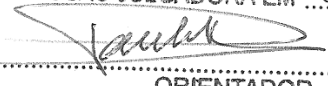


ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA  
TESE DEFENDIDA POR Elvira Rafikova  
..... E APROVADA  
PELA COMISSÃO JULGADORA EM 26.1.02.12010  
  
.....  
ORIENTADOR

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA  
MECÂNICA**

# **Controle de um Robô Móvel Através de Realimentação de Estados Utilizando Visão Estereoscópica**

Autor: Elvira Rafikova  
Orientador: Paulo Roberto Gardel Kurka

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA  
MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

# **Controle de um Robô Móvel Através de Realimentação de Estados Utilizando Visão Estereoscópica**

Autor: Elvira Rafikova

Orientador: Paulo Roberto Gardel Kurka

Curso: Engenharia Mecânica

Área de Concentração: Mecânica dos Sólidos e Projeto Mecânico

Tese de doutorado apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Doutor em Engenharia Mecânica.

Campinas, fevereiro 2010  
S.P. – Brasil

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE -  
UNICAMP

R124c Rafikova, Elvira  
Controle de um robô móvel através de realimentação  
de estados utilizando visão estereoscópica / Elvira  
Rafikova. --Campinas, SP: [s.n.], 2010.

Orientador: Paulo Roberto Gardel Kurka.  
Tese de Doutorado - Universidade Estadual de  
Campinas, Faculdade de Engenharia Mecânica.

1. Robôs móveis. 2. Sistemas de controle . 3.  
Navegação de robôs móveis. 4. Visão por computador.  
I. Kurka, Paulo Roberto Gardel. II. Universidade  
Estadual de Campinas. Faculdade de Engenharia  
Mecânica. III. Título.

Título em Inglês: Feedback control of a mobile robot using stereo vision

Palavras-chave em Inglês: Mobile Robot, Control systems , Mobile robot  
navigation, Computer vision

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Doutor em Engenharia Mecânica

Banca examinadora: Marco Henrique Terra, João Antônio Pereira, João  
Maurício Rosário, Auteliano Antunes dos Santos Junior

Data da defesa: 26/02/2010

Programa de Pós Graduação: Engenharia Mecânica

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA  
MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

**TESE DE DOUTORADO**

**Controle de um Robô Móvel Através de  
Realimentação de Estados Utilizando Visão  
Estereoscópica**

**Autor: Elvira Rafikova**

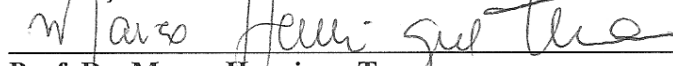
**Orientador: Paulo Roberto Gardel Kurka**

A Banca Examinadora composta pelos membros abaixo aprovou esta Tese:



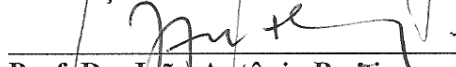
**Prof. Dr. Paulo Roberto Gardel Kurka**

**Instituição: UNICAMP**



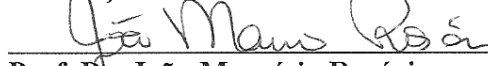
**Prof. Dr. Marco Henrique Terra**

**Instituição: USP- São Carlos**



**Prof. Dr. João Antônio Perreira**

**Instituição: UNESP- Ilha Solteira**



**Prof. Dr. João Maurício Rosário**

**Instituição: UNICAMP**



**Prof. Dr. Auteliano Antunes dos Santos Junior**

**Instituição: UNICAMP**

Campinas, 26 de fevereiro de 2010

## **Dedicatória:**

Para Marat Rafikov e Rimma Rafikova.

## **Agradecimentos**

Agradeço aqui, sem nenhuma ordem, a todos que participam da minha vida e principalmente da minha história acadêmica

Ao professor Paulo Kurka, não somente pela orientação, mas também pela amizade. Ao meu pai, que contribuiu enormemente para esse trabalho. Ao professor Fernando Lobo Pereira pela colaboração internacional e acolhimento na Universidade do Porto, em Portugal.

Agradeço aos meus colegas e amigos, Luciana Diógenes, Carlos Mingoto Junior, e aos colegas de laboratório pelo auxílio e apoio.

À Mauren Schweigert Pereira e Renira Carla Soares pela amizade infinita e atemporal.

À Karen de Lolo Guilherme Paulino com seus conselhos fundamentais e sua calma angelical diante das dificuldades.

Aos colegas da FEM, Francisco Ilson da Silva pela amizade incondicional que iniciou na cidade luz, maravilhosa Paris.

A todos que me suportaram , criticaram , elogiaram, odiaram ou amaram, agradeço profundamente.

*“Aquilo a que voce resiste, persiste.”*

Carl Jung.

## Resumo

RAFIKOVA, Elvira, *Controle de um Robô Móvel Através de Realimentação de Estados Utilizando Visão Estereoscópica*. 2010. 183 p.

Tese (Doutorado) - Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

O enfoque principal desse trabalho é o controle de trajetória e navegação no ambiente através da visão estereoscópica de um robô móvel de duas rodas de acionamento diferencial. Para o controle de posicionamento, são utilizadas: uma estratégia de controle ótima linear e uma estratégia subótima, não linear, em tempo contínuo, chamada de SDRE (*State Dependent Riccati Equation*), e por fim, uma estratégia de controle SDRE em tempo discreto. Todas essas estratégias são baseadas em funções de Lyapunov e aplicadas ao problema de regulação do robô a uma referência. Para a navegação do robô no ambiente é considerado um modelo navegação por odometria e um mecanismo de visão estereoscópica. A estimação do estado é realizada através do filtro de Kalman clássico. São apresentadas duas estratégias para a navegação do robô no ambiente. Uma delas, totalmente discreta com a utilização do métodos de controle SDRE discreto, observação de estado discreta através das câmeras e estimação de estado através do filtro de Kalman discreto. Outra, com a abordagem de horizonte recuável, utilizando controle SDRE contínuo e, observação e estimação de estado discretas. A eficácia dos métodos de controle e das estratégias de navegação do robô é verificada através de



simulações computacionais, nas quais a estratégia de navegação com horizonte recuável se mostra eficaz para a navegação precisa no ambiente.

*Palavras Chave:*

- Robôs Móveis, Sistemas de Controle por Realimentação de Estados, Navegação de Robôs Móveis, Visão Estereoscópica.

## Abstract

RAFIKOVA, Elvira, *Feedback Control of a Mobile Robot Using Stereo Vision.*, 2010. 164 p.  
Thesis (PhD in Mechanical Engineering): Faculty of Mechanical Engineering, State University of Campinas, Campinas.

The main approach of this thesis is the trajectory control and navigation of a differential steering mobile robot in the environment.

For the position control problem are used? A continuous-time , linear feedback control; a suboptimal, nonlinear , continuous-time feedback called SDRE (StateDependent Riccati Equation) control and a discrete – time SDRE control method. All of these methods are Lyapunov functions based and appplied to the reference tracking problem oh the nonholonomic robot.

For the purpose of the environmental navigation a model of odometry-stereo vision state observation system is considered. Meanwhile, the state estimation is given by classic Kalman filter.

Futhermore, two different navigation strategies are presented. The discret-time one, using both discret SDRE control method and state estimation. Another one, is a receding horizon strategy, using continuous-time SDRE controler and scret-time state estimation. The control method and navigation strategies eficaccy is verified through numerical simulations. Both navigation strategies demonstrate good results, although the receding horizon one provides more precise navigation.

### *Key Words*

Mobile Robot, Feedback Control Systems, Mobile Robt Navigation, Stereo Vision System

## Lista de Ilustrações

Figura 2. 1. Modelo Cinemático de Robô Móvel.....	11
Figura 2. 2 Variação da Coordenada $\theta$ no Robô .....	12
Figura 3. 1- Ângulos e Coordenadas de um Sistema de Duas Câmeras no Mundo. ....	21
Figura 3. 2 –Cálculo de Coordenadas para Localização do Robô Através dos Pontos P e A..	23
Figura 6. 1- Esquema do Sistema de Navegação .....	61
Figura 6. 2- Curva de Retorno no Sentido Anti-horário .....	66
Figura 6. 3 - Curva de Retorno no Sentido Horário .....	67
Figura 6. 4- Curva de Desvio no Sentido Anti-horário .....	68
Figura 6. 5- Curva de Desvio no Sentido Horário.....	69
Figura 7. 1 -Trajetória do Robô, Referência Reta, Método: Linear Por realimentação de estado Ótimo	80
Figura 7. 2- Trajetória em Desvios , Referência reta, Método : Linear Por realimentação de estado Ótimo. ....	81
Figura 7. 3-Trajétória Circular para Parâmetros $\theta_0 = 0, x_0 = 1, y_0 = 0$ , raio=2 .....	82

Figura 7. 4-Trajétória do Sistema para uma Trajetória de Referência Circular, Controle SDRE .....	83
Figura 7. 5-Trajétória em Desvios, Controle SDRE .....	84
Figura 7. 6- Trajetória do Sistema Evoluindo para Trajetória de Referência Reta, Controle SDRE.....	85
Figura 7. 7- Trajetória em Desvios, Controle SDRE .....	85
Figura 7. 8- Evolução Temporal das Coordenadas em Desvio, Controle por Realimentação de Estado Linear Ótimo .....	87
Figura 7. 9- Evolução Temporal das Coordenadas em Desvio, Controle por SDRE .....	88
Figura 7. 10- Trajetória do Robô Obtida pelos Dois Métodos.....	88
Figura 7. 11- Trajetória Controlada por SDRE Discreto no Tempo .....	90
Figura 7. 12- Coordenadas em Desvios Controladas por SDRE Discreto no Tempo.....	90
Figura 7. 13- Parâmetros P1 a) Trajetória Controlada SDRE b)Coordenadas em Desvios .....	93
Figura 7. 14- Parâmetros P2 a) Trajetória Controlada SDRE b)Coordenadas em Desvios .....	94
Figura 7. 15- Parâmetros P3 a) Trajetória Controlada SDRE b)Coordenadas em Desvios .....	95
Figura 7. 16- Parâmetros P4 a) Trajetória Controlada SDRE b) Coordenadas em Desvios .....	96
Figura 7. 17- Trajetória Controlada SDRE .....	97
Figura 7. 18-Coordenadas em Desvios Trajetória Circular.....	97
Figura 7. 19- Trajetória Controlada SDRE .....	98
Figura 7. 20- Coordenadas em Desvios Trajetória Circular.....	98

Figura 7. 21- Trajetória Controlada SDRE .....	99
Figura 7. 22- Coordenadas em Desvios Trajetória Circular.....	99
Figura 7. 23- Trajetória Controlada SDRE .....	100
Figura 7. 24- Coordenadas em Desvios Trajetória Circular.....	100
Figura 7. 25-Trajetoária Controlada Reta e Círculo .....	101
Figura 7. 26- Trajetória Estimada pela Visão. Resolução de 320 pixels. ....	103
Figura 7. 27- Trajetória Estimada pela Visão . Resolução de 640 pixels. ....	103
Figura 7. 28- Trajetória Estimada pela Odometria, erro menor. ....	105
Figura 7. 29- Trajetória Estimada pela Odometria, erro maior. ....	105
Figura 7. 30- Trajetória de Navegação do Robô com Parâmetros D1 .....	108
Figura 7. 31- Trajetória de Navegação do Robô com Parâmetros D2 .....	108
Figura 7. 32-Trajetoária de Navegação do Robô com Parâmetros D3 .....	109
Figura 7. 33- Trajetória de Navegação do Robô com Parâmetros D4 .....	110
Figura 7. 34- Trajetória de Navegação do Robô com Parâmetros C1.....	112
Figura 7. 35- Trajetória de Navegação do Robô com Parâmetros C2.....	113
Figura 7. 36- Trajetória de Navegação do Robô com Parâmetros C3.....	113
Figura 7. 37- Trajetória de Navegação do Robô com Parâmetros C4.....	114
Figura 7. 38- Trajetória de Navegação do Robô com Parâmetros C5.....	116
Figura 7. 39- Trajetória de Navegação do Robô com Parâmetros C6.....	116

Figura 7. 40- Trajetória de Navegação do Robô com Parâmetros C7.....	117
Figura 7. 41- Trajetória de Navegação do Robô com Parâmetros C8.....	118

## Lista de Tabelas

Tabela 7. 1- Valores de Conjuntos de Parâmetros para Simulação Controle SDRE .....	92
Tabela 7. 2-Parâmetros para Navegação Discreta.....	106
Tabela 7. 3- Parâmetros de Navegação com Estratégia de Horizonte Recuável (MPC) de C1 a C4. ....	111
Tabela 7. 4- Parâmetros de Navegação com Estratégia de Horizonte Recuável (MPC) de C5 a C8 .....	115

## Lista de Abreviaturas e Siglas

### *Letras Latinas*

$A$ -matriz do sistema

$\bar{A}$ -matriz do sistema de estimação

$B$ - matriz de controle do sistema

$\bar{B}$ - matriz de controle do sistema de estimação

$E$ - matriz essencial

$H$ -matriz de observação de estado

$K$ -ganho de Kalman

$N_d$ - número de voltas da roda direita lido na odometria

$N_e$  - número de voltas da roda esquerda lido na odometria

$N_{res}$ - resolução do sensor de odometria

$P(t)$ - solução da equação de Riccati no controle



$\bar{P}(t)$  - solução da equação de Riccati no filtro de Kalman

$Q$ - matriz de ponderação no controle

$R$ - matriz de ponderação no controle

$T$ . – matriz de transformação dos modelos de robô

$V$ - matriz de covariância do erro de visão estereoscópica

$W$ -matriz de covariância do erro de odometria

$Z$ - Vetor de coordenadas estimadas pela câmera

$b$ - distância entre as rodas do robô

$e$ - vetor de coordenadas em desvios

$\hat{e}$ - coordenada de erro de estimação

$r$ -raio da trajetória de referência [m]

$r_d$ - raio da roda direita [m]

$r_d$ - raio da roda esquerda [m]

$h(t,e)$ -vetor de partes não lineares no sistema controlado por controle por realimentação de estado linear ótimo

$f$ -distância focal da câmera

$t$ - tempo

$u$ - vetor de controle

$v_d$ - velocidade da roda direita do robô [m/s]

$v_e$ - velocidade da roda esquerda do robô [m/s]

$v_x$ - componente em $x$ da velocidade linear	[m/s]
$v_y$ - componente em $y$ da velocidade linear	[m/s]
$v_y$ - velocidade de referência	[m/s]
$\bar{v}_d$ - erro de velocidade da roda direita	[m/s]
$\bar{v}_e$ - erro de velocidade da roda esquerda	[m/s]
$\bar{v}_k$ - erro de velocidade linear na odometria	[m/s]
$w$ - erro de medição de estado pela odometria	
$x$ - coordenada da trajetória do robô	
$y$ - coordenada da trajetória do robô	
$\tilde{x}$ - coordenada da trajetória de referência	
$\tilde{y}$ - coordenada da trajetória de referência	
$\hat{x}$ -coordenada estimada	
$\hat{x}$ - coordenada estimada a priori	
$z$ - vetor de coordenadas transformadas do modelo de Brockett	

.....

### ***Letras Gregas***

$\beta$ - ângulo de estimativa de coordenada pela visão

$\gamma$  erro associado a medição de estado por câmeras

$\sigma$  desvio padrão do erro de odometria

$\theta$  coordenada angular do robô

$\omega$  velocidade angular do robô [rad/s]

$\omega_r$  velocidade angular de referência [rad/s]

.....

### ***Abreviações***

HJB – Equação de Hamilton Jacobi- Bellman

**SDRE** – State Dependent Riccati Equation

.....

## SUMÁRIO

Capítulo 1 - Introdução e Revisão Bibliográfica	1
1.1. Motivação em Controle de Robôs Móveis .....	1
1.2. Pontos Importantes em Controle Ótimo com realimentação de estados .....	3
1.3. Navegação Autônoma do Robô Através de Visão Estereoscópica e Odometria.....	6
1.4. Objetivos do Presente Trabalho.....	7
1.5. Estrutura da Tese .....	8
Capítulo 2 - Modelo de Robô Móvel	10
2.1. Equações Cinemáticas do Robô .....	11
2.2. O Modelo de Brockett para Robô Móvel .....	13
2.3. Um Modelo de Alternativo de Robô .....	14
Capítulo 3 - Medição de estado através de Odometria e Visão de duas Câmeras	16
3.1. Navegação por Odometria .....	16
3.2. Medição de posição do robô móvel através de duas câmeras.....	20
Capítulo 4- Controle Ótimo Baseado na Programação Dinâmica	29
4.1. Idéia básica de Programação dinâmica e Princípio de Otimalidade .....	29
4.2. Controle Ótimo de Sistemas através de Programação Dinâmica.....	30
4.3. Controle Ótimo de Sistemas de tempo contínuo através da equação de Hamilton Jacobi Bellman .....	31
4.4. Regulador Linear Quadrático com a Equação HJB .....	32
4.5. Formulação Geral de um Problema de controle ótimo não-linear .....	35
Capítulo 5 - Métodos de Controle Aplicados ao Robô Móvel	37
5.1. Formulação do controle linear por realimentação de estado ótimo.....	37
5.2. Controle SDRE- <i>State Dependent Riccati Equation</i> - Equação de Riccati Dependente do Estado .....	43
5.2.1. Aplicação do Controle SDRE ao Robô Móvel .....	52

5.3. Método de Controle SDRE para Sistemas Discretos Não Lineares .....	54
5.3.1. Método SDRE discreto.....	54
5.3.2. Aplicação do Método SDRE discreto para o robô móvel .....	56
Capítulo 6 - Proposta de Aplicação do Controle Dependente de Estado em um Sistema de Navegação de Robô Móvel .....	60
6.1. Visão geral do sistema de navegação.....	61
6.2. Definição de trajetória: .....	61
6.3. Localização do Robô.....	69
6.3.1. Visão.....	69
6.3.2. Estimação de estado do robô através da odometria.....	70
6.4. Navegação do robô .....	70
6.4.1. Navegação com Filtro de Kalman.....	70
6.4.2. – Estratégias de Navegação .....	74
6.4.2.1. Estratégia Mista .....	75
6.4.2.2. Estratégia Discreta.....	77
Capítulo 7 - Simulações Computacionais .....	78
7.1. Simulações para métodos de controle .....	78
7.1.1. Simulações para o método de controle ótimo por realimentação de estado linear .....	78
7.1.2. Resultados com o método de controle <i>SDRE-State Dependent Riccati Equation</i> .....	82
7.1.3. Comparação dos dois métodos.....	86
7.1.4. Simulação para o Controle SDRE Discreto .....	89
7.1.5. Controle SDRE Contínuo no Tempo .....	91
7.1.6. Simulação de União de Trajetórias Reta e Circulo .....	101
7.2. Simulações de Estimação de Estado Através da Odometria e da Visão .....	102
7.2.1. Simulação de Observação de Estado pela Visão.....	102
7.2.2. Simulação de Estimação de Estado pela Odometria .....	104
7.3. - Simulações de Navegação do Robô através do controle e filtro de Kalman.....	106
7.3.1. Estratégia de navegação discreta no tempo .....	106
7.3.2. Estratégia de navegação de horizonte recuável (MPC).....	111
7.4. Discussão dos Resultados .....	118
Conclusões e Trabalho Futuro .....	122
Referencias Bibliográficas .....	124
Anexo A : Reconstrução de Estado Linear Ótima através do Filtro de Kalman .....	133
Anexos B: Publicações em Anais de Congressos .....	142

# **Capítulo 1**

## **Introdução e Revisão Bibliográfica**

Este é o capítulo introdutório que tem o objetivo de expor as idéias iniciais sobre o controle de robôs móveis contidas na literatura. O primeiro tópico apresenta as motivações para o trabalho em controle de robôs móveis. Em seguida, são apresentados alguns pontos importantes sobre o controle ótimo através de realimentação de estados e navegação através de odometria e visão estereoscópica. Além disso, nesse capítulo, são apresentados os objetivos do presente trabalho e, por fim, a estrutura da tese.

### **1.1. Motivação em Controle de Robôs Móveis**

Robôs móveis com rodas, assim como os satélites, os submarinos e outros são sistemas robóticos especiais que cada vez mais são motivo de pesquisas por pertencerem a uma classe de sistemas mecânicos denominada de sistemas não-holonômicos. Os sistemas não-holonômicos são caracterizados por restrições cinemáticas, dificultando assim a utilização de métodos de controle clássicos, bem-estudados e tradicionalmente empregados para a solução de problemas ligados a sistemas robóticos. Ao contrário dos manipuladores mecânicos que são sistemas holonômicos e podem ser estabilizados para uma configuração desejável através de leis suaves de controle com realimentação de estados, os sistemas robóticos móveis são sujeitos a restrições não-holônicas o que dificulta o projeto de um controlador por realimentação de estado único, com leis suaves de estabilização, para atingir uma postura desejada , ou até mesmo uma trajetória de referência (Dixon et al. 2000).

Esse é um problema exaustivamente mencionado em controle de sistemas não-holonômicos, conhecido como condição de suficiência de Brockett (1983). Tal condição afirma que os sistemas não-holonômicos não podem ser estabilizados para uma configuração de repouso através de leis suaves de controle com realimentação de estados. Isto significa que, embora a propriedade de controlabilidade indique que qualquer configuração pode ser obtida em um tempo finito a partir de qualquer outra configuração, aplicando entradas de controle adequadas, nenhuma lei de controle contínua pode assegurar convergência assintótica do veículo para a configuração em repouso.

Outra abordagem é o controle híbrido proposto para esse tipo de sistema, nas referencias, de Aguiar e Pascoal (2000) e Hespanha (1996). Uma visão geral sobre os avanços em controle de sistemas não-holonômicos pode ser encontrada em Kolmanovski e McClamroch (1995). Já em Dixon et al. (2000) é apresentada uma abordagem global para a estabilização exponencial de robôs móveis a um dado conjunto de objetivos desejado. Em Walsh et al.(1994) um controle estabilizador local exponencial é proposto. Em Canudas de Wit e Sordalen (1992) é apresentado um método de linearização por realimentação de estados dinâmica para robôs móveis. Leis de controle de convergência global são propostas em Jiang e Nijmeijer (1999), Jiang (2000), Qu (2006). Bloch e Drakunov (1996) que utilizam o método de controle sliding modes para controlar o sistema integrador de Brockett. Outra técnica de controle que foi aplicada a robô móveis é o backstepping (Jiang e Nijmeijer, 1997), cujo principal problema é dificuldade de aplicação desse método para controle on line.

