por Turk e Brien (2002) e já utilizada para navegação de robôs móveis em (Chaimowicz et al., 2005). Esta extensão permitirá a utilização da metodologia para o controle de robôs aéreos, por exemplo.

Como continuidade, está sendo considerando o caso de curvas variantes no tempo conforme resultados iniciais apresentados em (Gonçalves et al., 2009). Trabalhos futuros também incluem a implementação da metodologia em robôs terrestres e aéreos. Outra metodologia de campos vetoriais proposta por nosso grupo já foi utilizada para guiar um aeromodelo autônomo com grande sucesso (Iscolde et al., 2010).

## AGRADECIMENTOS

Os autores gostariam de agradecer o suporte financeiro da Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). A implementação da metodologia na plataforma Player/Stage foi realizada por Bruno Dutra, a quem os autores expressam seus sinceros agradecimentos.

## REFERÊNCIAS

- Adorno, B. V. e Borges, G. A. (2006). Um método de planejamento de trajetória para robôs móveis através de passeios aleatório adaptativos e mapa de rotas, *Anais do Congresso Brasileiro de Automática*, pp. 3188–3193.
- Brandão, A. S., Filho, M. S. e Filho, T. F. B. (2006). Navegação de robôs móveis com desvio de obstáculos: Implementação de desvio tangencial modificado, *Anais do Simpósio Brasileiro de Automação Inteligente*.
- Ceccarelli, N., Marco, M. D. e Giannitrapani, A. (2008). Collective circular motion of multi-vehicle systems, *Automatica* **44**: 3025–3035.
- Celeste, W. C., Filho, T. F. B. e Filho, M. S. (2008). Controle de seguimento de caminho com sinais de comando limitados aplicado a robôs móveis unicyclos, *Anais do Congresso Brasileiro de Automática*.
- Chaimowicz, L., Michael, N. e Kumar, V. (2005). Controlling swarms of robots using interpolated implicit functions, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2498–2503.
- Chaves, L. F. e Lizarralde, F. (2006). Uma nova solução ao problema de navegação de robôs móveis autônomos, *Anais do Congresso Brasileiro de Automática*, pp. 2045–2050.
- Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. e Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*, The MIT Press.
- Connolly, C. I., Burns, J. B. e Weiss, R. (1990). Path planning using Laplace's equation, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2102–2106.
- da Costa e Silva Franco, A., da Costa, A. C. L., Junior, J. C. L. e Bitencourt, A. C. P. (2008). Geração e controle de trajetória para robôs móveis omnidirecionais, *Anais do Congresso Brasileiro de Automática*.
- de Lima Ottoni, G. e Lages, W. F. (2005). Navegação de robôs móveis em ambientes desconhecidos usando sonares e ultra-som, *Revista Controle & Automação* **14**(4): 402–411.
- Faria, G. e Romero, R. A. F. (2002). Navegação de robôs móveis utilizando aprendizagem por reforço e lógica fuzzy, *Revista Controle & Automação* **13**(3): 219–230.
- Frew, E. W., Lawrence, D. A., Dixon, C., Elston, J. e Pisano, W. J. (2007). Lyapunov guidance vector fields for unmanned aircraft applications, *Proceedings of the American Control Conference*, pp. 371–376.
- Garcia, R. F., Shiroma, P. M., Chaimowicz, L. e Campos, M. F. M. (2007). Um arcabouço para a localização de enxames de robôs, *Anais do Simpósio Brasileiro de Automação Inteligente*.
- Gerkey, B., Vaughan, R. T. e Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems, *Proceedings of the 11th International Conference on Advanced Robotics*, pp. 317–323.
- Gonçalves, V. M., Pimenta, L. C. A., Maia, C. A. e Pereira, G. A. S. (2009). Artificial vector fields for robot convergence and circulation of time-varying curves in n-dimensional spaces, *Proceedings of the American Control Conference*, pp. 2012–2017.
- Gonçalves, V. M., Maia, C. A. e Pereira, G. A. S. (2008). Navegação de robôs móveis utilizando ciclos limite determinados por meio de curvas implícitas, *Anais do XVII Congresso Brasileiro de Automática (CBA'08)*.
- Hsieh, M. A. e Kumar, V. (2006). Pattern generation with multiple robots, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2442–2447.