Outro problema não menos importante que o mencionado acima que, no entanto, recebe menos atenção acadêmica é o de realização de um controlador que leve um robô móvel a uma trajetória de referência, conhecido como *tracking*. Esse problema consiste em determinar uma postura, ou trajetória de referência a qual o robô deverá ser estabilizado através da minimização do erro entre essa referência e a trajetória atual do sistema. Para solução desse problema um mecanismo de controle bastante adequado é o por realimentação de estado, ou realimentação de estados, pelo qual, sistemas mecânicos, elétricos, ou biológicos, mantém seu equilíbrio. Aplica-se o controle, através de sinais de

diferença, determinados pela comparação dos valores atuais dos estados do sistema com os valores desejados, de forma a corrigir o desvio entre a saída e o sinal de referência.

No contexto desse problema encontramos os trabalhos de Godhavn e Egeland (1997) e Samson (1995). Outra abordagem é a utilização da política de horizonte recuável (receding horizon), também conhecido como Nonlinear Model Predictive Control-NMPC. Essa abordagem pode ser encontrada em Klancar (2007) que propõe controladores contínuos dependentes do tempo para o controle do robô não holonômico.

Ainda em direção a solução desse problema, um método de controle se destaca: linearização por realimentação de estados (state feedback linearization), que pode ser estático ou dinâmico. Essa abordagem se tornou popular nos anos de 1990 como uma ferramenta adequada para o problema por permitir o controle através da linearização de um sistema não-linear. Quando se obtém a linearização total e desacoplada do sistema não-linear consegue-se que o erro convirja a zero. Isso é obtido no método de linearização por realimentação estática (Isidori (1989), Neijmejer(1990), Slotine (1991)). Mas isso não se aplica a sistemas não-holonomicos (Thuilot (1996)). O método de linearização Dinâmico (Pomet (1995), Thuilot(1996), D´Andrea-Novet e Aranda (1994)) proporciona linearização completa para uma grande classe de robôs novéis desde que a velocidade linear seja diferente de zero. Apesar disso, o método não exibe resultados absolutamente satisfatórios no caso do robô móvel não-holonômico. Por ser um sistema subatuado, a coordenada de posicionamento angular pode não ser estabilizada completamente. Introduzem-se, então, compensadores dinâmicos que não funcionam quando a velocidade linear do robô é nula. Em Oriolo et al. (2002) foi proposto um controlador do robô móvel de rodas com acionamento diferencial, baseado no método de linearização por realimentação de estados dinâmica que lida com essa problemática.

## **1.2. Pontos Importantes em Controle Ótimo com realimentação de estados**

Para sistemas dinâmicos em geral, destacam-se três abordagens mais comuns para formulação do controle por realimentação de estado: (Khalil, 1996): estabilização,



rastreamento (*tracking*), e rejeição ou atenuação de distúrbios. No caso da estabilização, procuram-se leis de realimentação (variáveis ou invariantes no tempo) que estabilizem um sistema para um determinado regime desejado. Para o rastreamento, a meta básica é projetar o controle de forma que a saída controlada siga um sinal de referência. Para sistemas não-lineares, existem vários métodos de controle com realimentação, baseados na teoria de funções de Lyapunov. Embora o método direto de Lyapunov originalmente surgisse para o estudo de estabilidade de sistemas, atualmente, é usado para a resolução de problemas em controle de sistemas não-lineares. Existem vários trabalhos que exploram funções de Lyapunov para encontrar estratégias de controle que estabilizem sistemas conforme um critério considerado (por exemplo, Sontag (1989); Freeman e Primbs (1996)).

Desde as primeiras publicações na década 1960 (Krasovskii (1959); Kalman (1960), Letov (1961)), funções de Lyapunov são utilizadas para estudo de problemas do controle ótimo. Sabe-se que um problema do controle ótimo pode ser reduzido à resolução da equação diferencial parcial de Hamilton-Jacobi-Bellman. Num caso geral, existem muitas dificuldades na sua resolução. Em caso particular, para um sistema linear, com controle ótimo minimizando um funcional quadrático, a função quadrática de Lyapunov é a solução da equação de Hamilton-Jacobi-Bellman. Essa abordagem da função de Lyapunov de um sistema não-linear pode ser a solução da equação de Hamilton-Jacobi-Bellman tornou-se popular. Em (Bernstein, 1993) esta idéia foi apresentada de uma maneira construtiva para sistemas não-lineares de tempo contínuo com controle não-linear por realimentação de estado minimizando um funcional não-quadrático. A solução da equação de Hamilton-Jacobi-Bellman através da função de Lyapunov garante tanto a estabilidade como a otimalidade da solução. Em Haddad et al. (1998) este enfoque foi aplicado à resolução de um problema de controle não-linear robusto. O trabalho de Haddad e Chellaboina (1996) utilizou a mesma abordagem para sistemas não-lineares discretos no tempo.

De fato, não existem técnicas sistemáticas para obter funções de Lyapunov para um caso geral em sistemas não-lineares, mas o enfoque em consideração pode ser aplicado para aqueles sistemas não-lineares cuja função de Lyapunov é conhecida. Como exemplo, em Rafikov e Bathazar, 2004 foi proposta a função quadrática para resolver o problema do controle ótimo não-linear do sistema de Rössler e em Rafikov e Balthazar (2005) foi

proposto o controle por realimentação de estado linear que minimiza o funcional não-quadrático para sistemas não-lineares.

Uma técnica de controle ótimo, bastante promissora, que na última década ganha força é chamada controle por equação de Riccati Dependente de Estado (State Dependent Riccati Equation SDRE). Ela é similar ao tradicional Regulador Linear Quadrático- LQR, contudo é um método não-linear onde o controle é obtido da solução da equação de Riccati na forma dependente de estado. No entanto, diferentemente de métodos de controle não-lineares, o SDRE é de elementar implementação e altamente adequado a implementação on line . A idéia de utilização de controle por realimentação de estado baseado em SDRE para sistemas não lineares remonta a 1962 por Pearson. Nessa referência, a estabilização do sistema ao equilíbrio e a resolução da equação diferencial de Riccati dependente do tempo é obtida analiticamente para um exemplo de baixa ordem. Mais tarde a idéia foi expandida por Wernli & Cook (1975). Porém, a maior parte do desenvolvimento dessa estratégia aconteceu na década de 90 e na última década, com os trabalhos de Banks em 1992, Mracek & Cloutier (1998), Cloutier, et al. 1998, Cloutier, Stansbery & Sznajder (1999), que derivaram a condição necessária para existência de qualquer matriz de ganho de um controlador SDRE. Mracek e Cloutier (1998) provaram estabilidade assintótica local para um caso de sistema de ordem  $n > 1$  com um controlador SDRE por realimentação de estado.

O método SDRE tem demonstrado sua eficácia em diversos sistemas: design de autopiloto (Mracek e Cloutier, 1996), desenvolvimento de leis de guiamento avançadas (Cloutier e Zipfel, 1999), controle de satélites e espaçonaves (Hammet et al. 1998), robótica (Erdem e Alleyne, 2001), problema de benchmark (Mracek e Cloutier, 1998), controle de ventilação de dutos (Yu et al. (2001)), controle de veículo autônomo subaquático (Naik e Singh, 2007), sincronização de sistemas caóticos (Jayaram e Tadi, 2006). Uma visão geral dos avanços em técnica SDRE pode ser encontrada em (Çimen, 2008). Entre suas vantagens estão a não-necessidade de linearização do sistema, possibilidade de otimização de um critério, simples e imediata implementação dos algoritmos, e aplicação do controle “on line”.

### **1.3. Navegação Autônoma do Robô Através de Visão Estereoscópica e Odometria**

Navegação autônoma é uma tarefa de grande desafio para um robô móvel. O sucesso nessa tarefa depende de quatro competências essenciais: percepção, localização, cognição e controle de movimento. A percepção diz respeito ao sensoramento do ambiente e interpretação desses dados. Um robô móvel autônomo é equipado com sensores de posição que podem ser do tipo encoders, sonar, laser, radar, monovisão, visão estéreo. Estes sensores promovem a percepção do robô do meio ambiente parcial ou totalmente desconhecido provendo informação a respeito da localização espacial deste que será utilizada para a tomada de decisão autônoma na navegação. Ruídos induzem limitações à consistência de leitura dos sensores. Outro problema dos sensores é a ambigüidade resultante de informações semelhantes sobre locais ou objetos de naturezas diferentes o que dificulta realização de tarefas, a cognição e sua localização.

A localização responde a pergunta: “onde estou?” Ou seja, tem a função de proporcionar o cálculo da coordenada absoluta ou relativa a uma referência de navegação do robô no ambiente. Isso envolve fusão de informação de múltiplos sensores e estimação de posição a partir de estratégias. O problema de localização depende diretamente da qualidade da informação proveniente dos sensores, por isso é completamente limitada pelos erros, imprecisões e ruídos. Existem estratégias de navegação baseadas na localização em relação a um referencial apenas, outras no mapeamento do ambiente, envolvendo tanto localização quanto mapeamento do ambiente. Encontra-se também uma abordagem behaviorista que basicamente defende uma solução para a navegação baseada em um conjunto de comportamentos programados de reações diante de situações que o robô encontra. Por exemplo, virar à esquerda ou à direita diante de um obstáculo, seguir sempre pelo lado esquerdo da parede etc. Essa abordagem é interessante por dispensar os métodos de localização e incertezas relacionadas à medição de posição pelos sensores. No entanto, só é válida para alguns ambientes específicos.

A solução de navegação , seja pela localização, seja pelo mapeamento é diferente para cada aplicação. Cada sensor possui limitações e indicações. Para ambientes externos e de longa navegação os métodos mais indicados são navegação GPS, visão estereoscópica,

lazers. Para ambientes internos com circulação de pessoas e espaços pequenos, a odometria, a visão, sonares e navegações inerciais são mais indicados.

O método clássico de localização por odometria envolve dados de encoders em cada roda proporcionando informações do deslocamento de cada roda. Através de um modelo é prevista uma posição de localização do robô. Uma referencia com síntese interessante desse método encontra-se em Borenstein et al. (1993).

Outra abordagem para a percepção, mas não somente, é a visão estereoscópica que é o estado da arte em percepção e localização. Métodos de monovisão já são comuns em navegação de robôs.

Um método clássico, porém poderoso, que pode ser aplicado à localização é o filtro de Kalman (1960), pois proporciona a fusão de informação de sensores diferentes e estimação ótima do estado do robô. A abordagem probabilística atribui uma função de probabilidade ao estado estimado do robô e aos estados provenientes de sensores. A introdução sobre filtro de Kalman pode ser encontrada em Maybeck em Cox e Wilfong (1990). e um tratamento mais detalhado em Kwakernaak e Sivan (1972); Bar-Shalom (1993).

#### **1.4. Objetivos do Presente Trabalho**

O objetivo geral desse trabalho é controlar um robô móvel de rodas de forma eficaz, simples, em tempo real, minimizando algum critério de custo e fazê-lo navegar em um ambiente desconhecido através de informações de encoders e visão de duas câmeras.

Para esse propósito estuda-se, primeiramente, o modelo do robô móvel de duas rodas com acionamento diferencial. Propõe-se uma estratégia de controle ótimo linear com realimentação de estados, baseada no enfoque de Lyapunov para o sistema não-linear do robô móvel de acionamento diferencial. Aplica-se a abordagem de controle não-linear SDRE, com enfoque em funções de Lyapunov que resulta um controlador subótimo para o modelo não linear do robô móvel de acionamento diferencial. Ambos os controles lidam com o problema de regulação a uma trajetória de referência em uma aplicação online. Os resultados desse desenvolvimento foram publicados em: 6ª Conferencia em Dinâmica

Controle e Aplicações-DINCON 2007, 9th Biennial ASME Conference on Engineering Systems Design and Analysis- ESDA e 3rd IEEE Multi-Conference on Systems and Control- MSC 2009. Estes trabalhos são encontrados na seção de anexos no final da presente tese.

Estuda-se ainda, a integração do robô com o ambiente proporcionando a navegação através de sensores de odometria (*encoders*) e mecanismos da visão estereoscópica e processamento de imagens, desenvolvidas no âmbito do projeto PILGRIM, do grupo de pesquisa da UNICAMP – FEM/DPM. Por fim, verificam-se as estratégias de controle e navegação propostas para o robô, através de simulações.

## **1.5. Estrutura da Tese**

No capítulo 2 são apresentados os modelos de robô móvel relevantes para esse trabalho. Na primeira seção desse capítulo é apresentado o modelo cinemático do robô móvel de rodas. Na segunda seção do capítulo 2 disserta-se sobre o integrador de Brockett, sistema ao qual as equações cinemáticas do robô podem ser reduzidas. Na terceira seção do capítulo apresenta-se um modelo do robô móvel em coordenadas de desvios que é utilizado no decorrer de toda a tese como o modelo de robô móvel a ser controlado.

O assunto de localização do robô é abordado no capítulo 3 no qual estão descritos os métodos de medição de estado através da odometria e estimação de posição com uso de visão de duas câmeras.

O capítulo quatro é dedicado a uma introdução teórica sobre controle ótimo de sistemas através do princípio de Bellman. A primeira seção apresenta as idéias básicas de programação linear e princípio de otimalidade de Bellman. A segunda e a terceira seção descrevem o controle de sistemas contínuos com programação dinâmica. A quarta seção apresenta a idéia do regulador linear quadrático com a equação de HJB. Por fim, na última seção é formulado o problema geral de controle ótimo não-linear de sistemas, utilizando a equação de HJB.

O Capítulo 5 apresenta os métodos de controle propostos e aplicados nesse trabalho para o controle de trajetória do robô móvel. Na primeira seção tem-se a formulação do controle ótimo por realimentação de estado linear. Na segunda seção está descrito o método SDRE. Já na terceira seção é proposto um método de controle dependente de estados discreto no tempo e aplicado para controle de trajetória do robô móvel.

No capítulo 6 é proposta uma aplicação para a união de métodos de medição de estado, tratamento de erros e controle de trajetória do robô, proporcionando a sua navegação no ambiente. Duas estratégias de navegação são propostas: uma discreta no tempo, que utiliza o modelo de robô e controle discreto, e, uma estratégia mista de navegação na qual as medições de estado são realizadas de forma discreta no tempo e o controle de trajetória em intervalos contínuos no tempo entre cada medição.

Os resultados obtidos nesse trabalho são apresentados no capítulo 7, dividido em três partes principais. A primeira apresenta simulações numéricas dos métodos de controle propostos no capítulo 6 para diferentes parâmetros e trajetórias de referência. A segunda parte do capítulo apresenta simulações para métodos de medição de estado odometria e visão estereoscópica para verificação de erros envolvidos. A terceira parte contém simulações das duas estratégias de navegação descritas no capítulo 7 que integram a estimação de estado e controle de trajetória do robô móvel.

O anexo A contém tópicos sobre reconstrução de estado através do filtro de Kalman. Tais como o problema do observador de estado, observador de estado ótimo com abordagem estocástica e o filtro de Kalman discreto no tempo.

## Capítulo 2

### Modelo de Robô Móvel

Um dos modelos mais simples e comumente utilizados para representar um robô móvel de rodas, é o modelo de direção diferencial (*differential steering*). Considera-se que o veículo móvel possui duas rodas idênticas, paralelas, não deformáveis, montadas num mesmo eixo e que são controladas independentemente por motores. Isto significa que quando as rodas giram com a mesma velocidade o robô segue uma trajetória reta e quando as rodas giram com velocidades diferentes o robô faz uma curva. Pode haver ainda uma roda frontal ou traseira para garantir o apoio com o piso e promover a estabilidade do robô. Assume-se que o plano de cada roda é perpendicular ao solo e esta pode rolar apenas no eixo de montagem. Este fato, juntamente com o fato de haver apenas rolamento das rodas sem deslizamento no solo, faz com que haja restrição ao conjunto de vetores velocidade inicial do sistema, ou seja, não é possível, em um instante inicial, promover um movimento do sistema em qualquer direção, caracterizando esse sistema como não-holonômico. As características de movimento, descritas para o robô, permitem a sua classificação na categoria de não-holonômicos.

A figura 2.1 representa o esquema de vista do topo de um robô de duas rodas. O movimento do robô se dá no plano  $xy$  sendo essas as suas coordenadas de posição e a direção do robô é dada pela coordenada de rotação  $\theta$ .

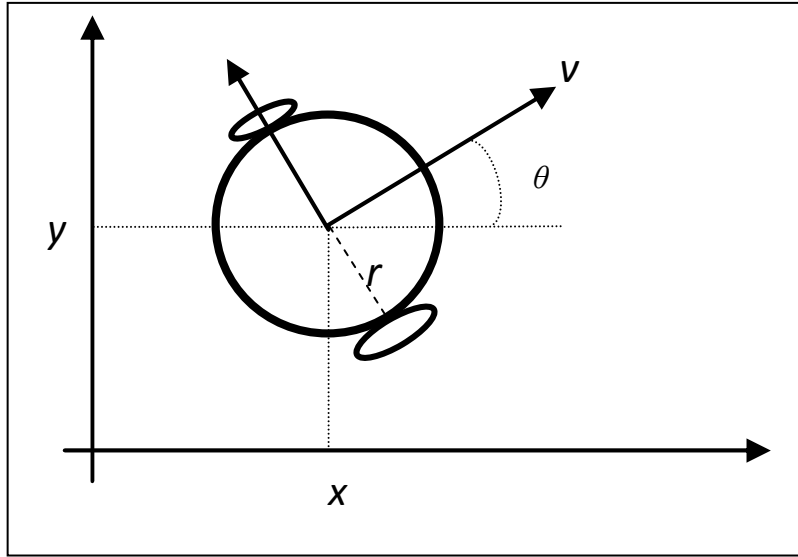


Figura 2. 1. Modelo Cinemático de Robô Móvel

### 2.1. Equações Cinemáticas do Robô

Para descrever o deslocamento nas coordenadas  $x$  e  $y$  considera-se que o movimento depende da velocidade linear do ponto central do eixo das rodas, ou seja, o ponto médio. E a velocidade do robô nesse ponto é dada por:

$$V = \frac{(V_d + V_e)}{2}, \quad (2.1)$$

onde  $V_d$  e  $V_e$  são a velocidades lineares da roda direita e esquerda do robô respectivamente.

A variação no tempo das coordenadas  $x$  e  $y$  pode ser descrita pela velocidade linear  $V$  projetada nas direções  $x$  e  $y$ , respectivamente, resultando nas seguintes equações diferenciais:

$$\begin{aligned} \frac{dx}{dt} &= \frac{(V_d + V_e)}{2} \cos(\theta) \\ \frac{dy}{dt} &= \frac{(V_d + V_e)}{2} \sin(\theta) \end{aligned} \quad (2.2)$$



Para descrever a variação da coordenada angular  $\theta$  considera-se um referencial fixado no ponto central roda esquerda do robô, ponto que faz contato com o solo. A roda direita descreve um arco com o raio que corresponde a distancia entre as rodas, como mostrado na figura 2.2. O robô em si é tratado como um corpo rígido, e cada ponto seu submete-se as mesmas mudanças de direção. A velocidade angular da roda é a variação do ângulo  $\alpha$  no tempo e é dada por:

$$\omega_{roda} = V/r , \quad (2.3)$$

onde  $V$  é a velocidade linear e  $r$  é o raio do movimento. Da mesma forma, a variação geral do ângulo  $\theta$  no tempo de toda plataforma é:

$$\frac{d\theta}{dt} = \omega = \frac{(V_d - V_e)}{2r} \quad (2.4)$$

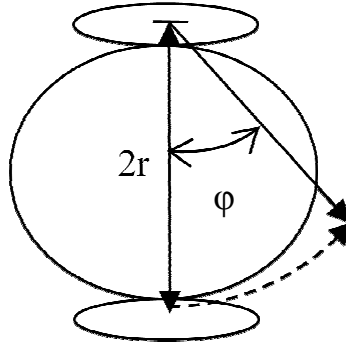


Figura 2. 2 Variação da Coordenada  $\theta$  no Robô

As equações cinemáticas do robô podem ser reescritas mais sinteticamente como:

$$\begin{aligned} \dot{\theta} &= \omega \\ \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \end{aligned} , \quad (2.5)$$

onde  $x$  e  $y$  denotam a posição do centro do eixo das rodas e  $\theta$  é a orientação do robô. A velocidade linear do robô é  $v$  e a velocidade angular é  $\omega$ . Este modelo também é chamado de modelo de robô uniciclo pois reduz a cinemática diferencial das rodas descrevendo-a

como cinemática de uma única roda central de controle. Tal modelo é encontrado comumente na literatura em Canudas de Wit e Sordalen (1992), Hespanha (1996), Kolmanovski e McClamroch (1995), Oriolo et. al. (2002), Alekander e Maddocks (1989).

## 2.2. O Modelo de Brockett para Robô Móvel

As equações cinemáticas do robô móvel (2.5) são submetidas a uma transformação que os leva a um sistema equivalente descrito na literatura e conhecido como integrador de Brockett (1983). A transformação visa facilitar o projeto de um controlador, já que propicia o fato do sistema poder ser escrito na forma linear matricial:

$$\dot{x} = Ax + Bu \quad (2.6)$$

O sistema (3.5), portanto, é submetido a uma transformação  $T$  que resulta em um novo sistema nas coordenadas transformadas  $z(t) = [z_1(t) \ z_2(t) \ z_3(t)]^T \in \mathbb{R}^3$ :

$$z = T \begin{bmatrix} \theta \\ x \\ y \end{bmatrix}, \quad (2.7)$$

onde  $T \in \mathbb{R}^{3 \times 3}$  é a matriz de transformação definida por:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & 2\sin(\theta) - \theta\cos(\theta) & -2\cos(\theta) - \theta\sin(\theta) \end{bmatrix} \quad (2.8)$$

Como resultado tem-se:

$$\begin{aligned} z_1 &= \theta \\ z_2 &= x\cos(\theta) + y\sin(\theta) \\ z_3 &= -2(x\sin(\theta) - y\cos(\theta)) + \theta(x\cos(\theta) + y\sin(\theta)) \end{aligned} \quad (2.9)$$

Derivando  $z$  obtém-se o sistema

$$\begin{aligned} \dot{z}_1 &= U_1 \\ \dot{z}_2 &= U_2 \\ \dot{z}_3 &= z_2 U_1 - z_1 U_2 \end{aligned} \quad , \quad (2.10)$$

onde:

$$\begin{aligned} U_1 &= \omega \\ U_2 &= v - \omega(x \sin \theta - y \cos \theta) \end{aligned} \quad (2.11)$$

O modelo (2.10) é o integrador de Brockett, um modelo simples com restrições não-holonômicas.  $U \in \mathbb{R}^2$  é o vetor de controle e  $\omega$  e  $v$  são as velocidades angular e tangencial, respectivamente.

Um dos objetivos do controle da trajetória de um robô móvel é levá-lo a uma trajetória de referência minimizando o erro de desvio dessa trajetória. Esse problema de controle é também chamado de *tracking* a uma trajetória. Para realizar esse tipo de controle em um robô móvel um controlador linear por realimentação de estado ótimo foi proposto no capítulo 5.

### 2.3. Um Modelo de Alternativo de Robô

Esse modelo pode ser encontrado na literatura em Kanayama *et al.* (1990) e Klancar (2007). Ele é utilizado no presente trabalho para o controle através de realimentação de estados do robô como uma alternativa ao modelo de Brockett. Nesse caso, um sistema em desvios é obtido diretamente do modelo (2.5) sem passar por transformações que o levem primeiramente ao modelo de Brockett. Isso facilita a implementação do controle do sistema e, ao mesmo tempo, auxilia a interpretação de resultados.

Considera-se um robô virtual que segue idealmente a trajetória de referência :

$$\begin{aligned} \dot{\tilde{x}} &= v_r \cos \tilde{\theta} \\ \dot{\tilde{y}} &= v_r \sin \tilde{\theta} \\ \dot{\tilde{\theta}} &= \omega_r \end{aligned} \quad , \quad (2.12)$$

onde as velocidades de referência angular e linear são, respectivamente,  $\omega_r$  e  $v_r$ , e contêm a informação essencial sobre a trajetória de referência. A forma do robô de referência é

semelhante a do próprio modelo cinemático com variáveis de estado de referência que são dependentes do tipo de trajetória de referência.

A diferença entre a trajetória do robô real e o robô de referência é considerada um erro ou desvio ( que deve ser minizado pelo controle) dada por:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} - x \\ \tilde{y} - y \\ \tilde{\theta} - \theta \end{bmatrix}, \quad (2.13)$$

onde  $e = (e_1 \ e_2 \ e_3)^T$  é o vetor de coordenadas em desvio,  $(x \ y \ \theta)^T$  e  $(\tilde{x} \ \tilde{y} \ \tilde{\theta})^T$  são , respectivamente, os vetores de estado do robô e do estado de referência do robô.

No problema de *tracking* por realimentação de estados do robô móvel, esse modelo representa não diretamente o problema de estabilização a uma trajetória de referência, mas sim a um robô de referência. Este, por sua vez, é influenciado pela trajetória de referência através das velocidades desejadas: angular e linear. Isso pode ser tornar vantajoso em termos de controle.

Em resumo, esse capítulo apresenta o modelamento do robô móvel considerado. Primeiramente mostrando as equações cinemáticas do robô e em seguida mostrando dois modelos que representam o sistema a ser controlado na forma de coordenadas de desvios. Dessa maneira, torna-se possível controlar sistema do robô não-holonômico.

## Capítulo 3

### Medição de estado através de Odometria e Visão de duas Câmeras

Nesse capítulo são apresentados os métodos de localização e medição de estado utilizados para realizar a navegação do robô que são a odometria e a visão de duas câmeras. Na primeira seção o processo de localização odométrico é descrito bem como a quantificação de erros sistemáticos. Na segunda seção é mostrado o cálculo de posição com utilização de duas câmeras através de dois pontos de referência.

#### 3.1. Navegação por Odometria

A determinação do estado de um corpo navegando no espaço, por um processo odométrico, remonta a Grécia Antiga. A palavra odometria vem do grego *hodos* (viagem) e *metron* (medida) significando literalmente medida do caminho percorrido. Odometria é o processo de obtenção da variação do estado no tempo através de informações do movimento dos atuadores das rodas. Esse processo, em geral, requer um sensor ou odômetro que capta o movimento das rodas e traduz em distância percorrida. Com base nisso, um modelo com as equações de odometria estima a variação do estado do sistema no tempo. No que diz respeito à navegação do robô móvel, essa técnica foi amplamente utilizada e é conhecida como *dead-reckoning* (ou na tradução livre: navegação às cegas), pois possibilita a navegação do robô sem uma visão ativa.

Considerando-se o modelo cinemático do robô:

$$\frac{dx}{dt} = v \cdot \cos(\theta)$$

$$\frac{dy}{dt} = v \cdot \sin(\theta)$$

$$\frac{d\theta}{dt} = \omega \quad (3.1)$$

Onde

$$v = \frac{(V_d + V_e)}{2} \text{ e } \omega = \frac{(V_d - V_e)}{2r}. \quad (3.2)$$

O sistema (3.1) pode ser discretizado através do método de Euler levando a :

$$x(t + \Delta t) = x(t) + v \cdot \Delta t \cdot \cos(\theta),$$

$$y(t + \Delta t) = y(t) + v \cdot \Delta t \cdot \sin(\theta),$$

$$\theta(t + \Delta t) = \theta(t) + \omega \cdot \Delta t \quad . \quad (3.3)$$

As velocidades no deslocamento infinitesimal são dadas por:

$$v_d = \frac{dl_d}{dt} = \frac{\Delta l_d}{\Delta t},$$

$$v_e = \frac{dl_e}{dt} = \frac{\Delta l_e}{\Delta t} . \quad (3.4)$$

Considerando que os encoders utilizados possuem uma resolução  $N_{res}$  e que as leituras do número de passos nas rodas direita e esquerda respectivamente são dadas por  $N_d$  e  $N_e$ , então os deslocamentos são dados por :

$$\Delta l_d = 2\pi r_d \frac{N_d}{N_{res}}$$

$$\Delta l_e = 2\pi r_e \frac{N_e}{N_{res}} \quad . \quad (3.5)$$

As velocidades da roda esquerda e direita têm a forma:

$$\begin{aligned} v_d &= \frac{2\pi r_d}{\Delta t} \frac{N_d}{N_{res}} \\ v_e &= \frac{2\pi r_e}{\Delta t} \frac{N_e}{N_{res}} \end{aligned} \quad (3.5)$$

Às coordenadas (3.3) do robô do momento  $t_k$  ao  $t_{k+1}$  são acrescidas as coordenadas anteriores através das seguintes equações, conhecidas as leituras dos *encoders*:

$$\begin{aligned} x(t + \Delta t) &= x(t) + (N_d \cdot r_d + N_e r_e) \frac{\pi}{N_{res}} \cdot \cos(\theta) \\ y(t + \Delta t) &= y(t) + (N_d \cdot r_d + N_e r_e) \frac{\pi}{N_{res}} \cdot \sin(\theta) \\ \theta(t + \Delta t) &= \theta(t) + (N_d \cdot r_d - N_e r_e) \frac{2\pi}{b \cdot N_{res}} \end{aligned} \quad (3.7)$$

Considerado o incremento de deslocamento do robô, equação (3.5), pode-se reescrever o modelo (3.7) substituindo as velocidades em (3.2). Além disso,  $\Delta t$  é considerado um número inteiro. O modelo na forma matricial é :

$$x_{k+1} = \bar{A}x_k + \bar{B}u_k \quad (3.8)$$

$$\text{onde } \bar{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \bar{B} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \text{ e controle } u_k = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix}.$$

Na navegação através da odometria o sistema está sujeito a erros sistemáticos e não sistemáticos (Valgas e Borenstein, 1993). Os erros sistemáticos são causados por imperfeições no modelo cinemático tais como medidas incorretas dos raios das rodas ou comprimento do eixo que as separa, bem como pelo intrínseco de quantização de *encoders*. Esse erro é cumulativo durante a navegação gerando distorções na determinação do estado do robô, no entanto ele é modelável e previsível. Já os erros não sistemáticos são situações

inesperadas que surgem durante a evolução da trajetória do robô como imperfeições do terreno e deslizamento das rodas.

A navegação por odometria, considerando-se o erro sistemático, pode ser dada pelo sistema:

$$x_{k+1} = \bar{A}x_k + \bar{B}u_k + w_k, \quad (3.9)$$

onde  $w_k = \begin{bmatrix} \bar{v}_k \\ \bar{\omega}_k \end{bmatrix}$  é o erro em velocidades linear e angular do robô. O erro em velocidade  $\bar{v}_k$ , de acordo com as equações de navegação pela odometria pode ser dado como somatório do erro das duas rodas, direita e esquerda:

$$\bar{v}_k = \frac{\bar{v}_d + \bar{v}_e}{2} \quad (3.10)$$

e o erro do deslocamento angular:

$$\bar{\omega}_k = \frac{\bar{v}_d - \bar{v}_e}{b}, \quad (3.11)$$

onde  $b$  é a distancia entre as rodas: direita e esquerda.

No modelo de deslocamento pela odometria, considera-se que o erro da velocidade é fruto do deslocamento realizado pela roda direita e esquerda individualmente num intervalo de tempo:

$$\begin{aligned} \bar{v}_d &= \frac{\Delta l_d}{\Delta t}, \\ \bar{v}_e &= \frac{\Delta l_e}{\Delta t}, \end{aligned} \quad (3.12)$$

sendo que quando  $\Delta t = 1$  esse deslocamento é dado por:

$$\begin{aligned} \overline{\Delta l_d} &= 2\pi r_d \frac{\bar{N}_d}{N_{res}}, \\ \overline{\Delta l_e} &= 2\pi r_e \frac{\bar{N}_e}{N_{res}}, \end{aligned} \quad (3.13)$$



onde  $\bar{N}_d$  e  $\bar{N}_e$  representam o erro em número de passos para cada roda,  $N_{res}$  é a resolução do sensor de odometria de cada roda e  $r_d$  e  $r_e$  são os raios das rodas direita e esquerda, respectivamente, que podem ser por finalidade de simplificação modelados como sendo ambos iguais e de valor  $r$ .

Substituindo (3.13) e (3.12) em (3.10) obtém-se a expressão do erro em velocidade

$$\bar{v} = (\bar{N}_d + \bar{N}_e) \frac{\pi r}{N_{res}}. \quad (3.14)$$

Pelo processo análogo obtemos a expressão do erro para a variação angular:

$$\bar{\omega} = (\bar{N}_d - \bar{N}_e) \frac{2\pi r}{b.N_{res}}. \quad (3.15)$$

Reescrevendo obtém-se a expressão para o erro de odometria:

$$w_k = \begin{bmatrix} \bar{v}_k \\ \bar{\omega}_k \end{bmatrix} = \begin{bmatrix} (\bar{N}_d + \bar{N}_e) \frac{\pi r}{N_{res}} \\ (\bar{N}_d - \bar{N}_e) \frac{2\pi r}{b.N_{res}} \end{bmatrix}. \quad (3.16)$$

### 3.2. Medição de posição do robô móvel através de duas câmeras.

Existem diversas formas de obtenção da localização de um robô móvel em um ambiente. Entre os sensores e métodos de observação de estado estão o radar, o sonar, o encoders, os métodos de visão mono e estereoscópica, o GPS. Uma das mais promissoras é a localização através de visão robótica. A localização do robô no mundo pode ser efetuada através de um par de câmeras, fixando-se no mundo pontos de orientação com coordenadas absolutas. Considerando-se um caso mais realístico, são introduzidos erros de projeção nas telas das câmeras, fazendo com que a posição dos objetos no mundo seja diferentes do que o encontrado pelas câmeras. A informação da localização dos objetos no referencial do robô, juntamente com suas posições absolutas, resulta na informação de sua possível localização.

### 3.2.1. - Posicionamento do Robô Baseado em Dois Pontos da Imagem

A determinação da posição espacial de um robô pode ser feita a partir da projeção conhecida de pelo menos dois pontos em duas câmeras, presentes no referencial próprio do robô.

Consideram-se os referenciais usados no modelo de um robô navegando num ambiente bidimensional, como ilustrado na Figura 3.1. As referências do mundo são representadas por  $(X,Y)$ , as do robô como  $(x,y)$ , a de duas câmeras como  $a$  ( $x_{ca},y_{ca}$ ) e  $b$  ( $x_{cb},y_{cb}$ ) além de referenciais  $r$  e  $s$  de posição de imagem na câmera. As câmeras estão afastadas de uma distância  $t$  e seus referenciais encontram-se rotacionados de um ângulo  $\theta$  em relação ao mundo, assim como o referencial do robô. A rotação do robô  $\theta$  (valor do ângulo da posição real) é definida como o ângulo entre o eixo  $z$  do robô e o eixo  $X$  do mundo.

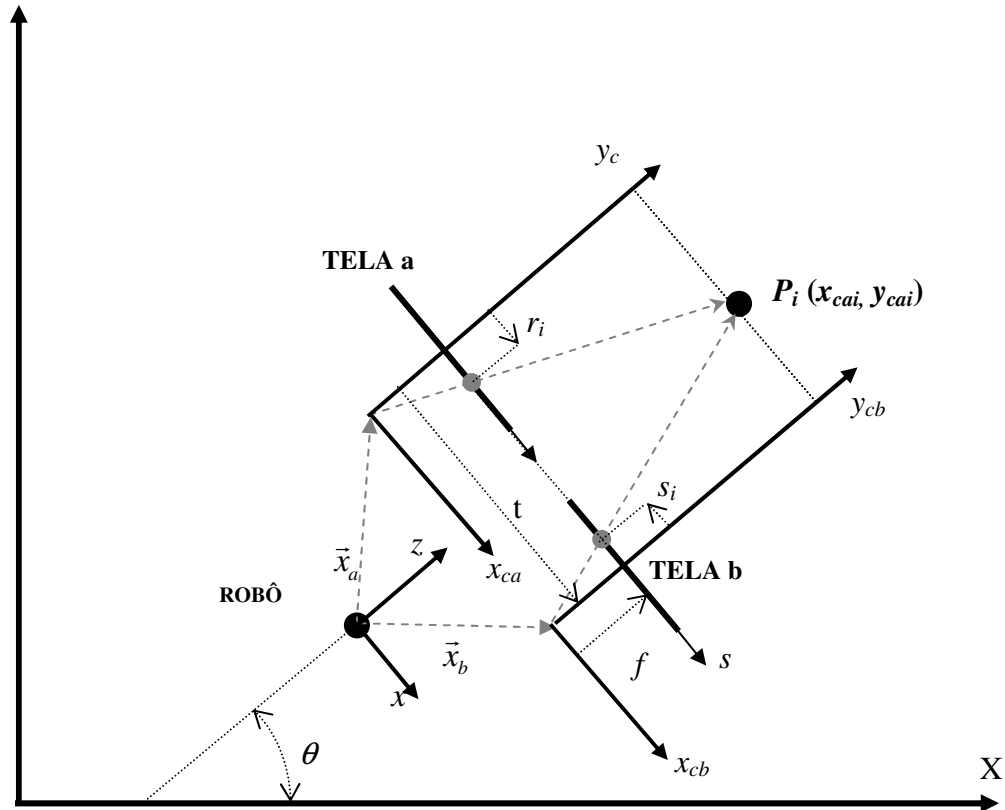


Figura 3. 1- Ângulos e Coordenadas de um Sistema de Duas Câmeras no Mundo.

O vetor de coordenadas  $(x_{cak}, y_{cak})$  descreve a posição do ponto  $P_k$  em relação aos eixos de referências da câmera  $a$ . Os elementos de  $x_{cak}$  são calculados a partir da equação abaixo:

$$x_{cak} = \begin{bmatrix} x_{cak} \\ y_{cak} \end{bmatrix} = \frac{t}{(r_k - s_k)} \begin{bmatrix} r_k \\ f \end{bmatrix} \quad (3.17)$$

A projeção e o mapeamento em uma tela linear de número de pixels limitado fazem com que a ela seja associada uma incerteza quanto à posição dos pontos, calculados pela equação (3.17), que pode ser obtida derivando-se a equação (3.17):

$$dx_{cak} = \begin{bmatrix} dx_{cak} \\ dy_{cak} \end{bmatrix}, \quad (3.18)$$

resultando em :

$$dx_{cak} = \begin{bmatrix} \frac{t}{(r_k - s_k)} dr_k - \frac{r_k t}{(r_k - s_k)^2} + \frac{r_k t}{(r_k - s_k)^2} ds_k \\ -\frac{tf}{(r_k - s_k)^2} dr_k + \frac{tf}{(r_k - s_k)^2} ds_k \end{bmatrix}. \quad (3.19)$$

Reescrevendo (3.19), temos

$$dx_{cak} = \begin{bmatrix} \frac{t(r-s)-rt}{(r_k-s_k)^2} & \frac{r_k t}{(r_k-s_k)^2} \\ -\frac{tf}{(r_k-s_k)^2} & \frac{tf}{(r_k-s_k)^2} \end{bmatrix} \begin{bmatrix} dr_k \\ ds_k \end{bmatrix} \quad (3.20)$$

Desta forma, o erro de estimação do vetor  $x_{cak}$  pode ser representado através de uma matriz  $B_k$  :

$$dx_{cak} = B_k dr_k, \quad (3.21)$$

onde a matriz  $B_k$  e o vetor  $dr_k$  são definidos como:

$$B_k = \frac{y_{cak}^2}{ft} \begin{bmatrix} -\frac{s_k}{f} & \frac{r_k}{f} \\ -1 & 1 \end{bmatrix} \quad (3.22)$$

e

$$dr_k = \begin{bmatrix} dr_k \\ ds_k \end{bmatrix}. \quad (3.23)$$

O vetor de estado  $Z$ , ou *pose* do robô, obtida por imagens, pode ser definido a partir de dois pontos fixos  $P$  e  $A$  no mundo, conforme mostrado na figura (3.19). Nessa figura também se utiliza a nomenclatura de  $Z_x, Z_y, Z_\theta$ , para indicar as coordenadas de posição do robô no mundo, bem como a sua coordenada de rotação  $Z_\theta$ .

A Figura (3.2) traz outra representação das relações geométricas existentes entre o robô e dois pontos  $P$  e  $A$  na região planar de movimento do robô. Nela encontramos os ângulos  $\alpha, \beta, \gamma$  que são necessários para a obtenção da localização espacial do robô.

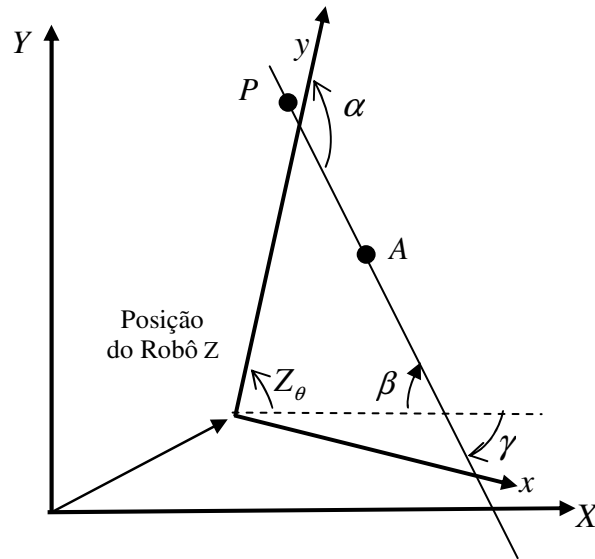


Figura 3. 2 –Cálculo de Coordenadas para Localização do Robô Através dos Pontos P e A.

A equação (3.24) mostra como a pose do robô  $Z$  está relacionada com as coordenadas dos pontos  $P$  e  $A$  no mundo e com as coordenadas desses pontos, vistos das câmeras:

$$Z = \begin{bmatrix} Z_x \\ Z_y \\ Z_\theta \end{bmatrix} = \begin{bmatrix} \frac{1}{2} [P + A - R^T (x_{caP} + x_{caA} + 2x_a)] \\ \alpha + \gamma \end{bmatrix}, \quad (3.24)$$

onde a matriz de rotação  $R$  é definida pelo ângulo  $Z_\theta$  estimado pelas câmeras:

$$R = \begin{bmatrix} \sin(Z_\theta) & -\cos(Z_\theta) \\ \cos(Z_\theta) & \sin(Z_\theta) \end{bmatrix}. \quad (3.25)$$

A posição angular  $Z_\theta$  do robô é calculada a partir dos ângulos  $\alpha$  e  $\gamma$ , como mostra a Figura 4.2. O ângulo entre a direção  $X$  e  $\overline{PA}$  é dado por  $\gamma$ . O ângulo  $\alpha$  é a inclinação entre o eixo  $y$  do robô segmento  $\overline{PA}$  dado por:

$$\alpha = \frac{\pi}{2} + \beta, \quad (3.26)$$

onde:

$$\beta = \arctan\left(-\frac{z_{rP}-z_{rA}}{x_{rP}-x_{rA}}\right) = \arctan\left(\frac{z_{rP}-z_{rA}}{x_{rA}-x_{rP}}\right), \quad (3.27)$$

com  $P_r=(x_{rP}, z_{rP})$  e  $A_r=(x_{rA}, z_{rA})$  sendo as coordenadas dos pontos  $P$  e  $A$  respectivamente no referencial do robô. Assim, pode-se reescrever:

$$P_r = x_{caP} + x_a \quad (3.28)$$

e

$$A_r = x_{caA} + x_a \quad (3.29)$$

### 3.2.2. Modelo de Erro de Estimação de Posição através de duas câmeras

Considera-se um processo de medição de estado dado por  $z_k$

$$z_{k+1} = H_k \hat{x}_k + \gamma_k \quad (3.30)$$

onde  $H_k$  é uma matriz de observação e  $\hat{x}_k$  é o estado estimado por um estimador associado (como por exemplo o filtro de Kalman). O processo de medição do estado envolve um erro  $\gamma_k$ . Essa é uma variável aleatória e é assumida com distribuição normal e possui um desvio padrão  $\bar{\delta}$ .

Esse erro é uma combinação de vários erros que ocorrem durante o processo de medição de estado através de visão estereoscópica. Entre esses está o erro sistemático de discretização pela projeção dos pontos em uma tela com número limitado de pixels.

O modelamento e a covariância desses erros considerados para a medição do estado do robô através de duas câmeras será desenvolvida a seguir. O seu uso será feito posteriormente na introdução no sistema de um estimador de estado como o filtro de Kalman.

As incertezas associadas ao vetor de estado do robô calculado pelas câmeras são dadas pelas derivadas de cada um dos termos da equação (3.24):

$$dZ = \begin{bmatrix} dZ_x \\ dZ_y \\ dZ_\theta \end{bmatrix}. \quad (3.31)$$

A derivada da equação (3.24) é:

$$dZ = \left[ \frac{1}{2} \left( -\frac{\partial R^T}{\partial Z_\theta} (x_{caP} + x_{caA} + 2x_a) dZ_\theta - R^T (dx_{caP} + dx_{caA} + 2dx_a) + dP + dA \right) \right]_{dZ_\theta} \quad (3.32)$$

e os vetores  $dA$  e  $dP$  estão relacionados aos erros de variação de posição dos pontos P e A, respectivamente, no mundo. No entanto, os valores desses pontos são absolutos e conhecidos por isso não possuem erros associados. O vetor  $x_a$  também é absoluto e, portanto seu erro  $dx_a$  é nulo também.