- Hsieh, M. A., Loizou, S. e Kumar, V. (2007). Stabilization of multiple robots on stable orbits via local sensing, *Proceedings of the IEEE International Conference on Robotics Automation*, pp. 2312–2317.
- Iscold, P., Pereira, G. A. S. e Torres, L. A. B. (2010). The development of a hand-launched small uav for ground reconnaissance, *IEEE Transactions on Aerospace and Electronic Systems* (artigo aceito para publicação).
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *International Journal of Robotics Research* **5**(1): 90–98.
- Langer, R. A., Oliveira, G. H. C. e dos S. Coelho, L. (2006). Uma abordagem tipo bug para planejamento de trajetórias em robôs móveis, *Anais do Congresso Brasileiro de Automática*, pp. 1007–1012.
- Lawrence, D. A., Frew, E. W. e Pisano, W. J. (2009). Lyapunov vector fields for autonomous unmanned aircraft flight control, *Journal of Guidance, Control, and Dynamics* **31**(5): 1220–1229.
- Pereira, G. A. S. e Chaimowicz, L. (2007). Robôs móveis, in L. A. Aguirre, A. P. A. da Silva, M. F. M. Campos e W. C. do Amaral (eds), *Enciclopédia de Automática: Controle e Automação*, Vol. 3, Editora Blucher, São Paulo, pp. 369–386.
- Pereira, G. A. S., Rebelo, D. R., Iscold, P. e Torres, L. A. B. (2008). A vector field approach to guide small UAVs through a sequence of waypoints, *Anais do XVII Congresso Brasileiro de Automática (CBA'08)*.
- Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M. e Campos, M. F. M. (2006). Robot navigation based on electrostatic field computation, *IEEE Transactions on Magnetics* **42**(4): 1459–1462.
- Pimenta, L. C. A., Mendes, M. L., Mesquita, R. C. e Pereira, G. A. S. (2007). Fluids in electrostatic fields: An analogy for multi-robot control, *IEEE Transactions on Magnetics* **43**: 1765–1768.
- Pimenta, L. C. A., Pereira, G. A. S. e Mesquita, R. C. (2007). Fully continuous vector fields for mobile robot navigation on sequences of discrete triangular regions, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1992–1997.
- Pisano, W. J., Lawrence, D. A. e Mohseni, K. (2006). Concentration gradient and information energy for decentralized uav control, *AIAA Guidance, Navigation, and Control Conference and Exhibit*.
- Quigley, M., Goodrich, M. A., Griffiths, S., Eldredge, A. e Beard, R. (2005). Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled camera, *Proceedings of the IEEE International Conference on Robotics Automation*, pp. 2600–2605.
- Rimon, E. e Koditschek, D. E. (1992). Exact robot navigation using artificial potencial functions, *IEEE Transactions on Robotics and Automation* **8**(5): 501–517.
- Slotine, J.-J. e Li, W. (1991). *Applied Nonlinear Control*, Prentice Hall.
- Tavares Neto, R. F. e dos Santos Coelho, L. (2005). Planejamento de rotas para robôs de inspeção usando um algoritmo híbrido de colônia de formigas e algoritmo cultural, *Anais do Simpósio Brasileiro de Automação Inteligente*.
- Thrun, S., Burgard, W. e Fox, D. (2005). *Probabilistic Robotics*, The MIT Press.
- Turk, G. e Brien, J. F. (2002). Modelling with implicit surfaces that interpolate, *ACM Transactions of Graphics* pp. 855–873.
- Zhang, F. e Leonard, N. E. (2007). Coordinated patterns of unit speed particles on a closed curve, *Systems and Control Letters* **56**(6): 397–407.