A derivada de  $dZ_\theta$  é dada por :

$$dZ_\theta = d\alpha + d\gamma = d\beta + d\gamma \quad (3.33)$$

sendo  $d\gamma = 0$  pois  $\gamma$  é um ângulo conhecido. Para o cálculo de  $dZ_\theta$  basta a derivada de  $\beta$  conforme a dedução abaixo:

$$dZ_\theta = d\beta \rightarrow d(\tan(\beta)) = \frac{1}{\cos^2(\beta)} d\beta \quad (3.34)$$

Derivando-se a equação (3.26), tem-se:

$$\frac{1}{\cos^2(\beta)} d\beta = \frac{1}{x_{rA}-x_{rP}} dz_{rP} - \frac{1}{x_{rA}-x_{rP}} dz_{rA} + \frac{z_{rP}-z_{rA}}{(x_{rA}-x_{rP})^2} dz_{rP} - \frac{z_{rP}-z_{rA}}{(x_{rA}-x_{rP})^2} dx_{rA}. \quad (3.35)$$

Isolando-se  $d\beta$  :

$$d\beta = \frac{\cos^2(\beta)}{x_{rA}-x_{rP}} (\tan(\beta) dx_{rP} + dy_{rP} - \tan(\beta) dx_{rA} - dy_{rA}),$$

ou

$$d\beta = \frac{\cos^2(\beta)}{x_{rA}-x_{rP}} \begin{bmatrix} \tan(\beta) & 1 & -\tan(\beta) & -1 \end{bmatrix} \begin{bmatrix} dx_{rP} \\ dy_{rP} \\ dx_{rA} \\ dy_{rA} \end{bmatrix} = C dr_{PA} \quad (3.36)$$

A equação (3.37) pode também ser representada através de uma matriz C e um vetor  $dr_{PA}$ . A matriz C e o vetor  $dr_{PA}$  que são definidos como:

$$C = \frac{\cos^2(\beta)}{x_{rA}-x_{rP}} \begin{bmatrix} \tan(\beta) & 1 & -\tan(\beta) & -1 \end{bmatrix} \quad (3.37)$$

e

$$dr_{PA} = \begin{bmatrix} dr_P \\ dr_A \end{bmatrix}. \quad (3.38)$$

e sendo  $dx_a = 0$ :

$$\begin{bmatrix} dx_{rP} \\ dy_{rP} \\ dx_{rA} \\ dy_{rA} \end{bmatrix} = \begin{bmatrix} dx_{caP} \\ dy_{caP} \\ dx_{caA} \\ dy_{caA} \end{bmatrix} \quad (3.39)$$

Substituindo  $dZ_\theta$  em (3.32), tem-se:

$$dZ = \begin{bmatrix} -\frac{1}{2} \left( \frac{\partial R^T}{\partial Z_\theta} (x_{caP} + x_{caA} + 2x_a) C dr_{PA} + R^T (B_P dr_P + B_A dr_A) \right) \\ C dr_{PA} \end{bmatrix}. \quad (3.40)$$

Finalmente  $dZ$  é reescrito como:

$$dZ = \begin{bmatrix} -\frac{1}{2} \left( \frac{\partial R^T}{\partial Z_\theta} C (x_{caP} + x_{caA} + 2x_a) + R^T [B_P \quad B_A] \right) \\ C \end{bmatrix} dr_{PA} \quad (3.41)$$

Da relação (3.40) define-se uma matriz  $D$  como:

$$D = -\frac{1}{2} \left[ \frac{\partial R^T}{\partial Z_\theta} (x_{caP} + x_{caA} + 2x_a) C + R^T [B_P \quad B_A] \right], \quad (3.42)$$

onde a matriz  $R$  tem a seguinte forma:

$$R = \begin{bmatrix} \cos(Z_\theta) & \sin(Z_\theta) \\ -\sin(Z_\theta) & \cos(Z_\theta) \end{bmatrix} \quad (3.43)$$

Reescrevendo-se  $dZ$  como:

$$dZ = \begin{bmatrix} D \\ C \end{bmatrix} \cdot dr_{PA} \quad (3.44)$$



Assume-se que as incertezas das projeções dos pontos nas câmeras são associadas a um erro de discretização  $\rho$  dado por:

$$\rho = \frac{l}{2N_{pix}}, \quad (3.45)$$

onde  $l$  é o comprimento da tela e  $N_{pix}$  é o número de *pixels* em ambas as câmeras.

A matriz de covariância  $\bar{V}$  é dada pela incerteza  $dZ$  da seguinte forma:

$$\bar{V} = \begin{bmatrix} D \\ C \end{bmatrix} E [D^T \quad C^T], \quad (3.46)$$

onde a matriz  $E$  é dada por:

$$E = \bar{\sigma}^2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.47)$$

sendo  $\bar{\sigma}$  o desvio padrão de erro da câmera.

Em síntese esse capítulo apresentou os dois principais métodos de medição de estado do robô necessários para o processo de localização no ambiente. A navegação por odometria, um método clássico o estado do robô é calculado a partir do deslocamento que cada roda desse percorre. O outro método, visão estereoscópica, embora utilizado nesse trabalho de uma maneira simplificada, possibilita a localização visual no ambiente com de cálculo de profundidade e consequentemente triangulação do robô a partir de pontos de referência.

O capítulo seguinte inicia a abordagem de controle de robôs móveis com a introdução de tópicos de controle ótimo de sistemas dinâmicos sob a ótica do princípio de otimalidade de Bellman.

## Capítulo 4

### Controle Ótimo Baseado na Programação Dinâmica

Este capítulo é dedicado a formulação de problemas gerais de controle ótimo linear e não-linear numa abordagem de Programação Dinâmica de Bellman. Na primeira seção é exposta a idéia básica da programação Dinâmica e princípio de Otimalidade através de exemplos clássicos. Na seção seguinte aborda-se o controle ótimo de sistemas através da programação dinâmica. Na seção 4.3 é formulado o problema geral de controle ótimo para sistemas lineares com abordagem de programação dinâmica e equação HJB. E por último, na seção 4.4 é formulado um problema geral de controle ótimo não-linear nessa mesma abordagem.

#### 4.1. Idéia básica de Programação dinâmica e Princípio de Otimalidade

Programação dinâmica é uma técnica concebida por Bellman, que leva seu nome, porque tem um fundo computacional e porque é adequada a resolução de controle ótimo para sistemas dinâmicos. Sua idéia básica é um problema de otimização discreto, de vários estágios no qual a cada passo é escolhida uma decisão, dentre um conjunto finito de decisões possíveis, a partir de um critério de otimização associado a cada uma dessas decisões possíveis.

Uma das bases da programação dinâmica é o princípio de otimalidade. Ele define que cada porção de um segmento ótimo é também ótimo (Bellman, 1957). Alternativamente (Kirk, 1970) a política (controle) ótima tem a propriedade de independentemente de decisões anteriores deverá continuar uma política ótima.

## 4.2. Controle Ótimo de Sistemas através de Programação Dinâmica

Seja uma planta descrita por

$$x_{k+1} = f(x_k, u_k, k) \quad (4.1)$$

E o funcional de custo associado:

$$J = S(x_{k_f}, k_f) + \sum_{k=i}^{k_f-1} V(x_k, u_k) \quad , \quad (4.2)$$

onde  $x_k$  é o vetor de estado de ordem  $n$  e  $u_k$  é o vetor de controle com  $r$  elementos. Para encontrar um controle ótimo  $u^*$  empregando o princípio de otimalidade e obter o estado ótimo  $x^*$  é preciso calcular todos os estados e custos admissíveis pelas restrições da planta em cada momento do tempo. Para isso, dado um conjunto de valores admissíveis para controle  $u$  calcula-se a cada instante  $k$  do tempo os estados admissíveis pelo sistema e os custos associados. Escolhe-se então o valor correspondente ao custo mínimo como o controle ótimo  $u_k$  para aquele trecho/instante  $k$  do tempo.

A cada momento  $k$ , o princípio de otimalidade por ser utilizado para reescrever (Naydu, 2002):

$$J_k^*(x_k) = \min_{u_k} [V(x_k, u_k) + J_{k+1}^*(x_{k+1}^*)] \quad (4.3)$$

A relação (4.3) é uma forma do princípio de otimalidade aplicada ao controle ótimo de sistemas também chamado de equação da programação dinâmica. Ela significa que se for possível achar o controle, o estado e o custo ótimos de qualquer estágio  $k+1$  ao estágio final  $k_f$  então é possível encontrar valores ótimos para cada estágio único do  $k$  ao  $k+1$ . (Naydu 2002)

### 4.3. Controle Ótimo de Sistemas de tempo contínuo através da equação de Hamilton Jacobi Bellman

Já foi dito anteriormente que a técnica de programação dinâmica de Bellman é de natureza discreta, no entanto ela pode ser aplicada para o controle ótimo de sistemas contínuos. Há duas maneiras de tornar isso possível seja através de uma discretização valendo-se de métodos como método de Euler, seja aplicando a programação dinâmica diretamente ao sistema de tempo contínuo. Neste último caso, isso resultará na equação de Hamilton –Jacobi –Bellman (HJB) como explicado a seguir:

Considerando um sistema de tempo contínuo

$$\dot{x} = f(x(t), u(t), t) \quad , \quad (4.4)$$

e um funcional de performance:

$$J(x(t_0), t_0) = \int_{t_0}^{t_f} V(x(t), u(t), t) dt, \quad (4.5)$$

deseja-se encontrar um controle ótimo  $u^*$  que conduz o sistema (4.4) ao estado ótimo minimizando o funcional (4.5). Aplicando –se o princípio de otimalidade, na trajetória ótima o índice de performance  $J^*$  é mínimo e calculado ao longo dessa trajetória (do tempo  $t$  ao  $t_f$ ) pode ser definido como :

$$J^*(x^*(t), t) = \int_t^{t_f} V(x^*(\tau), u^*(\tau), \tau) d\tau \quad . \quad (4.6)$$

Deseja-se obter o controle  $u^*$  para qualquer condição inicial, portanto deve-se expressar  $J(x(t_0), t_0)$  como uma função de  $x_0$  e  $t_0$ .

$$\begin{aligned} \frac{dJ^*(x^*(t), t)}{dt} &= \left( \frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)^T \dot{x}^*(t) + \frac{\partial J^*(x^*(t), t)}{\partial t} = \\ &= \left( \frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)^T f(x^*(t), u^*(t), t) + \frac{\partial J^*(x^*(t), t)}{\partial t} \end{aligned} \quad (5.7)$$

Da equação (4.6) temos que:

$$\frac{dJ^*(x^*(t), t)}{dt} = -V(x^*(t), u^*(t), t) \quad (4.8)$$

Utilizando (4.7) e (4.8) resulta em :

$$\frac{\partial J^*(x^*(t), t)}{\partial x^*} + V(x^*(t), u^*(t), t) + \left( \frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)^T f(x^*(t), u^*(t), t) = 0 \quad (4.9)$$

Introduzindo-se o Hamiltoniano como:

$$\mathcal{H} = V(x^*(t), u^*(t), t) + \left( \frac{\partial J^*(x^*(t), t)}{\partial x^*} \right)^T f(x^*(t), u^*(t), t) \quad (4.10)$$

Substituindo a expressão do Hamiltoniano na equação (4.9) resulta:

$$\frac{\partial J^*(x^*(t), t)}{\partial x^*} + \mathcal{H} \left( x^*(t), \frac{\partial J^*(x^*(t), t)}{\partial x^*}, u^*(t), t \right) = 0; \forall t \in [t_0, t_f] \quad (4.11)$$

Tendo como condição de contorno de (4.6) a restrição:

$$J^*(x^*(t_f), t_f) = 0 \quad (4.12)$$

Caso o funcional de custo, no momento terminal, contenha uma função de custo  $S$ , a condição de contorno será:

$$J^*(x^*(t_f), t_f) = S(x^*(t_f), t_f) \quad (5.13)$$

A equação (4.11) é a equação de Hamilton-Jacobi e por ser análoga ao processo discreto de programação dinâmica, é denominada de Hamilton –Jacobi –Bellman HJB.

#### 4.4. Regulador Linear Quadrático com a Equação HJB

Considerando um sistema linear:

$$\dot{x} = A(t)x(t) + B(t)u(t)$$

$$x(0) = x_0 \quad (4.14)$$

Onde  $x \in R^n$ , é o vetor de estado,  $u \in R^m$  é o vetor de controle,  $A \in R^{n \times n}$  e  $B \in R^{n \times m}$  são matrizes positivo-semidefinidas.

Um problema de controle ótimo minimizando o índice de performance dado por:

$$J[u] = \int_{t_0}^{t_f} [x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)]dt \quad (4.15)$$

é um problema chamado de regulador linear ótimo com funcional quadrático, no qual as matrizes :  $Q \in \mathfrak{R}^{n \times n}$  é simétrica semidefinida positiva e  $R \in \mathfrak{R}^{m \times m}$  é simétrica positivo-definida.

Se  $t_f$  é finito então o problema (4.14)-(4.15) é chamado de problema linear quadrático de controle ótimo com horizonte finito. Nesse caso todas as matrizes dependem do tempo. Se  $t_f = \infty$  então todas as matrizes são constantes e o problema mencionado é denominado de problema de controle ótimo linear quadrático com horizonte infinito ou regulador linear quadrático.

Esse problema pode ser resolvido através da abordagem da programação dinâmica onde a equação HJB tem a seguinte forma:

$$\min_{u(t)} \left[ \frac{\partial V}{\partial t} + (gradV)^T (A(t)x + B(t)u) + x^T(t)Q(t)x(t) + u^T(t)R(t)u(t) \right] = 0, \quad (4.16)$$

onde o vetor  $gradV = \left[ \frac{\partial V}{\partial x_1} \quad \frac{\partial V}{\partial x_2} \quad \dots \quad \frac{\partial V}{\partial x_n} \right]^T$ .

A solução para equação HJB (5.16) será procurada em forma de função de Lyapunov:

$$V(x, t) = x^T P(t) x \quad (5.17)$$

Onde  $P(t)$  é uma matriz a ser determinada.

Admitindo que  $gradV = 2P(t)x$ , da condição de mínimo de (4.16) obtemos o controle ótimo:

$$u = -R^{-1}(t)B^T(t)P(t)x. \quad (4.18)$$

Substituindo  $u$ , dado por (4.18) na equação HJB (4.16) e admitindo que  $P(t)$  é simétrica, obtém-se:

$$x^T \left[ \frac{\partial P(t)}{\partial t} + 2A(t) - 2P(t)B(t)R(t)^{-1}B(t)^T P(t) + Q(t) + P(t)B(t)R(t)^{-1}B(t)^T P(t) \right] x = 0, \quad (4.19)$$

Essa equação é satisfeita para quaisquer valores de  $x$  se, e somente se,  $P(t)$  satisfaz a seguinte equação diferencial de Riccati:

$$\frac{dP(t)}{dt} + P(t)A(t) + A(t)^T P(t) - P(t)B(t)R(t)^{-1}B(t)^T P(t) + Q(t) = 0 \quad (4.20)$$

Por definição  $V(x, t_f) = 0$ , assim a condição final para a equação (4.20) será:

$$P(t_f) = 0 \quad (4.21)$$

Para sistemas autônomos, as matrizes  $A$ ,  $B$ ,  $Q$  e  $R$  são constantes. A seguir é utilizada a mesma formulação para o sistema autônomo:

$$\dot{x} = Ax(t) + Bu(t) \quad (4.22)$$

com

$$J[u] = \int_0^\infty [x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)]dt \quad (4.23)$$

Para  $t_f = \infty$  a função  $V(x)$  não depende do tempo e a equação de Riccati se torna algébrica:

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (4.24)$$

Já a lei de controle ótimo na forma

$$u = -R^{-1}B^T Px \quad (4.25)$$

Fornece um valor mínimo ao funcional (4.23) calculado nas trajetórias ótimas do sistema (4.22).

A estabilidade assintótica do sistema (4.14) é assegurada através da propriedade positivo definida das matrizes  $Q$  e  $R$ . A condição suficiente para que o valor de  $J_0$  seja finito é a controlabilidade completa para matrizes  $A$  e  $B$ , ou seja, a matriz de controlabilidade:

$$[B \quad AB \quad \dots \quad A^{n-1}B] \quad (4.26)$$

deve ser não-singular, i.e., conter  $n$  colunas linearmente independentes. Nesse caso:

$$\text{rank}[B \quad AB \quad \dots \quad A^{n-1}B] = n \quad (4.27)$$

#### 4.5. Formulação Geral de um Problema de Controle Ótimo Não-Linear

Consideraremos o problema do controle ótimo com horizonte infinito para um sistema autônomo não-linear na seguinte forma:

$$\dot{x} = f(x) + B(x)u \quad (4.28)$$

$$x(0) = x_0 \quad (4.29)$$

onde  $x \in \mathfrak{R}^n$  é vetor de estado,  $u \in \mathfrak{R}^m$  é vetor de controle, e  $t \in [0, \infty)$ , com funções  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  e  $B: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  que pertencem ao espaço  $C^1$ , e  $B(x) \neq 0$  para todos os  $x$ . Assume-se que a origem  $x = 0$  é ponto de equilíbrio do sistema sem controle, ou seja,  $f(0) = 0$ . O problema do regulador não-linear com horizonte infinito é definido, minimizando o seguinte funcional, que é não-quadrático em relação a  $x$ , mas quadrático em relação a  $u$ :

$$V[x] = \min_{u(t)} \int_0^{\infty} [x^T Q(x)x + u^T R(x)u] dt, \quad (4.30)$$



onde as matrizes de coeficientes de peso de variáveis de estado e de controle dependem de variáveis de estado tal que a matriz  $Q: \mathfrak{R}^n \rightarrow \mathfrak{R}^{n \times n}$  é semidefinida positiva e a matriz  $R: \mathfrak{R}^n \rightarrow \mathfrak{R}^{m \times m}$  é definida positiva para todos os  $x$ . Assumimos que as funções  $f(x)$ ,  $B(x)$ ,  $Q(x)$  e  $R(x)$  são suficientemente suaves tal que a função valor, definida pela equação (4.30) é continuamente diferenciável. Sabe-se que o controlador não-linear ótimo para sistema (4.28) pode ser proposto na seguinte forma:

$$u = -\frac{1}{2} R^{-1}(x) B^T(x) \frac{\partial V}{\partial x}, \quad (4.32)$$

onde  $\frac{\partial V}{\partial x}$  satisfaz a equação de Hamilton-Jacobi-Bellman (HJB):

$$\frac{\partial V^T}{\partial x}(x) f(x) - \frac{1}{4} \frac{\partial V^T}{\partial x}(x) B(x) R^{-1}(x) B^T(x) \frac{\partial V}{\partial x} + x^T Q(x) x = 0. \quad (4.32)$$

Em um caso geral, a equação de Hamilton-Jacobi-Bellman (4.32) providencia a solução do problema do controle ótimo para sistema não-linear (4.28); mas na maioria dos casos é impossível resolver (4.32) analiticamente. Existem vários trabalhos na literatura que tentam resolver a equação diferencial parcial não-linear numericamente, mas isto leva a cálculos que dificilmente podem ser realizados *on line*. Outros métodos tentam evitar a resolução da equação (4.32). Um desses métodos é SDRE (*State Dependent Riccati Equation*), que permite encontrar o controle subótimo do sistema (4.28) será apresentado na seção 5.2 dessa tese.

## Capítulo 5

### Métodos de Controle Aplicados ao Robô Móvel

Nesse capítulo são apresentados os métodos de controle propostos e aplicados para o controle de trajetória de um robô móvel. O primeiro deles é o método por realimentação de estado linear proposto neste trabalho para o problema de controle de sistemas não-holonômicos. O segundo método é controle SDRE, baseado na equação de Riccati dependente de estado e o terceiro é o método SDRE aplicado para sistemas discretos.

#### 5.1. Formulação do Controle linear por Realimentação de Estado Ótimo.

Considerando o sistema de Brockett, equivalente ao modelo cinemático do robô na forma :

$$\begin{aligned} \dot{z}_1 &= U_1 \\ \dot{z}_2 &= U_2 \\ \dot{z}_3 &= z_2 U_1 - z_1 U_2 \end{aligned} \quad , \quad (5.1)$$

onde:

$$\begin{aligned} U_1 &= \omega \\ U_2 &= v - \omega(x \sin \theta - y \cos \theta) \end{aligned} \quad (5.2)$$

que é obtido através da transformação:

$$z = T \begin{bmatrix} \theta \\ x \\ y \end{bmatrix}, \quad (5.3)$$

onde  $T \in \mathbb{R}^{3 \times 3}$  é a matriz de transformação definida por:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & 2\sin(\theta) - \theta\cos(\theta) & -2\cos(\theta) - \theta\sin(\theta) \end{bmatrix} \quad (5.4)$$

Em seguida, apresentam-se as coordenadas transformadas em termos de desvio, definindo uma trajetória de referência como:

$$\tilde{x} = \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \tilde{x}_3(t) \end{bmatrix} \quad (5.5)$$

O fato do controle não ser aplicado ao sistema original  $x, y, \theta$ , mas ao sistema de Brockett (5.1) descrito em coordenadas  $z$ , transformadas, demanda a transformação a trajetória de referencia em coordenadas  $\tilde{z}$  para viabilizar a sua utilização:

$$\tilde{z} = \begin{bmatrix} \tilde{z}_1(t) \\ \tilde{z}_2(t) \\ \tilde{z}_3(t) \end{bmatrix} = \tilde{T} \begin{bmatrix} \tilde{\theta}(t) \\ \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix}, \quad (5.6)$$

onde:

$$\tilde{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\tilde{\theta}) & \sin(\tilde{\theta}) \\ 0 & 2\sin(\tilde{\theta}) - \tilde{\theta}\cos(\tilde{\theta}) & -2\cos(\tilde{\theta}) - \tilde{\theta}\sin(\tilde{\theta}) \end{bmatrix}. \quad (5.7)$$

Deste modo o desvio do robô da trajetória de referência pode ser definido como um erro  $e(t) \in \mathbb{R}^3$ :

$$e = z - \tilde{z} \quad (5.8)$$

Ao mesmo tempo, o vetor de controle  $[\tilde{u}_1 \ \tilde{u}_2]^T$  representa o controle *feedforward* que mantém o sistema na trajetória desejada (5.5) e satisfaz as seguintes condições:

$$\begin{aligned}\dot{\tilde{z}}_1 &= \tilde{u}_1 \\ \dot{\tilde{z}}_2 &= \tilde{u}_2 \\ \dot{\tilde{z}}_3 &= \tilde{z}_1 \tilde{u}_2 - \tilde{z}_2 \tilde{u}_1\end{aligned}\tag{5.9}$$

O controle  $U$  pode ser visto como a soma dos controles *feedforward* (em malha aberta) e realimentação de estados (*feedback*):

$$U = u + \tilde{u}\tag{5.10}$$

Conseqüentemente, o controle por realimentação de estado  $u$  que realiza o *tracking* do sistema à trajetória (5.5) é dado por :

$$u = U - \tilde{u}\tag{6.11}$$

Tendo –se em conta que as relações (5.4),(5.8) equivalem também para as derivadas temporais daquelas expressões, segue então que o sistema (5.5),(5.9) pode ser expresso através da diferença (5.8) e (5.11):

$$\begin{aligned}\dot{e}_1 &= \dot{x}_1 - \dot{\tilde{x}}_1 = U_1 - \tilde{u}_1 \\ \dot{e}_2 &= \dot{x}_2 - \dot{\tilde{x}}_2 = U_2 - \tilde{u}_2 \\ \dot{e}_3 &= x_1 U_2 - x_2 U_1 - \tilde{x}_1 \tilde{u}_2 + \tilde{x}_2 \tilde{u}_1\end{aligned}\tag{5.12}$$

O sistema (5.9) em coordenadas de desvio é dado por:

$$\begin{aligned}\dot{e}_1 &= u_1 \\ \dot{e}_2 &= u_2 \\ \dot{e}_3 &= \tilde{u}_2 e_1 - \tilde{u}_1 e_2 + (\tilde{z}_1 + e_1)u_2 - (\tilde{z}_2 + e_2)u_1\end{aligned}\tag{5.13}$$

O sistema (5.13) pode ser reescrito na seguinte forma matricial:

$$\dot{e} = A(t)e + h(e, u) + B(t)u,$$

com

$$e(0) = e_0, \quad (5.14)$$

onde os valores das matrizes são:

$$A(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{u}_2 & -\tilde{u}_1 & 0 \end{bmatrix}, h(e, u) = \begin{bmatrix} 0 \\ 0 \\ e_1 u_2 - e_2 u_1 \end{bmatrix}, B(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\tilde{z}_2 & \tilde{z}_1 \end{bmatrix}. \quad (5.15)$$

Na equação (5.14),  $e \in \mathfrak{R}^n$  é o vetor de estado,  $A(t) \in \mathfrak{R}^{n \times n}$  e  $B(t) \in \mathfrak{R}^{n \times m}$  são matrizes cujos elementos são dependentes do tempo,  $u \in \mathfrak{R}^m$  é o vetor de controle e  $h(e, u) \in \mathfrak{R}^n$  é um vetor que agrega como elementos funções não-lineares contínuas com  $h(0,0) = 0$ . É importante notar que uma representação clássica de sistema controlado contém apenas as matrizes  $A(t)$  e  $B(t)$  o que significa que o sistema encontra-se na forma linear. O teorema a seguir assegura a estabilidade local assintótica do controlador.

**Teorema 5.1** . Caso existam matrizes  $Q(t)$  e  $R(t)$  positivas definidas, sendo  $Q(t)$  simétrica, tal que a função

$$l(e, u) = e^T Q \cdot e - h^T(e, u) P \cdot e - e^T \cdot P \cdot h(e, u), \quad (5.16)$$

é positiva definida, então o controle por realimentação de estado linear  $u$ :

$$u = -R^{-1} B^T P(t) e \quad (5.17)$$

é ótimo para levar o sistema (5.13) da condição inicial ao estado final:

$$e(t_f) = 0, \quad (5.18)$$

minimizando o funcional:

$$J = \int_0^{t_f} [l(e, u) + u^T R(t) u] dt. \quad (5.19)$$

A matriz  $P(t)$  na equação (5.17) é simétrica, positiva definida (para todo  $t \in [0, t_f]$ ) e é a solução da equação diferencial matricial de Riccati:

$$\dot{P}(t) + P(t)A(t) + A(t)^T P(t) - P(t)B(t)R(t)^{-1}B(t)^T P(t) + Q(t) = 0, \quad (5.20)$$

satisfazendo a condição final:

$$P(t_f) = 0 \quad . \quad (5.21)$$

Além disto, existe uma vizinhança  $\Gamma_0 \subset \Gamma$ ,  $\Gamma_0 \subset \mathbb{R}^n$ , da origem, tal que se  $e_0 \in \Gamma_0$ , a solução  $e(t) = 0, t \geq 0$  do sistema controlado (5.9)(5.14) é local e assintoticamente estável, e  $J_{min} = e_0^T P(0)e_0$ .

Finalmente, se  $\Gamma = \mathbb{R}^n$  então a solução  $e(t) = 0, t \geq 0$  do sistema controlado (5.14) é global e assintoticamente estável.

**Prova:** Consideremos um controle linear por realimentação de estado (5.17) com a matriz  $P$  determinada pela equação (5.20) que transfere o sistema não linear (5.14) do estado inicial ao estado final (5.18) minimizando o funcional (5.19), onde a função  $l(e, u)$  deve ser determinada.

De acordo com as regras da Programação Dinâmica (Bryson e Ho, 1975) sabe-se que se o mínimo de funcional (5.19) existe e se  $V$  é uma função suave das condições iniciais, então ela satisfaz a equação de Hamilton-Jacobi –Bellman a seguir:

$$\min_u \left( \frac{dV}{dt} + e^T \tilde{Q}e + u^T R u \right) = 0 \quad . \quad (5.22)$$

Considerando uma função de Lyapunov:

$$V = e^T P(t)e, \quad (5.23)$$

onde  $P(t)$  é uma matriz simétrica positiva definida que satisfaz a equação de Riccati (5.20).

A derivada da função  $V$ , calculada na trajetória ótima com controle dado por (5.17) é :

$$\begin{aligned}
\dot{V} &= \dot{e}^T P e + e^T \dot{P} e + e^T P \dot{e} = [e^T A^T + \\
&+ h^T(e, u) - e^T P B (R^{-1})^T B^T] P e + e^T \dot{P} e + e^T P [A e + \\
&+ h(e, u) - B R^{-1} B^T P e].
\end{aligned} \tag{5.24}$$

Substituindo  $\dot{V}$  na equação de Hamilton-Jacobi-Bellman obtém-se:

$$\begin{aligned}
&e^T [\dot{P} + A^T P + P A - P B R^{-1} B^T P] e + h^T(e, u) P e + \\
&+ e^T P h(e, u) + l(e, u) = 0
\end{aligned} \tag{5.25}$$

Daí decorre que:

$$l(e, u) = e^T Q e - h^T(e, u) P e - e^T P h(e, y) \tag{5.26}$$

Note que para condições onde a função  $l(e, u)$  é positiva definida e a matriz  $R$  é positiva definida, a derivada da função (5.26), calculada na trajetória ótima do sistema (5.14), é dada por:

$$\dot{V} = -l(e, u) - u^T R u, \tag{5.27}$$

Sendo que esta é negativa definida. A função (5.26), então, é uma função de Lyapunov e o sistema controlado (5.14) é local e assintoticamente estável. Integrando a derivada da função de Lyapunov (5.26) dada por

$$\dot{V} = -e^T \tilde{Q} e - u^T R u \tag{5.28}$$

ao longo da trajetória ótima, obtém-se que:

$$J_{min} = -e_0^T P(0) e_0^T \tag{5.29}$$

Finalmente se  $\Gamma = \mathfrak{R}^n$ , a estabilidade assintótica global segue como uma consequência direta da condição de não-limitação para a função de Lyapunov (5.26) :  $V(e) \rightarrow \infty$  e  $\|e\| \rightarrow \infty$ .

Destaca-se que, de acordo com a teoria de controle ótimo de sistemas lineares com funcional quadrático (Anderson e Moor, 1990), a solução da equação não-linear diferencial matricial de Riccati (5.20) é uma matriz positivo definida e simétrica para todo  $t \in [0, t_f]$ . Quando a função  $h(e, u) = h(e)$  esse teorema é equivalente ao postulado em (Rafikov e Balthazar, 2004)

## **5.2. Controle SDRE- *State Dependent Riccati Equation*- Equação de Riccati Dependente do Estado**

Desde o surgimento do Regulador Linear Quadrático (LQR), vários pesquisadores, motivados pela idéia desse método, propuseram um enfoque similar para resolver o problema do regulador não-linear (Pearson, 1962; Garrard et al., 1967, Burghart , 1969; Werli e Cook, 1975). Na última década, vários trabalhos foram publicados aplicando uma técnica de controle chamado de *State-Dependent Riccati Equation* (SDRE) para uma variedade de problemas de controle. Destacam-se o projeto de autopiloto (Mracek e Cloutier, 1996), desenvolvimento de leis de guiamento avançadas (Cloutier e Zipfel, 1999), controle de satélites e espaçonaves (Hammet et al. ,1998), controle de braço robótico (Erdem e Alleyne, 2001), problema de benchmark (Mracek e Cloutier, 1998), controle de ventilação de dutos (Yu et al. (2001)), controle de veículo autônomo subaquático (Naik e Singh, 2007), sincronização de sistemas caóticos (Jayaram e Tadi, 2006), entre outras. Uma visão geral dos avanços em técnica SDRE pode ser encontrada em (Çimen, 2008).

Diferentemente de métodos de controle não-lineares, o SDRE é de implementação elementar e altamente adequado o uso *on line* . A idéia de utilização de controle *por realimentação de estado* baseado em SDRE para sistemas não lineares remonta a 1962 por Pearson. Nessa referência, a estabilização do sistema ao equilíbrio e a resolução da equação diferencial de Riccati dependente do tempo é obtida analiticamente para um exemplo de baixa ordem. Mais tarde a idéia foi expandida por Wernli & Cook (1975). A maior parte do desenvolvimento dessa estratégia aconteceu, porém, nas décadas de 90 e na última década com os trabalhos de Banks em 1992 , Mracek & Cloutier (1998), Cloutier, et al. em 1998,



Cloutier, Stansbery & Sznaier (1999), que derivaram o condição necessária para existência de qualquer matriz de ganho de um controlador SDRE. Mracek e Cloutier (1998) provaram a estabilidade assintótica local para um caso de sistema de ordem  $n > 1$  com um controlador SDRE por realimentação de estado.

A seguir será exposto um enfoque geral do problema de controle ótimo não-linear sendo proposta uma solução de controlador dependente de estados e será também explicado o método SDRE.

Considera-se o problema do controle ótimo com horizonte infinito para um sistema autônomo não-linear :

$$\dot{x} = f(x) + B(x)u \quad (5.30)$$

$$x(0) = x_0 \quad (5.31)$$

onde  $x \in \mathfrak{R}^n$  é vetor de estado,  $u \in \mathfrak{R}^m$  é vetor de controle, e  $t \in [0, \infty)$ , com funções  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  e  $B: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  que pertencem ao espaço  $C^1$ , e  $B(x) \neq 0$  para todos os  $x$ . Assume-se que a origem  $x = 0$  é ponto de equilíbrio do sistema sem controle, ou seja,  $f(0) = 0$ . O problema do regulador não-linear com horizonte infinito é definido, através da minimização do seguinte funcional não-quadrático em relação a  $x$ , mas quadrático em relação a  $u$ :

$$V[x] = \min_{u(t)} \int_0^{\infty} [x^T Q(x)x + u^T R(x)u] dt, \quad (5.32)$$

As matrizes de coeficientes de peso de variáveis de estado e de controle, nesse funcional, dependem de variáveis de estado tal que a matriz  $Q: \mathfrak{R}^n \rightarrow \mathfrak{R}^{n \times n}$  é positiva semidefinida e a matriz  $R: \mathfrak{R}^n \rightarrow \mathfrak{R}^{m \times m}$  é positiva definida para todo o  $x$ . Assume-se que as funções  $f(x)$ ,  $B(x)$ ,  $Q(x)$  e  $R(x)$  são suficientemente suaves e a função valor definida pela (5.32) é continuamente diferenciável.

O controlador não-linear ótimo para sistema (5.32) pode ser proposto na forma (5.33):

$$u = -\frac{1}{2} R^{-1}(x) B^T(x) \frac{\partial V}{\partial x} \quad (5.33)$$

De acordo com Mracek e Cloutier (1998) o método SDRE para encontrar a solução subótima do problema (5.30)-(5.32) consiste das seguintes etapas:

- 1) Uso direto da parametrização para apresentar o sistema não-linear em forma similar à linear, com matrizes dependentes de variáveis do estado:

$$\dot{x} = A(x)x + B(x)u \quad (5.34)$$

onde:

$$f(x) = A(x)x ; \quad (5.35)$$

- 2) Solução da equação de Riccati com matrizes dependentes do estado para valores iniciais do sistema:

$$P(x)A(x) + A^T(x)P(x) - P(x)B(x)R^{-1}(x)B^T(x)P(x) + Q(x) = 0 \quad (5.36)$$

onde as funções  $P(x)$  são positivas definidas para todos os valores de  $x$ ;

- 3) Projeto do controlador *por realimentação de estado* não-linear:

$$u = -R^{-1} B^T P(x) x \quad (5.37)$$

- 4) Integração de (5.34) com o controle (5.37), encontrando o estado do sistema no próximo momento do tempo. Repetição do processo a partir da etapa 2.

Sob as hipóteses  $f(0)=0$  e  $f \in C^1(\mathfrak{R}^n)$ , sempre existe, pelo menos, uma função contínua matricial  $A(x)$ , que satisfaz (5.35). A matriz  $A : \mathfrak{R}^n \rightarrow \mathfrak{R}^{n \times n}$ , encontrada por meio da parametrização, não é única para  $n > 1$ .

Foram consideradas as seguintes propriedades do método SDRE, associadas com a forma (5.33) com coeficientes dependentes do estado em Mracek e Cloutier (1998), Çimen (2008), Cloutier et al. (1999)

**Definição 5.1.** A representação dependente do estado (5.34) é uma parametrização do sistema (5.30) estabilizável (controlável) na região  $\Omega \in \mathfrak{R}^n$ , se o par  $\{A(x), B(x)\}$  é estabilizável (controlável) ponto a ponto (*pointwise*) no sentido linear para todos os valores de  $x \in \Omega$ .

**Definição 5.2.** A representação dependente do estado (5.34) é uma parametrização do sistema (5.29) detectável (observável) na região  $\Omega \in \mathfrak{R}^n$ , se o par  $\{A(x), Q^{1/2}(x)\}$  é detectável (observável) ponto a ponto (*pointwise*) no sentido linear para todos os valores de  $x \in \Omega$ .

**Definição 5.3.** A representação dependente do estado (5.34) é de Hurwitz na região  $\Omega \in \mathfrak{R}^n$ , se os autovalores da matriz  $A(x)$  estão na parte esquerda do plano aberto  $\text{Re}(s) < 0$ , ou seja, têm as partes reais negativas para todos os valores de  $x \in \Omega$ .

**Definição 5.4.** Diz-se que o controle (5.33) é recuperável pelo controle SDRE na região  $\Omega$  se existe a parametrização dependente de estado (5.35) estabilizável ponto a ponto, uma matriz de peso  $Q(x)$  positiva semidefinida ponto a ponto, e uma matriz de peso  $R(x)$  positiva definida tais que o controlador resultante (5.37), dependente do estado satisfaz (5.33) para todos os valores de  $x$ .

**Teorema 5.2** (Cloutier et al. (1999)). O controle (5.33) é recuperável pelo controle SDRE (5.37) na região  $\Omega$ , se existe a parametrização dependente de estado (5.35) tal que a matriz

de dinâmica de malha fechada  $A_{CL} = A(x) - B(x)K(x)$  é de Hurwitz ponto a ponto em  $\Omega$ , e o ganho  $K(x)$  satisfaz a propriedade de fase mínima em cada ponto em  $\Omega$ . Ou seja, os zeros de malha de ganho  $K(x)[sI - A(x)]^{-1}B(x)$  encontram-se na parte esquerda do plano fechado, ponto a ponto.

O teorema 5.2 fornece as condições necessárias e suficientes para a recuperabilidade dos controles SDRE, mas é de difícil aplicação devido ao fato que há um número infinito de parametrizações dependentes de estado.

**Teorema 5.3** (Mracek e Cloutier, 1998 ).

Assumindo que  $A(x)$ ,  $B(x)$ ,  $Q(x)$  e  $R(x)$  são funções-valor de matrizes  $C^1(\mathfrak{R}^n)$  que os respectivos pares são estabilizáveis e detectáveis ponto-a-ponto no sentido linear para todo  $x \in \Omega$ . Então o regulador não linear (5.37) produz uma solução em malha fechada que é assintoticamente estável localmente.

**Teorema 5.4** (Cloutier et al, 1996). Se a matriz coeficiente de malha fechada  $A_{CL} = A(x) - B(x)K(x)$  é simétrica para todo  $x$ , então sob as condições dadas por teorema 5.2, a solução em malha fechada SDRE é global e assintoticamente estável.

**Teorema 5.5** (Mracek e Cloutier, 1998 )

Num caso geral, multivariável ( $n > 1$ ), a solução com realimentação de estados SDRE e suas trajetórias de estado e coestado associados satisfazem a primeira condição necessária para a otimalidade ( $\partial H / \partial u = 0$ ) do problema de regulador não linear ótimo (5.30) e (5.32): Além disso, se as funções matriciais  $A(x)$ ,  $B(x)$ ,  $Q(x)$  e  $R(x)$ , bem como seus gradientes  $\partial A(x) / \partial x$ ,  $\partial B(x) / \partial x$ ,  $\partial P(x) / \partial x$ ,  $\partial Q(x) / \partial x$  e  $\partial R(x) / \partial x$  são limitados na vizinhança  $\Omega$  da origem, sob estabilidade assintótica, o estado  $x$  é levado ao zero. A segunda condição necessária para a otimalidade ( $\dot{\lambda} = \partial H / \partial x$ ) é satisfeita assintoticamente a uma taxa quadrática.

**Teorema 5.6** (Mracek e Cloutier, 1998 )

Para sistemas escalares  $x \in \mathfrak{R}^1$ , a solução SDRE assintoticamente estabilizável do problema de regulador não linear ótimo é sempre (globalmente) ótima em  $\mathfrak{R}^1$ .

O teorema 5.6 mostra que a solução com realimentação de estado SDRE (5.34) proporciona a solução global ótima para o problema não linear quando  $n=1$ . Para um caso geral  $n > 1$ , a escolha apropriada para  $A(x)$  possui um papel importante para obtenção de boas soluções para o problema de controle. Os resultados a seguir apresentam as condições do problema de regulador não linear ótimo, para o caso de sistema multivariável, com  $n > 1$ .

**Teorema 5.7** (Huang e Lu, 1996)

Caso ambos  $f(x)$  e  $B(x)$  em (5.30) são suaves o suficiente com  $f(0)=0$  e a função de custo ótimo  $V(x)$  tem uma derivada na seguinte forma:

$$\frac{\partial V}{\partial x} = 2P(x)x \quad (5.38)$$

Para uma função matricial positivo-definida, se  $P: \mathfrak{R}^n \rightarrow \mathfrak{R}^{n \times n}$  então, existe sempre uma parametrização (5.35) tal que  $P(x)$  é a solução da equação de Riccati dependente de estado (5.36) que proporciona o controlador ótimo com realimentação de estado.

**Teorema 5.8** (Huang e Lu, 1996)

Se  $V$  é a solução positivo definida da equação HJB (5.33), então existe no máximo uma função positivo definida matricial  $P(x)$  tal que a condição (5.38) é satisfeita.

**Lema 5.1** (Huang e Lu, 1996) Suponha que uma função vetorial  $p: X \rightarrow \mathfrak{R}^n$  seja da classe  $C^1(\mathfrak{R}^n)$  e  $p(x) = [p_1(x), \dots, p_n(x)]^T$  para  $x \in X$ . Então existe  $V: X \rightarrow \mathfrak{R}^n$  tal que:

$$\frac{\partial V}{\partial x} = p(x)$$

se, e somente se:

$$\frac{\partial p_i}{\partial x_j}(x) = \frac{\partial p_j}{\partial x_i}(x), \quad (5.39)$$

para todo  $x \in X$  e  $i, j = 1, 2, \dots, n$ . Além disso, se (5.39) é satisfeita, então uma função  $V : X \rightarrow \Re^n$  com  $V(0) = 0$  é dada por

$$V = x^T \int_0^1 p(tx) dt \quad (5.40)$$

A condição de otimalidade segue diretamente do lema 5.2.

**Teorema 5.9** (Huang e Lu, 1996).

Considerando o sistema (5.34) e supondo que SDRE (5.36) possui uma solução matricial positivo-definida  $P : \Re^n \rightarrow \Re^{n \times n}$ . Se a função vetorial  $p(x) = 2P(x)x$  satisfaz (5.39) então (5.37) é o controlador com realimentação de estados para o problema (5.30) com índice de performance (5.32). A função valor é definida por:

$$V = 2x^T \int_0^1 tP(tx) dt x. \quad (5.41)$$

Os resultados acima obtidos são úteis para casos particulares quando, por exemplo, a função  $P$  pode ser expressa analiticamente. Em geral, a função matricial  $P$  é obtida numericamente.

Ainda no caso  $n > 1$ , por causa da liberdade que se tem na escolha de  $A(x)$ , a matriz parametrizada, o controle obtido não é mais ótimo mas sim subótimo e não minimiza o índice de performance (5.32). Por isso é importante saber qual a forma do funcional que é minimizado pelo controle subótimo (5.37) e o teorema 5.10 a esse respeito. Nesse sentido, consideramos o problema de controle (5.34) – (5.37).

**Teorema 5.10.**(Rafikov e Balthazar, 2009)

Se, para qualquer parametrização (5.34) controlável e observável, com matrizes de ponderação assumidas como dependentes de estado tal que  $Q: \mathfrak{R}^n \rightarrow \mathfrak{R}^{n \times n}$  é positiva semidefinida e  $R: \mathfrak{R}^n \rightarrow \mathfrak{R}^{m \times m}$  é positiva definida para todo  $x$ , a função:

$$L(x) = x^T Q(x)x - x^T \dot{P}(x)x \quad (5.42)$$

é positiva definida, então o controle por realimentação de estado (5.36) e (5.37) é ótimo para transferir o sistema não linear (5.26)-(5.34) do estado inicial ao estado final :

$$x(\infty) = 0 \quad (5.43)$$

minimizando o funcional

$$J = \int_0^{\infty} [L(x) + u^T R(x) u] dt \quad (5.44)$$

*Prova:* Consideremos um problema ótimo inverso, assumindo que o controle por realimentação de estado (5.37) é conhecido com uma matriz  $P$  determinada por SDRE (5.36), para uma dada parametrização (5.35) controlável e observável. De acordo com a formulação do teorema 5.9 , esse controle proporciona performance ótima para a parametrização do sistema (5.34), minimizando o funcional (5.44) com a função  $L(x)$  a ser determinada.

De acordo com as regras de programação dinâmica (Bryson, e Ho, 1975), sabe-se que se o mínimo do funcional (5.44) existe e se  $V$  é uma função suave das condições iniciais, então, ela satisfaz a equação de Hamilton-Jacobi-Bellman:

$$\min_u \left( \frac{dV}{dt} + L(x) + u^T R(x) u \right) = 0 \quad (5.45)$$

Considerando uma função de Lyapunov:

$$V = x^T P(x) x, \quad (5.46)$$

onde  $P(x)$  é uma matriz simétrica positivo definida que satisfaz a equação de Riccati (5.36).

Note que a derivada da função  $V$  calculada ao longo da trajetória com controle dado por (5.36) e (5.37) é:

$$\begin{aligned} \dot{V} &= \dot{x}^T P(x) x + x^T \dot{P}(x) x + x^T P \dot{x} = \\ &= [x^T A^T(x) - x^T P(x) B(x) (R^{-1}(x))^T B^T(x)] P(x) x + x^T \dot{P}(x) x + \\ &+ x^T P(x) [A(x) x - B(x) R^{-1}(x) B^T P(x) x] \end{aligned} \quad (5.47)$$

Substituindo  $\dot{V}$  na equação de HJB (5.43) obtém-se:

$$\begin{aligned} L(x) &+ [x^T A^T(x) - x^T P(x) B(x) (R^{-1}(x))^T B^T(x)] P(x) x + x^T \dot{P}(x) x + \\ &+ x^T P(x) [A(x) x - B(x) R^{-1}(x) B^T P(x) x] + \\ &+ x^T P(x) B(x) (R^{-1}(x))^T R(x) R^{-1}(x) B^T(x) P(x) x = 0 \end{aligned} \quad (5.48)$$

ou,

$$\begin{aligned} L(x) &+ x^T [A^T(x) P(x) - P(x) B(x) (R^{-1}(x))^T B^T(x) P(x) + \dot{P}(x) + \\ &+ P(x) A(x) - P(x) B(x) R^{-1}(x) B^T P(x) + \\ &+ P(x) B(x) (R^{-1}(x))^T B^T(x) P(x)] x = 0 \end{aligned} \quad (5.49)$$

ou

$$\begin{aligned} L(x) &+ x^T [A^T(x) P(x) + \dot{P}(x) + \\ &+ P(x) A(x) - P(x) B(x) R^{-1}(x) B^T P(x)] x = 0. \end{aligned} \quad (5.50)$$



Admitindo que :

$$P(x)A(x) + A^T(x)P(x) - P(x)B(x)R^{-1}(x)B^T(x)P(x) + Q(x) = 0, \quad (5.51)$$

obtém-se:

$$L(x) + x^T [\dot{P}(x) - Q(x)]x = 0. \quad (5.52)$$

Então:

$$L(x) = x^T Q(x)x - x^T \dot{P}(x)x \quad (5.53)$$

Com isso se conclui a prova do teorema 5.10.

### 5.2.1. Aplicação do Controle SDRE ao Robô Móvel

O primeiro procedimento para aplicação da técnica de controle SDRE é a parametrização do sistema considerado. No caso do sistema Brockett, retomamos os sistema (5.13) que representa o integrador de Brockett em coordenada de desvios:

$$\begin{aligned} \dot{e}_1 &= u_1 \\ \dot{e}_2 &= u_2 \\ \dot{e}_3 &= \tilde{u}_2 e_1 - \tilde{u}_1 e_2 + (\tilde{z}_1 + e_1)u_2 - (\tilde{z}_2 + e_2)u_1 \end{aligned} \quad (5.54)$$

O sistema (5.54) pode ser reescrito na forma matricial. Devido a transformações para levar o sistema de Brockett a coordenadas de desvio, a parametrização para a forma dependente do estado é escolhida agrupando-se os termos dependentes de  $u$  na matriz  $B$  e o restante na matriz  $A$  conforme segue:

$$\dot{e} = A(e)e + B(e)u, \quad (5.55)$$

$$\text{Onde } A(e) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{u}_2 & -\tilde{u}_1 & 0 \end{bmatrix} \text{ e } B(e) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -y_2 - \tilde{z}_2 & y_1 + \tilde{z}_1 \end{bmatrix}.$$

No caso do modelo alternativo encontrado em Kanayama *et al.* , 1990, para realizar a parametrização considera-se que o sistema segue um robô de referência:

$$\begin{aligned}\dot{\tilde{x}} &= v_r \cos \tilde{\theta} \\ \dot{\tilde{y}} &= v_r \sin \tilde{\theta} \\ \dot{\tilde{\theta}} &= \omega_r\end{aligned}\tag{5.56}$$

e o erro em relação ao robô de referência é dado por uma transformação:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} - x \\ \tilde{y} - y \\ \tilde{\theta} - \theta \end{bmatrix}\tag{5.57}$$

Desenvolvendo :

$$\begin{aligned}e_1 &= (\dot{\tilde{x}} - \dot{x}).\cos(\theta) - (\tilde{x} - x).\dot{\theta}.\sin(\theta) + (\dot{\tilde{y}} - \dot{y}).\sin(\theta) + \dot{\theta}.\cos(\theta).(\tilde{y} - y) \\ e_2 &= -(\dot{\tilde{x}} - \dot{x}).\sin(\theta) - (\tilde{x} - x).\dot{\theta}.\cos(\theta) + (\dot{\tilde{y}} - \dot{y}).\cos(\theta) - (\tilde{y} - y).\dot{\theta}.\sin(\theta) \\ e_3 &= \dot{\tilde{\theta}} - \dot{\theta}\end{aligned}\tag{5.58}$$

Substituindo  $\dot{\theta}$  por  $\omega$ :

$$\begin{aligned}e_1 &= (v_r \cos \tilde{\theta} - v \cos \theta).\cos \theta - (\tilde{x} - x).\omega.\sin(\theta) + (v_r \sin \tilde{\theta} - v \sin \theta).\sin(\theta) + \\ &\quad + \omega.\cos(\theta).(\tilde{y} - y) \\ e_2 &= -(v_r \cos \tilde{\theta} - v \cos \theta).\sin(\theta) - (\tilde{x} - x).\omega.\cos(\theta) + \\ &\quad + (v_r \sin \tilde{\theta} - v \sin \theta).\cos(\theta) - (\tilde{y} - y).\omega.\sin(\theta) \\ e_3 &= \dot{\tilde{\theta}} - \dot{\theta}\end{aligned}\tag{5.59}$$

O sistema toma, então, a seguinte forma:

$$\begin{aligned}e_1 &= v_r \cos(\tilde{\theta} - \theta) - v + \omega e_2 \\ e_2 &= v_r \sin(\tilde{\theta} - \theta) - \omega e_1 \\ e_3 &= \omega_r - \omega\end{aligned}\tag{5.60}$$

Definindo um vetor de controle como:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \omega_r - \omega \\ v_r \cos(\tilde{\theta} - \theta) - v \end{bmatrix} \quad (5.61)$$

e considerando (5.61), o sistema (5.60) se torna:

$$\begin{aligned} \dot{e}_1 &= u_2 + e_2 \omega_r - u_1 e_2 \\ \dot{e}_2 &= v_r \sin e_3 - \omega_r e_1 + e_1 u_1 \\ \dot{e}_3 &= u_1 \end{aligned} \quad (5.62)$$

Este pode ser reescrito na forma matricial:

$$\dot{e} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & \frac{v_r \sin(e_3)}{e_3} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} -e_2 & 1 \\ e_1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (5.63)$$

O sistema (5.63) já se encontra na forma parametrizada dependente do estado.

A equação algébrica de Riccati, dependente de estado, apresentada abaixo, é resolvida numericamente:

$$P(e)A(e) + A^T(e)P(e) - P(e)B(e)R^{-1}(e)B^T(e)P(e) + Q(e) = 0, \quad (5.64)$$

onde as funções  $P(e)$  são definidas positivas para todos os valores de  $e$ .

O controlador não-linear por realimentação de estado é dado por:

$$u = -R^{-1}(e)B^T(e)P(e)e \quad (5.65)$$

Em seguida, integra-se (5.63) e (5.65) para calcular o estado do sistema no próximo passo e assim repete-se o processo recursivamente.

### 5.3. Método de Controle SDRE para Sistemas Discretos Não Lineares

#### 5.3.1. Método SDRE discreto

Considera-se um sistema não-linear, discreto, descrito por:

$$x_{k+1} = f(x_k) + B(x_k)u_k \quad (5.66)$$

e o funcional de custo associado:

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x_k^T Q(x_k) x_k + u_k^T R(x_k) u_k] \quad (5.67)$$

onde  $x(k)$  é o vetor de estado de ordem  $n$ ,  $u(k)$  é o vetor de controle de ordem  $r$  e onde  $Q(x(k))$  é uma matriz positiva semidefinida para cada valor de  $k$  e  $R(x(k))$  é matriz positiva definida para cada valor de  $k$ .

O método SDRE para encontrar a solução subótima do problema (5.37)-(5.38) consiste nas seguintes etapas:

- 1) Uso direto da parametrização para apresentar o sistema não-linear (5.66) em forma similar à linear, com matrizes dependentes de variáveis do estado:

$$x_k = A(x_k) x_k + B(x_k) u_k \quad (5.68)$$

onde

$$f(x_k) = A(x_k) x_k \quad (5.69)$$

é uma parametrização (não-única) que pode ser realizada se  $f(0) = 0$ ;

- 2) Solução da equação de Riccati com matrizes dependentes do estado para valores  $x(k)$  do sistema:

$$P(x_k) = A^T(x_k) P(x_{k-1}) [I + B(x_k) R^{-1}(x_k) B^T(x_k) P(x_{k-1})]^{-1} A(x_k) + Q(x_k) \quad (5.70)$$

ou em forma:

$$P(x_k) = A^T(x_k) \{ P(x_{k-1}) - P(x_{k-1}) B(x_k) [B^T(x_k) P(x_{k-1}) B + R]^{-1} B^T(x_k) P(x_{k-1}) \} A(x_k) + Q(x_k) \quad (5.71)$$

3) Projeto do controlador por realimentação de estado discreto não-linear:

$$u = -L(x_k) x_k \quad (5.72)$$

ou

$$u = -L_a(x_k) x_k, \quad (5.73)$$

onde

$$L(x_k) = R^{-1}(x_k) B^T(x_k) A^{-T}(x_k) [P(x_{k-1}) - Q(x_k)] \quad (5.74)$$

e

$$L_a(x_k) = [B^T(x_k) P(x_{k-1}) B(x_k) + R(x_k)]^{-1} B^T(x_k) P(x_{k-1}) A(x_k) \quad (5.75)$$

4) Encontrar o estado ótimo do sistema  $x(k+1)$  no próximo momento do tempo.  
Repetir o processo a partir da etapa 2.

### 5.3.2. Aplicação do Método SDRE discreto para o robô móvel

Considerando um modelo de robô móvel discreto no tempo na forma:

$$\begin{aligned} x_{k+1} &= x_k + v_k \cos \theta_k \\ y_{k+1} &= y_k + v_k \sin \theta_k \\ \theta_{k+1} &= \theta_k + \omega_k \end{aligned} \quad (5.76)$$

Objetiva-se com que o robô siga um robô de referência na forma:

$$\begin{aligned} \tilde{x}_{k+1} &= \tilde{x}_k + \tilde{v}_k \cos \tilde{\theta}_k \\ \tilde{y}_{k+1} &= \tilde{y}_k + \tilde{v}_k \sin \tilde{\theta}_k \\ \tilde{\theta}_{k+1} &= \tilde{\theta}_k + \tilde{\omega}_k \end{aligned} \quad (5.77)$$

As velocidades de referência angular e linear são, respectivamente,  $\omega_r$  e  $v_r$  e contêm a informação essencial sobre a trajetória de referência que o robô deve seguir. Por isso definimos um desvio, subtraindo (5.76) de (5.77):

$$\begin{aligned}\tilde{x}_{k+1} - x_{k+1} &= \tilde{x}_k - x_k + \tilde{v}_k \cos \tilde{\theta}_k - v_k \cos \theta_k \\ \tilde{y}_{k+1} - y_{k+1} &= \tilde{y}_k - y_k + \tilde{v}_k \sin \tilde{\theta}_k - v_k \sin \theta_k \\ \tilde{\theta}_{k+1} - \theta_{k+1} &= \tilde{\theta}_k - \theta_k - \tilde{\omega}_k - \omega_k\end{aligned}\tag{5.78}$$

Introduzindo-se designações:

$$\begin{aligned}e_{1k} &= \tilde{x}_k - x_k \\ e_{2k} &= \tilde{y}_k - y_k \\ e_{3k} &= \tilde{\theta}_k - \theta_k\end{aligned}\tag{5.79}$$

Assim como para controle:

$$\begin{aligned}u_{1k} &= \tilde{V}_k - V_k \\ u_{2k} &= \tilde{\omega}_k - \omega_k\end{aligned}\tag{5.80}$$

Então, as equações descritas em (5.78) se tornam:

$$\begin{aligned}e_{1k+1} &= e_{1k} + \tilde{v}_k \cos \tilde{\theta}_k - v_k \cos \theta_k \\ e_{2k+1} &= e_{2k} + \tilde{v}_k \sin \tilde{\theta}_k - v_k \sin \theta_k \\ e_{3k+1} &= e_{3k} + u_{2k}\end{aligned}\tag{5.81}$$

Na primeira equação do sistema (5.81) adiciona-se e subtraí-se  $\tilde{v}_k \cos \tilde{\theta}_k$ . Na segunda equação de (5.81) adiciona-se e subtraí-se o termo  $\tilde{v}_k \sin \tilde{\theta}_k$ , obtendo assim:

$$\begin{aligned}e_{1k+1} &= e_{1k} + \tilde{v}_k \cos \tilde{\theta}_k - v_k \cos \theta_k + \tilde{v}_k \cos \tilde{\theta}_k - \tilde{v}_k \cos \tilde{\theta}_k \\ e_{2k+1} &= e_{2k} + \tilde{v}_k \sin \tilde{\theta}_k - v_k \sin \theta_k + \tilde{v}_k \sin \tilde{\theta}_k - \tilde{v}_k \sin \tilde{\theta}_k\end{aligned}\tag{5.82}$$

Reorganizando temos, para as duas primeiras equações de (5.74),(5.82):

$$\begin{aligned}e_{1k+1} &= e_{1k} + \tilde{v}_k (\cos \tilde{\theta}_k - \cos \theta_k) + (\tilde{v}_k - v_k) \cos \theta_k \\ e_{2k+1} &= e_{2k} + \tilde{v}_k (\sin \tilde{\theta}_k - \sin \theta_k) + (\tilde{v}_k - v_k) \sin \theta_k\end{aligned}\tag{5.83}$$

Levando em conta das relações  $e_{3k} = \tilde{\theta}_k - \theta_k$  e  $u_{1k} = \tilde{V}_k - V_k$  e inserindo-as em (5.83) temos:

$$\begin{aligned} e_{1k+1} &= e_{1k} + \tilde{v}_k [\cos \tilde{\theta}_k - \cos(\tilde{\theta}_k - e_{3k})] + u_{1k} \cos \theta_k \\ e_{2k+1} &= e_{2k} + \tilde{v}_k [\sin \tilde{\theta}_k - \sin(\tilde{\theta}_k - e_{3k})] + u_{1k} \sin \theta_k \end{aligned} \quad (5.84)$$

Dividindo-se e multiplicando –se por  $e_{3k}$  alguns termos de (5.84) chega-se a forma dependente de estado:

$$\begin{aligned} e_{1k+1} &= e_{1k} + \frac{\tilde{v}_k [\cos \tilde{\theta}_k - \cos(\tilde{\theta}_k - e_{3k})] \cdot e_{3k}}{e_{3k}} + u_{1k} \cos \theta_k \\ e_{2k+1} &= e_{2k} + \frac{\tilde{v}_k [\sin \tilde{\theta}_k - \sin(\tilde{\theta}_k - e_{3k})] \cdot e_{3k}}{e_{3k}} + u_{1k} \sin \theta_k \\ e_{3k+1} &= e_{3k} + u_{2k} \end{aligned} \quad (5.85)$$

O sistema (5.85) pode ser reescrito na forma:

$$e_{k+1} = A e_k + B u_k, \quad (5.86)$$

$$\text{onde } A = \begin{bmatrix} 1 & 0 & \frac{\tilde{v}_k [\cos \tilde{\theta}_k - \cos(\tilde{\theta}_k - e_{3k})] \cdot e_{3k}}{e_{3k}} \\ 0 & 1 & \frac{\tilde{v}_k [\sin \tilde{\theta}_k - \sin(\tilde{\theta}_k - e_{3k})] \cdot e_{3k}}{e_{3k}} \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \cos \theta_k & 0 \\ \sin \theta_k & 0 \\ 0 & 1 \end{bmatrix}, \quad e_k = \begin{bmatrix} e_{1k} \\ e_{2k} \\ e_{3k} \end{bmatrix} \text{ e } u_k = \begin{bmatrix} u_{1k} \\ u_{2k} \end{bmatrix}.$$

O sistema (5.86) representa uma possível parametrização do sistema (5.76) em coordenadas de desvio  $e$ .

Em seguida, para obter o controlador SDRE discreto é preciso prosseguir com a resolução numérica da equação de Riccati na forma:

$$P(e_k) = A^T(e_k) \cdot P(e_{k-1}) [I + B(e_k) R^{-1}(e_k) B^T(e_k) P(e_{k-1})]^{-1} A(e_k) + Q(e_k) \quad (5.87)$$

ou na forma:

$$P(e_k) = A^T(e_k) \{ P(e_{k-1}) - P(e_{k-1}) B(e_k) [B^T(e_k) P(e_{k-1}) B(e_k) + R]^{-1} B^T(e_k) P(e_{k-1}) \} A(e_k) + Q(e_k) \quad (5.88)$$

O controlador discreto SDRE é dado por

$$u = -L(e_k)e_k \quad (5.89)$$

ou

$$u = -L_a(e_k)e_k, \quad (5.90)$$

onde

$$L(e_k) = R^{-1}(e_k)B^T(e_k)A^{-T}(e_k)[P(e_k) - Q(e_k)] \quad (5.91)$$

e

$$L_a(e_k) = [B^T(e_k)P(e_k)B(e_k) + R(e_k)]^{-1}B(e_k)P(e_k)A(e_k) \quad (5.92)$$

Minimizamos, assim, um funcional na forma :

$$J = \frac{1}{2} \sum_{k=0}^{\infty} e_k^T Q(e_k) e_k + u_k^T R(e_k) u_k \quad (5.93)$$

Onde  $e_k = [e_{k1} \ e_{k2} \ e_{k3}]^T$  ,  $u_k = [\tilde{v}_k - v_k \ \tilde{\omega}_k - \omega_k]^T$  e  $Q(e_k)Q$  e  $R(e_k)$  são matrizes dependentes do estado positivo-semidefinida e positiva definida , respectivamente, para cada valor de  $k$ .

As simulações numéricas dos métodos apresentados nesse capítulo são mostradas no capítulo 7 , já o capítulo seguinte descreve o sistema simplificado para a realização de simulações numéricas da navegação do robô no ambiente , com a utilização de medições de estado , métodos de controle e estimação de estado através do filtro de Kalman.



## **Capítulo 6**

### **Proposta de Aplicação do Controle Dependente de Estado em um Sistema de Navegação de Robô Móvel**

O problema de navegação de um robô móvel pode ser desmembrado em diversos problemas menores cujas soluções em si são objeto de continuadas pesquisas. A navegação autônoma de um robô engloba problemas de planejamento de trajetória, localização através da medição de estado e estimação, controle de trajetória, identificação e desvio de obstáculos, comunicação do robô com o controle e eventualmente outros robôs, execução de tarefas específicas ou missões. No caso de uma arquitetura de vários robôs é necessário um sistema de planejamento, gerenciamento e execução de tarefas em colaboração e uma rede de comunicação entre os elementos envolvidos.

Neste trabalho a proposta é realizar uma aplicação de navegação robótica simples para demonstrar a aplicabilidade e a eficácia de algoritmos de controle propostos nas seções anteriores. O objetivo deste capítulo é integrar as ferramentas de visão de duas câmeras, a estimação de estado através da odometria e o controle da trajetória do robô para a construção de uma plataforma de navegação autônoma. Para isso é desenvolvido um sistema que possui diversos módulos descritos a seguir.

### 6.1. Visão geral do sistema de navegação

No sistema de navegação as medições de estado são provenientes de odometria e de estimação de estado através de visão de duas câmeras. Essas medições são integradas no estimador de estado que utiliza filtro de Kalman para dar maior confiabilidade à estimação. O sistema de navegação é representado de maneira esquemática pela figura 6.1 abaixo. O módulo de cálculo de referência define os parâmetros de uma determinada trajetória de referência escolhida pelo módulo de controle que também tem como entrada uma estimativa de seu estado. O módulo de controle é responsável pelo cálculo do controle  $u$  e pela simulação da trajetória real controlada do robô.

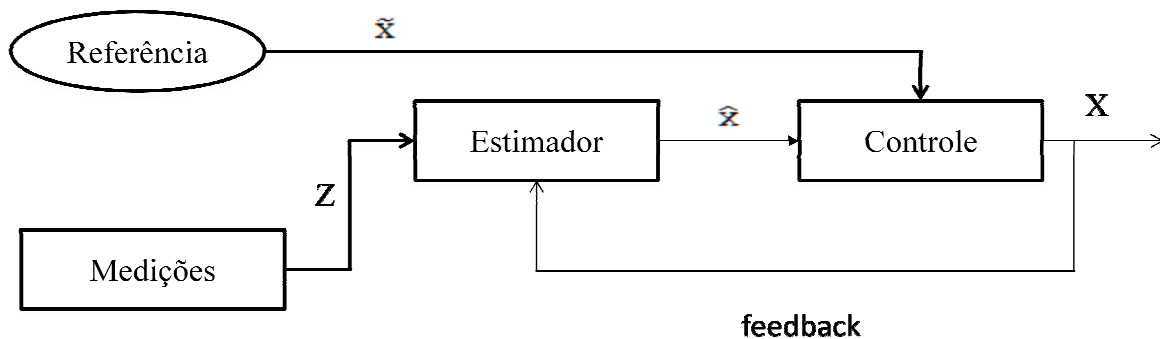


Figura 6. 1- Esquema do Sistema de Navegação

Cada uma das funcionalidades desses módulos, mostrados na figura acima, são descritos a seguir detalhadamente.

### 6.2. Definição de trajetória:

O robô encontra-se numa localização genérica. São dados dois pontos de localização A (alvo genérico) e P no espaço cujas coordenadas são absolutas. O robô inicia seu movimento no ponto  $x_0$  e deseja-se que siga uma trajetória de referência até o alvo ponto A.

No momento  $t=0$  o estado próprio de localização é desconhecido pelo robô. Por isso é preciso que se estime o estado pela medição das câmeras. Logo após é invocado o módulo

que decide a trajetória de referência a ser seguida que pode ser reta ou curva. Caso haja informação de algum obstáculo, determinada pela visão, escolhe-se um arco circular seguro em torno desse obstáculo para contorná-lo. Caso não haja informação sobre obstáculos determina-se uma trajetória linear até um alvo.

O controle do robô pode ser realizado através de dois métodos: um discreto e um contínuo. Para implementar o controle discreto é preciso a definição de uma trajetória de referência na forma discreta. Como já foi descrito no Capítulo 5 devido à formulação de controle o robô deve seguir um robô de referência na seguinte forma discreta:

$$\begin{aligned}\tilde{x}_{k+1} &= \tilde{x}_k + \tilde{v}_k \cos \tilde{\theta}_k \\ \tilde{y}_{k+1} &= \tilde{y}_k + \tilde{v}_k \sin \tilde{\theta}_k, \\ \tilde{\theta}_{k+1} &= \tilde{\theta}_k + \tilde{\omega}_k\end{aligned}\tag{6.1}$$

onde  $\tilde{v}_k$  e  $\tilde{\omega}_k$  são constantes.

Para a formulação do controle contínuo a forma da trajetória de referência escolhida é parametrizada pelo tempo, o que possibilita ao controle não somente levar o robô a uma referência, mas realizá-lo controlando a velocidade inicial e terminal. É possível, portanto levar o robô desde uma posição inicial até uma postura de repouso desejado.

Considerando uma trajetória de referência para coordenadas  $\tilde{x}(t)$  e  $\tilde{y}(t)$ , na forma parametrizada por um parâmetro  $t$ , representando o tempo:

$$\begin{aligned}\tilde{x}(t) &= a_1 t^3 + b_1 t^2 + c_1 t + d_1 \\ \tilde{y}(t) &= a_2 t^3 + b_2 t^2 + c_2 t + d_2\end{aligned}\tag{6.2}$$

Derivando-se (6.2) no tempo tem-se:

$$\begin{aligned}v_x(t) &= 3a_1 t^2 + 2b_1 t + c_1 \\ v_y(t) &= 3a_2 t^2 + 2b_2 t + c_2\end{aligned}\tag{6.3}$$

Considerando-se que a trajetória de referência evolui de um ponto inicial  $(\tilde{x}_0, \tilde{y}_0)$  até um ponto final  $(\tilde{x}_f, \tilde{y}_f)$ , e, considerando a equação (6.2) para a coordenada  $x$ , no momento  $t = t_f$  tem-se que :

$$\begin{aligned} a_1 t^3 + b_1 t^2 + c_1 t + d_1 &= \tilde{x}_f \\ 3a_1 t^2 + 2b_1 t + c_1 &= \dot{\tilde{x}}_f. \end{aligned} \quad (6.4)$$

Similarmente a (6.4), para coordenada  $y$ , obtém-se:

$$\begin{aligned} a_2 t_f^3 + b_2 t_f^2 + c_2 t_f + d_2 &= \tilde{x}_f \\ 3a_2 t_f^2 + 2b_2 t_f + c_2 &= \dot{\tilde{y}}_f. \end{aligned} \quad (6.5)$$

Considerando que no instante  $t=0$  tem-se que  $\tilde{x}_0 = d_1$ ,  $\tilde{y}_0 = d_2$  e que a velocidade inicial em  $x$  da trajetória de referência é dada por  $\tilde{v}_{0_x} = \dot{\tilde{x}}_0 = c_1$  e  $\tilde{v}_{0_y} = \dot{\tilde{y}}_0 = c_2$ . Temos então um sistema para  $x$ , na forma:

$$\begin{cases} a_1 t_f^3 + b_1 t_f^2 = \tilde{x}_f - \tilde{x}_0 - \dot{\tilde{x}}_0 t_f \\ 3a_1 t_f^2 + 2b_1 t_f = \dot{\tilde{x}}_f - \dot{\tilde{x}}_0 \end{cases} \quad (6.6)$$

Resolvendo o sistema com incógnitas  $a_1$  e  $b_1$  através do método de Cramer, o determinante principal é dado por:

$$\Delta = \begin{vmatrix} t_1^3 & t_1^2 \\ 3t_1^2 & 2t_1 \end{vmatrix} \quad (6.7)$$

E os dois determinantes secundários são dados por:

$$\Delta_{x1} = \begin{vmatrix} \tilde{x}_f - \tilde{x}_0 - \dot{\tilde{x}}_0 t_f & t_1^2 \\ \dot{\tilde{x}}_f - \dot{\tilde{x}}_0 & 2t_1 \end{vmatrix}$$

e

$$\Delta_{x2} = \begin{vmatrix} t_1^3 & \tilde{x}_f - \tilde{x}_0 - \dot{\tilde{x}}_0 t_f \\ 3t_1^2 & \dot{\tilde{x}}_f - \dot{x}_0 \end{vmatrix}. \quad (6.8)$$

Os parâmetros  $a_1$  e  $b_1$  obtidos de:

$$a_1 = \frac{\Delta_{x1}}{\Delta}, b_1 = \frac{\Delta_{x2}}{\Delta} \quad (6.9)$$

Realizando o mesmo processo de (6.6) a (6.9) para a coordenada  $y$ , resolve-se o sistema:

$$\begin{cases} a_1 t_f^3 + b_1 t_f^2 = \tilde{y}_f - \tilde{y}_0 - \dot{\tilde{y}}_0 t_f \\ 3a_1 t_f^2 + 2b_1 t_f = \dot{\tilde{y}}_f - \dot{y}_0 \end{cases}. \quad (6.10)$$

Os parâmetros  $a_2$  e  $b_2$  são determinados por:

$$a_1 = \frac{\Delta_{y1}}{\Delta}, b_1 = \frac{\Delta_{y2}}{\Delta}, \quad (6.11)$$

onde os determinantes principal e secundários são dados por:

$$\Delta = \begin{vmatrix} t_1^3 & t_1^2 \\ 3t_1^2 & 2t_1 \end{vmatrix},$$

$$\Delta_{y1} = \begin{vmatrix} \tilde{y}_f - \tilde{y}_0 - \dot{\tilde{y}}_0 t_f & t_1^2 \\ \dot{\tilde{y}}_f - \dot{y}_0 & 2t_1 \end{vmatrix}$$

e

$$\Delta_{y2} = \begin{vmatrix} t_1^3 & \tilde{y}_f - \tilde{y}_0 - \dot{\tilde{y}}_0 t_f \\ 3t_1^2 & \dot{\tilde{y}}_f - \dot{y}_0 \end{vmatrix} \quad (6.12)$$

Uma vez determinados os parâmetros  $a$  e  $b$  para  $x$  e  $y$ , é possível calcular, tanto as coordenadas de referência  $xy$  a cada instante  $t$ , quanto a velocidade de referência  $v_r$  do robô através das suas componentes  $v_x$  e  $v_y$ , que é dada por

$$v_r = \sqrt{v_x^2 + v_y^2} \quad (6.13)$$

Já a velocidade angular de referência  $\omega_r$  depende do tipo da trajetória. É nula para uma reta e constante para curva. Determinadas essas velocidades, a referência é dada, em um caso geral, em forma de um robô de referência (Kanayama *et. al.*, 1990 ):

$$\begin{aligned}\dot{\tilde{x}} &= v_r \cos \tilde{\theta} \\ \dot{\tilde{y}} &= v_r \sin \tilde{\theta} \\ \dot{\tilde{\theta}} &= \omega_r\end{aligned}\tag{6.14}$$

Para estabelecer uma trajetória de referência em forma de curva consideramos que essa trajetória possui uma forma circular de raio  $r$ . Define-se que o robô deva seguir um movimento de arco circular uniforme, com velocidade angular de referência  $\omega_r$  constante. Assim, as coordenadas da trajetória de referência são dadas pela integração do robô de referência (6.14).

$$\begin{aligned}\tilde{x} &= \int v_r \cos \tilde{\theta} dt \\ \tilde{y} &= \int v_r \sin \tilde{\theta} dt \\ \tilde{\theta} &= \int \omega_r dt\end{aligned}\tag{6.15}$$

A coordenada de referência  $\tilde{\theta}$  tem a seguinte expressão

$$\tilde{\theta} = \int \omega_r = \omega_r t + \tilde{\theta}_0\tag{6.16}$$

Inserindo o valor de  $\tilde{\theta}$  nas duas primeiras equações de (6.15) obtém-se as seguintes integrações:

$$\tilde{x} = \int \omega_r r \cos(\omega_r t + \tilde{\theta}) dt = r \sin(\omega_r t + \tilde{\theta}) + c_x$$

e

$$\tilde{y} = \int \omega_r r \sin(\omega_r t + \tilde{\theta}) dt = -r \cos(\omega_r t + \tilde{\theta}) + c_y\tag{6.17}$$

Um ponto importante é a determinação do sentido da trajetória de arco circular e da coordenada do centro de movimento. O centro de movimento pode ser determinado através

da relação (6.18) no momento inicial  $t=0$  da trajetória de referência, de acordo com as condições iniciais  $\tilde{x}_0 \tilde{y}_0 \tilde{\theta}_0$  dessa referência, ou seja,

$$c_x = \tilde{x}_0 - r \sin(\tilde{\theta}_0)$$

e

$$c_y = \tilde{y}_0 + r \cos(\tilde{\theta}_0) . \quad (6.18)$$

Ao determinar essa trajetória depara-se com, no mínimo, quatro situações de direção e sentido do movimento desejado. Primeiramente, uma curva de retorno do robô. Esse retorno pode significar uma circunferência para trás, no sentido horário ou anti-horário. No primeiro caso, como mostra a figura 6.2 deseja-se que o robô realize um movimento circular de retorno para esquerda no sentido anti-horário. Nesse caso, a velocidade angular será positiva e o ângulo inicial da trajetória de referência  $\tilde{\theta}_0$ , em relação às coordenadas no mundo, será perpendicular ao deslocamento angular inicial do robô  $\theta_0$  e poderá ser expresso em função deste, como demonstra a figura 6.2, como:

$$\tilde{\theta}_0 = \theta_0 + \frac{\pi}{2} \quad (6.19)$$

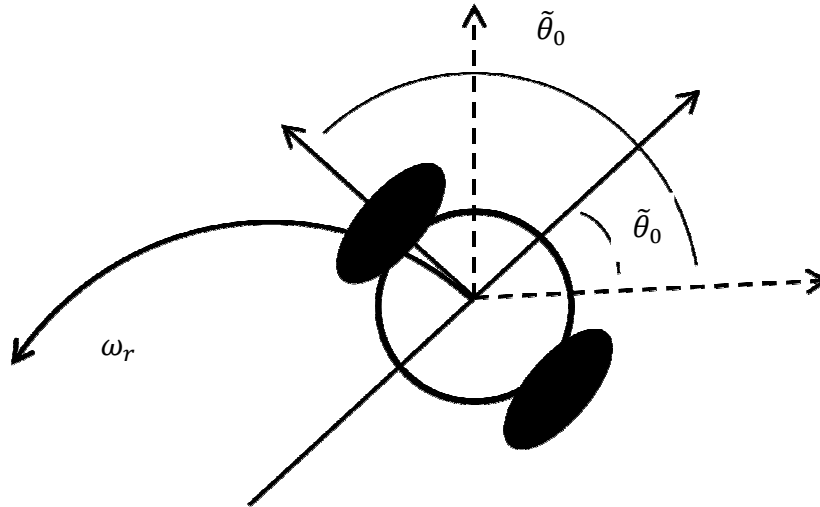


Figura 6. 2- Curva de Retorno no Sentido Anti-horário

No segundo caso, deseja-se um movimento de retorno para direita do robô, no sentido horário, como ilustra a figura 6.3:

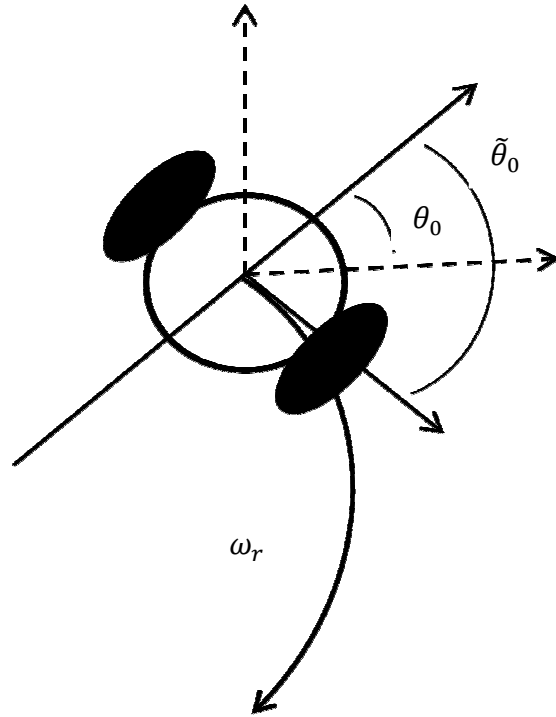


Figura 6. 3 - Curva de Retorno no Sentido Horário

Os ângulos  $\theta_0$  e  $\tilde{\theta}_0$ , na Figura 6.3, representam, respectivamente, o deslocamento angular inicial do robô em relação ao sistema de coordenadas no mundo e o ângulo inicial da trajetória de referência. A velocidade angular, nesse caso é negativa e o ângulo inicial da trajetória de referência pode ser calculado de (6.20), conforme ilustrado na figura 6.3:

$$\tilde{\theta}_0 = \theta_0 - \frac{\pi}{2} \quad (6.20)$$

Uma vez calculado o ângulo inicial da trajetória de referencia  $\tilde{\theta}_0$ , calcula-se o centro do movimento através da expressão (6.18).

Outra situação pode ser considerada, na qual o robô necessita realizar uma trajetória circular deslocando-se para a sua frente, chamada aqui de curva de desvio, que pode ser no sentido horário ou anti-horário. No caso curva de desvio pela direita, no sentido anti-horário,



conforme gráfico 6.4, a velocidade angular é positiva e o ângulo inicial da trajetória será perpendicular ao do robô e é dado por:

$$\tilde{\theta}_0 = \theta_0 - \frac{\pi}{2} \quad (6.21)$$

O centro do movimento então é calculado através de (6.18).

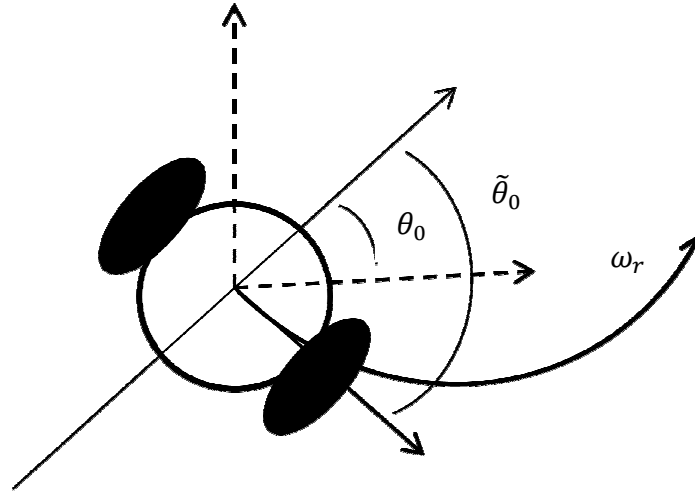


Figura 6. 4- Curva de Desvio no Sentido Anti-horário

No caso da curva de desvio, pela esquerda no sentido horário, a velocidade angular é negativa e o ângulo inicial da trajetória de referência é dado por, como se pode ver na figura 6.5:

$$\tilde{\theta}_0 = \theta_0 + \frac{\pi}{2} \quad (6.22)$$

Figura 6. 5- Curva de Desvio no Sentido Horário

O centro do movimento também é calculado através de (6.18). Em cada um desses casos, uma vez determinado o centro do movimento, a trajetória de referência a cada passo é dada pelo robô de referência. Os últimos dois casos representados pelas Figuras 6.4 e 6.5 são especialmente úteis no caso de desvio de um obstáculo interposto no caminho. Nesse caso, pode-se traçar uma trajetória de referência com um raio de segurança cujo arco de trajetória envolva o obstáculo.

### **6.3. Localização do Robô**

#### **6.3.1. Visão**

Para se localizar no ambiente é necessário que o robô conheça sua posição. Para isso é implementado um algoritmo de visão de duas câmeras que localiza o robô no ambiente através de dois pontos de referência cujas coordenadas são absolutas. O processo detalhado de obtenção desses ângulos é descrito no capítulo 3.

### **6.3.2. Estimação de estado do robô através da odometria**

A estimação da coordenada do robô através da odometria é um processo comumente utilizado, no qual uma leitura de *encoder* é inserida no modelo de odometria estimando-se, dessa forma, a posição corrente do robô.

A utilização da odometria nessa aplicação é descrita no capítulo 3 bem como o cálculo de erros associados a esse processo. Como resultado, obtém-se o estado estimado a priori do robô com uma covariância  $W$  associada ao erro. Esses parâmetros são utilizados posteriormente no filtro de Kalman para otimizar as estimativas de posição obtidas pelas medições de câmeras e odometria.

## **6.4. Navegação do robô**

### **6.4.1. Navegação com Filtro de Kalman**

As estimações do estado do robô sejam através das câmeras, seja por odometria, carregam uma informação com ruído e/ou erros sistemáticos. Os erros da odometria, por exemplo, se tornam cumulativos com o avanço do robô no ambiente, já a visão robótica produz medições com grande erro quanto mais se afasta das referências de orientação (*landmarks*). Para assegurar a confiabilidade às medições em todos os momentos da navegação torna-se necessária a implementação de um estimador de estado. Nesse caso é utilizado o filtro de Kalman que possibilita tratamento estocástico do processo de estimação de estados.

O filtro de Kalman, como já visto, divide o processo de estimação em duas fases a predição e a correção. No processo de predição estima-se um estado a priori e na correção atualiza-se o estado estimado através de uma ponderação entre as covariâncias do processo estimado e da medição proveniente das câmeras. A seguir detalha-se a implementação do filtro de Kalman.

Deseja-se realizar o estimador para um sistema discreto controlado:

$$\hat{x}_{k+1} = \bar{A}\hat{x}_k + \bar{B}u_k + \bar{B}w_k$$

$$z_k = H\hat{x}_k + \gamma_k, \quad (6.23)$$

onde  $\hat{x}_k$  é o vetor de estado  $\hat{x}_k = [\hat{x}_1 \quad \hat{x}_2 \quad \hat{x}_3]^T$  de coordenadas estimadas. A dinâmica do sistema está sujeita a perturbações aleatórias  $w_k$  e as medições  $z_k$  estão sujeitas a ruído aleatório  $\gamma_k$ . Consideram-se ambos os erros como estatisticamente independentes. O controle  $u_k = [u_1 \quad u_2]^T$  provém do sistema de controle de trajetória e as matrizes  $\bar{A}$  e  $\bar{B}$  têm os valores:

$$\bar{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \bar{B} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \text{ e } H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.24)$$

Na formulação tradicional do filtro de Kalman a covariância do erro é constante. No caso atual considera-se a covariância do erro  $w_k$  variável com a matriz  $\bar{B}$ , matriz de controle do sistema de estimação e definida como:

$$W = E[\bar{B} \cdot w_k \cdot (\bar{B} \cdot w_k)^T], \quad (6.25)$$

onde  $w_k = \begin{bmatrix} \bar{v}_k \\ \bar{\omega}_k \end{bmatrix}$  representa o erro sistemático devido ao deslocamento na odometria.

De acordo com o processo descrito na seção 3.1 consideramos o erro  $w_k$  como proveniente do processo de estimação de estado através da odometria:

$$w_k = \begin{bmatrix} \bar{v}_k \\ \bar{\omega}_k \end{bmatrix} = \begin{bmatrix} (\bar{N}_d + \bar{N}_e) \frac{\pi r}{N_{res}} \\ (\bar{N}_d - \bar{N}_e) \frac{2\pi r}{b \cdot N_{res}} \end{bmatrix} \quad (6.26)$$

A covariância  $W$  desse erro (6.26) é dada por:

$$W = E[\bar{B} \cdot w_k \cdot (\bar{B} \cdot w_k)^T] =$$

$$= E \left[ \begin{pmatrix} \bar{v}_k \cdot \cos(\theta_k) \\ \bar{v}_k \cdot \sin(\theta_k) \\ \bar{\omega}_k \end{pmatrix} \cdot (\bar{v}_k \cdot \cos(\theta_k) \quad \bar{v}_k \cdot \sin(\theta_k) \quad \bar{\omega}_k) \right], \quad (6.27)$$

ou seja :

$$W = E \begin{bmatrix} \bar{v}_k^2 \cdot \cos^2(\theta_k) & \bar{v}_k^2 \cdot \sin(\theta_k) \cdot \cos(\theta_k) & \bar{\omega}_k \cdot \bar{v}_k \cdot \cos(\theta_k) \\ \bar{v}_k^2 \cdot \sin(\theta_k) \cdot \cos(\theta_k) & \bar{v}_k^2 \cdot \sin^2(\theta_k) & \bar{\omega}_k \cdot \bar{v}_k \cdot \sin(\theta_k) \\ \bar{\omega}_k \cdot \bar{v}_k \cdot \cos(\theta_k) & \bar{\omega}_k \cdot \bar{v}_k \cdot \sin(\theta_k) & \bar{\omega}_k^2 \end{bmatrix}. \quad (6.28)$$

Os valores  $\bar{v}_k^2$ ,  $\bar{\omega}_k^2$  e  $\bar{v}_k \cdot \bar{\omega}_k$  podem ser definidos como:

$$\bar{v}_k^2 = \left[ (\bar{N}_d + \bar{N}_e) \frac{\pi r}{N_{res}} \right]^2 = \left( \frac{\pi r}{N_{res}} \right)^2 \cdot (\bar{N}_d^2 + 2\bar{N}_d \bar{N}_e + \bar{N}_e^2), \quad (6.29)$$

$$\bar{\omega}_k^2 = \left[ (\bar{N}_d - \bar{N}_e) \frac{2\pi r}{b \cdot N_{res}} \right]^2 = \left( \frac{2\pi r}{b \cdot N_{res}} \right)^2 \cdot (\bar{N}_d^2 - 2\bar{N}_d \bar{N}_e + \bar{N}_e^2) \sim$$

e

$$\bar{v}_k \cdot \bar{\omega}_k = \frac{2}{b} \left( \frac{\pi r}{N_{res}} \right)^2 \cdot (\bar{N}_d^2 - \bar{N}_e^2) \quad (6.30)$$

A hipótese essencial é que os erros em número de passos no encoder  $\bar{N}_d$  e  $\bar{N}_e$  são independentes entre si e possuem distribuição normal, de forma que :

$$E[\bar{N}_d^2] = E[\bar{N}_d \cdot \bar{N}_d^T] = \sigma^2,$$

$$E[\bar{N}_e^2] = E[\bar{N}_e \cdot \bar{N}_e^T] = \sigma^2$$

e

$$E[\bar{N}_d \cdot \bar{N}_e] = 0 \quad (6.31)$$

Tendo –se em mente (6.31) e substituindo (6.30) em (6.28) tem-se:

$$W = \begin{bmatrix} 2.\sigma^2.\left(\frac{\pi r}{N_{res}}\right)^2.\cos^2(\theta_k) & 2.\sigma^2.\left(\frac{\pi r}{N_{res}}\right)^2.\sin(\theta_k).\cos(\theta_k) & 0 \\ 2.\sigma^2.\left(\frac{\pi r}{N_{res}}\right)^2.\sin(\theta_k).\cos(\theta_k) & 2.\sigma^2.\left(\frac{\pi r}{N_{res}}\right)^2.\sin^2(\theta_k) & 0 \\ 0 & 0 & 2.\sigma^2.\left(\frac{2\pi r}{bN_{res}}\right)^2 \end{bmatrix}, \quad (6.32)$$

que pode ser reescrito finalmente como:

$$W = 2.\sigma^2.\left(\frac{\pi r}{N_{res}}\right)^2 \begin{bmatrix} \cos^2(\theta_k) & \sin(\theta_k).\cos(\theta_k) & 0 \\ \sin(\theta_k).\cos(\theta_k) & \sin^2(\theta_k) & 0 \\ 0 & 0 & \left(\frac{2}{b}\right)^2 \end{bmatrix}. \quad (6.33)$$

Com relação ao ruído  $\gamma_k$  oriundo da medição do estado  $z_k$  através da visão, a sua covariância  $V$  é dada através da equação (6.34) calculada pelo processo explicado na seção 3.2. A matriz  $\bar{V}$  tem a forma:

$$\bar{V} = \begin{bmatrix} D \\ C \end{bmatrix} E \begin{bmatrix} D^T & C^T \end{bmatrix}, \quad (6.34)$$

onde:

$$E = \bar{\sigma}^2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = -\frac{1}{2} \left[ \frac{\partial R^T}{\partial Z_\theta} (x_{caP} + x_{caA} + 2x_a)C + R^T [B_P \quad B_A] \right]$$

e

$$C = \frac{\cos^2(\beta)}{x_{rA} - x_{rP}} [\tan(\beta) \quad 1 \quad -\tan(\beta) \quad -1]. \quad (6.35)$$

O ângulo  $\beta$  é calculado por  $\beta = \arctan\left(\frac{y_{rP} - y_{rA}}{x_{rA} - x_{rP}}\right)$ , onde  $x_{rA}, y_{rA}$  e  $x_{rP}, y_{rP}$  são as coordenadas dos pontos  $P$  e  $A$  vistos pelo robô.

A estimação do estado pelo do filtro de Kalman é realizada através de dois estágios distintos: a predição (*prediction*) e correção (*update*).

Na predição acontece a estimação *a priori* do estado  $\hat{x}_k^-$  realizada através de medições de encoder e equações de odometria

$$\hat{x}_{k+1}^- = \bar{A}\hat{x}_k^- + \bar{B}u_k \quad , \quad (6.36)$$

onde  $\bar{A}$  e  $\bar{B}$  são matrizes dadas por (6.24), e  $u_k$  o controle.

A covariância do estado estimado a priori é calculada por :

$$P_{k+1}^- = \bar{A}P_k^- \bar{A}^T + W \quad , \quad (6.37)$$

onde  $W$  foi definido anteriormente e é dado por (6.33).

A fase de correção inicia com o cálculo da matriz de ganho de Kalman  $K_k$  que depende das covariâncias do estado a priori e da covariância do estado medido pelas câmeras:

$$K_k = P_k^- H^T (H P_k^- H^T + \bar{V})^{-1} \quad (6.38)$$

Introduzindo (6.38) na equação seguinte tem-se o cálculo de estado estimado *a posteriori* que corresponde ao estado ótimo estimado pelo filtro de Kalman.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \quad , \quad (6.39)$$

onde  $z_k$  é o estado disponibilizado pela medição através da visão.

A covariância desse estado estimado *a posteriori* pode ser calculada como

$$P_k = (I - K_k H) P_k^- + W \quad (6.40)$$

#### 6.4.2. – Estratégias de Navegação

Esse é o modulo do sistema de navegação que lida propriamente com a regulação da trajetória que o robô executa. Nesse sistema são consideradas duas situações diferentes que

requerem dois tipos de controle diferentes. É considerado um modelo de robô discreto e para o controle de tal modelo é considerado um controle discreto e outro modelo contínuo para qual é aplicado um controle de tempo contínuo. Em ambas as situações, assim como na prática, as medições de estado do robô são disponibilizadas para o sistema de forma discreta no tempo. Além disso, devido ao tempo que é necessário para a captura e processamento dessa informação de estado, o tempo de espera entre as capturas pode ser bastante elevado (superior a 10 segundos).

A integração entre o controle de trajetória do robô e as medições de estado realizadas pela odometria e pela visão de duas câmeras é realizado pelo filtro de Kalman que lida com os erros sistemáticos das medições, otimizando a estimação de estado.

#### 6.4.2.1. Estratégia Mista

Nessa abordagem considera-se que o robô e o controle são calculados de forma contínua através do controle SDRE contínuo descrito na seção 5.2. Porém, as medições sobre a localização do robô são disponíveis de forma discreta. Isso resulta que em pequenos intervalos do tempo o robô é estabilizado a uma trajetória estimada através do controle contínuo, reiniciando uma nova estabilização a cada momento discreto no qual se disponibiliza uma estimação de estado.

O modelo do robô utilizado é contínuo no tempo e tem a seguinte forma:

$$\begin{aligned}\dot{e}_1 &= u_2 + e_2 \omega_r - u_1 e_2 \\ \dot{e}_2 &= v_r \sin e_3 - \omega_r e_1 + e_1 u_1 \\ \dot{e}_3 &= u_1\end{aligned}\tag{6.41}$$

A estimativa definitiva de estado é dada pelo filtro de Kalman. A trajetória do robô, portanto, consiste de várias trajetórias contínuas em trechos discretos. Os pontos de união desses trechos são aqueles onde há disponibilidade de medições.

O robô, portanto, inicia sua trajetória num ponto  $x_0$  e utiliza a visão para a estimação de sua localização. Uma vez que os pontos  $A$  e  $P$  que servem de referência para a localização encontram-se mais ao final da trajetória essa estimação do estado do robô



inicial é dotada de um grande erro. Define-se, posteriormente, o ponto final do trecho da trajetória de referência, com base na orientação dessa ao alvo  $A$ , e o módulo de controle então calcula uma estimacão a priori do quanto o robô se deslocaria do ponto inicial estimado ao longo dessa trajetória de referência estimada. Ao final desse calculo, é estimado um estado a priori  $x^-$ , ao qual se adiciona um erro calculado por (6.26):

$$w_k = \begin{bmatrix} \bar{v}_k \\ \bar{\omega}_k \end{bmatrix} = \begin{bmatrix} (\bar{N}_d + \bar{N}_e) \frac{\pi r}{N_{res}} \\ (\bar{N}_d - \bar{N}_e) \frac{2\pi r}{b.N_{res}} \end{bmatrix}, \quad (6.26)$$

que representa o erro no deslocamento do robô da posição  $x_0$  a posição  $x_k$  devido à imprecisão da tradução do comando de controle  $u$  nos atuadores das rodas.

Ao mesmo tempo, o mesmo controle  $u$  é aplicado à coordenada real do robô para simular a sua movimentação efetiva no espaço. Além disso, uma medição de coordenadas é obtida pelas câmeras.

O estado estimado a priori, com covariância  $W$  associada é comparado com a medição pela visão  $Z$  com covariância associada  $V$ . A equação do filtro do Kalman, dada pela equação (6.39):

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \quad (6.39)$$

É usada para se encontrar o estado estimado pelo filtro (a posteriori) e a covariância  $P$  desse estado, dado anteriormente na equação:

$$P_k = (I - K_k H) P_k^- + W \quad (6.40)$$

O estado (6.39) é considerado como a estimativa de posição atual do robô e o início de uma nova trajetória de referência para próximo intervalo de controle e assim o processo é repetido sucessivamente.

#### 6.4.2.2. Estratégia Discreta

O modelo na forma discreta foi descrito no capítulo 5 e tem a forma:

$$\begin{aligned}x_{k+1} &= x_k + v_k \cos \theta_k \\y_{k+1} &= y_k + v_k \sin \theta_k \\ \theta_{k+1} &= \theta_k + \omega_k\end{aligned}\tag{6.42}$$

Essa abordagem utiliza-se de um método de controle discreto descrito na seção 5.3 do capítulo 5 e o robô de referência é determinada através da equação (5.76). O robô inicia o cálculo de sua trajetória no ponto estimado pela visão que possui um erro associado de covariância  $V$ . O estado do robô é medido pela câmera e estimado pelas informações da odometria. A fusão dessas informações para a determinação de estimativa do estado do robô é realizada através do filtro de Kalman. O processo ocorre recursivamente segundo o algoritmo seguinte:

- Cálculo da referência do robô
- Realização do controle do sistema
- Estimação de posição do robô pelas câmeras e pela odometria.
- Cálculo do ganho de Kalman com a ponderação de covariâncias das duas medições. Com base nisso calcula-se o estado estimado a posteriori que é considerado como o estado de localização do robô. Repete-se sucessivamente esse processo.

As simulações das estratégias descritas nesse capítulo são encontrados na terceira parte do capítulo seguinte . Este próximo capítulo é dedicado a todas as simulações numéricas relevantes desse trabalho como as trajetórias controladas com os métodos de controle apresentados no capítulo 5 e as medições de estado através de visão de duas câmeras e odometria, apresentados no capítulo 4.

## Capítulo 7

### Simulações Computacionais

Nesse capítulo são apresentadas as simulações numéricas desse trabalho. O capítulo se divide em três partes principais: simulações dos métodos de controle para o modelo matemático do robô móvel, simulações da estimação de estado pela visão estereoscópica e odometria, bem como a influência das imprecisões dessa estimação na trajetória do robô e, por fim, a última parte é dedicada às simulações de navegação do robô integrando todos os módulos acima citados.

#### 7.1. Simulações para métodos de controle

Esta seção é dedicada à apresentação de simulações numéricas obtidas para diferentes modelos descritos no capítulo 2 e métodos de controle descritos no capítulo 5. O problema de controle considerado é o de *tracking* a uma trajetória de referência. As trajetórias de referência consideradas são a reta e o círculo. As simulações são efetuadas no ambiente MATLAB™. Para resolução de equações diferenciais é utilizado o método de Runge Kutta de quarta ordem.

##### 7.1.1. Simulações para o método de controle ótimo por realimentação de estado linear

No capítulo 5 é formulado o problema de controle ótimo por realimentação de estado linear ótimo para o sistema de Brockett:

$$\begin{aligned}\dot{z}_1 &= U_1 \\ \dot{z}_2 &= U_2 \\ \dot{z}_3 &= z_2 U_1 - z_1 U_2\end{aligned}, \quad (7.1)$$

onde:

$$\begin{aligned}U_1 &= \omega \\ U_2 &= v - \omega(x \sin \theta - y \cos \theta)\end{aligned} \quad (7.2)$$

Esse sistema é levado a coordenadas em desvios através do processo descrito na seção 4.1 e sua forma torna-se:

$$\begin{aligned}\dot{e}_1 &= u_1, \\ \dot{e}_2 &= u_2, \\ \dot{e}_3 &= \tilde{u}_2 e_1 - \tilde{u}_1 e_2 + (\tilde{z}_1 + e_1)u_2 - (\tilde{z}_2 + e_2)u_1.\end{aligned} \quad (7.3)$$

O sistema (7.3) pode ser reescrito na seguinte forma matricial:

$$\dot{e} = A(t)e + h(e, u) + B(t)u, \quad (7.4)$$

onde os valores das matrizes são:

$$A(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{u}_2 & -\tilde{u}_1 & 0 \end{bmatrix}, h(e, u) = \begin{bmatrix} 0 \\ 0 \\ e_1 u_2 - e_2 u_1 \end{bmatrix}, B(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\tilde{z}_2 & \tilde{z}_1 \end{bmatrix}, \quad (7.5)$$

e a trajetória de referência em coordenada de desvios tem a forma:

$$\tilde{z}(t) = \begin{bmatrix} \frac{\pi}{4} \\ \sqrt{2}t \\ -\frac{\pi\sqrt{2}}{4}t \end{bmatrix}, \quad (7.6)$$

obtida através da transformação :

$$\tilde{z} = \tilde{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\tilde{\theta}) & \sin(\tilde{\theta}) \\ 0 & 2\sin(\tilde{\theta}) - \tilde{\theta}\cos(\tilde{\theta}) & -2\cos(\tilde{\theta}) - \tilde{\theta}\sin(\tilde{\theta}) \end{bmatrix} \tilde{x}. \quad (7.7)$$

A trajetória de referência em coordenadas de posição e orientação é:

$$\tilde{x}(t) = \begin{bmatrix} \frac{\pi}{4} \\ t \\ t \end{bmatrix}, \quad (7.8)$$

A figura 7.1 representa uma trajetória de deslocamento  $xy$  do robô, para condições iniciais:

$$\theta_0 = -\pi/4, x_0 = 1, y_0 = 0.$$

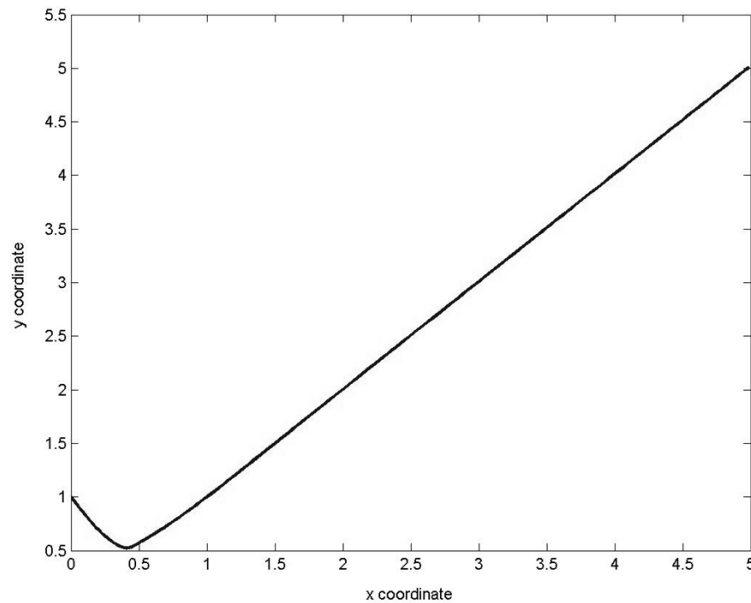


Figura 7. 1 -Trajetória do Robô, Referência Reta, Método: Linear Por realimentação de estado Ótimo

A figura 7.2 mostra a evolução das coordenadas em erro do sistema para os parâmetros acima mencionados.

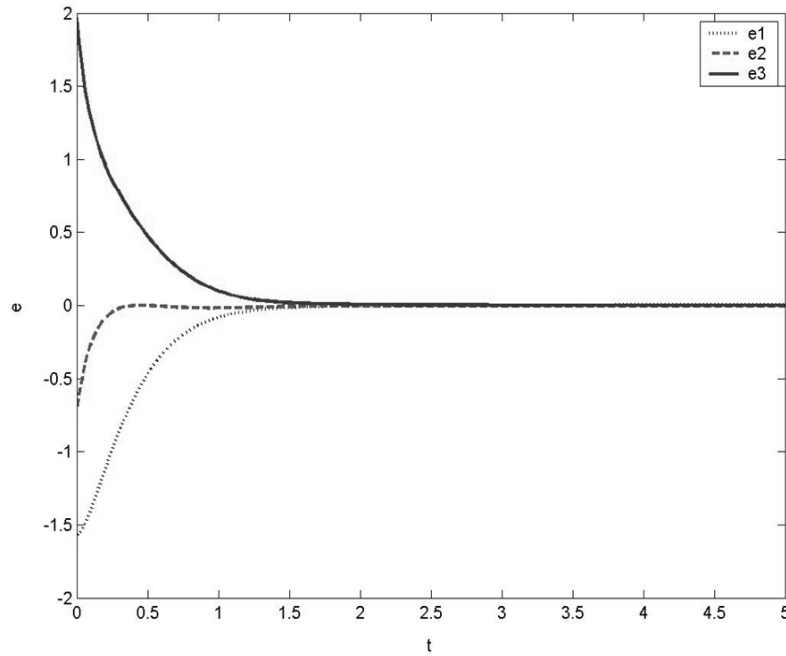


Figura 7. 2- Trajetória em Desvios , Referência reta, Método : Linear Por realimentação de estado Ótimo.

Dessas figuras é possível observar que o sistema leva cerca de 1,5 segundos para convergir à trajetória desejada o que é um tempo bastante rápido.

Considerando agora uma trajetória de referência circular, define-se uma referência na forma (7.9) de raio  $r = 2$ :

$$\begin{aligned}\tilde{\theta} &= \tilde{\omega}t, \\ \tilde{x} &= r\cos\tilde{\theta}, \\ \tilde{y} &= r\sin\tilde{\theta},\end{aligned}\tag{7.9}$$

onde  $\tilde{\omega} = 0.5$  rad/s e é a velocidade angular da trajetória de referência.

Na simulação, ilustrada pela figura 7.3, para as condições iniciais  $\theta_0 = 0, x_0 = 1, y_0 = 0$ , a trajetória do sistema (linha sólida) estabiliza em torno da trajetória desejada (linha em hachuras).

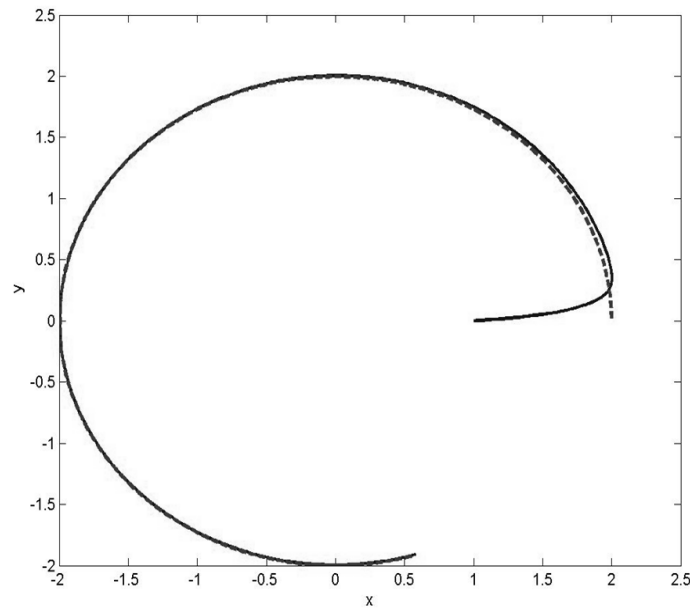


Figura 7. 3-Trajetória Circular para Parâmetros  $\theta_0 = 0, x_0 = 1, y_0 = 0$ , raio=2

As simulações representadas pelas figuras de 7.1 a 7.3 demonstram a eficácia do método de controle linear aplicado a um sistema não-linear e subatuado. No entanto esse método é não é adequado para aplicações online. Por isso, a seguir, são realizadas simulações usando o método SDRE.

#### 7.1.2. Resultados com o método de controle *SDRE-State Dependent Riccati Equation*

Uma das principais vantagens desse método é a possibilidade de sua implementação *online*, permitindo dessa forma, o cálculo do controle  $u$  a cada iteração ou passo do robô.

Dois exemplos de simulações com esse método estão descritos a seguir, ambos para o modelo de Brockett em coordenadas de desvio mostrados na seção anterior (equação 7.3). Na primeira simulação a trajetória de referência é um círculo de raio  $\tilde{r} = 3$ ,

a velocidade desejada com a qual o robô deve percorrer a trajetória é  $\tilde{\omega} = 0.5$  rad/s, em um intervalo de tempo de 0 a 10 segundos.

As condições iniciais do sistema são:  $\theta_0 = -\pi/4, x_0 = 0, y_0 = 0$ .

A figura 7.4 representa a trajetória de fase do sistema. A linha sólida representa a trajetória de referência e a linha pontilhada, a trajetória do robô móvel. Nota-se que, para essas condições iniciais, a trajetória do robô converge perfeitamente para a trajetória desejada. A figura 7.5 que demonstra a trajetória em desvios do sistema e confirma a rápida convergência, onde após menos de 2 segundos todas as trajetórias em desvios tendem a zero.

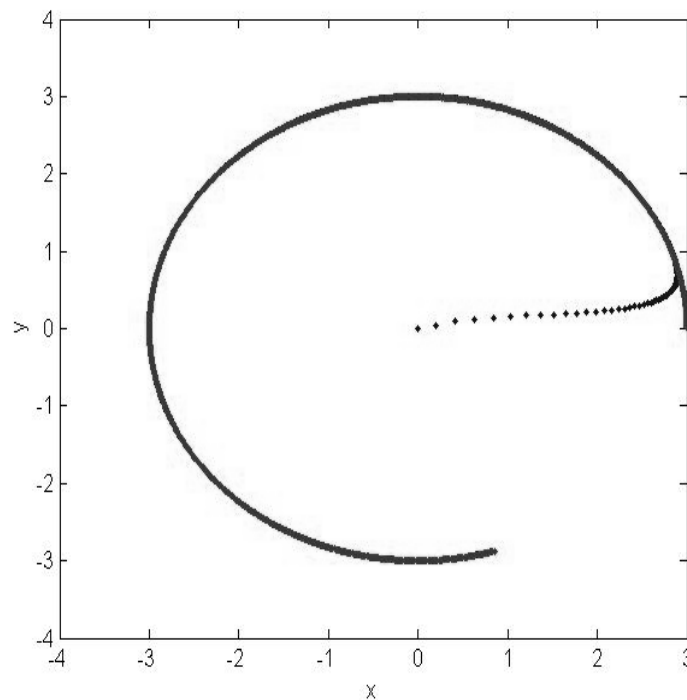


Figura 7. 4-Trajétória do Sistema para uma Trajetória de Referência Circular, Controle SDRE



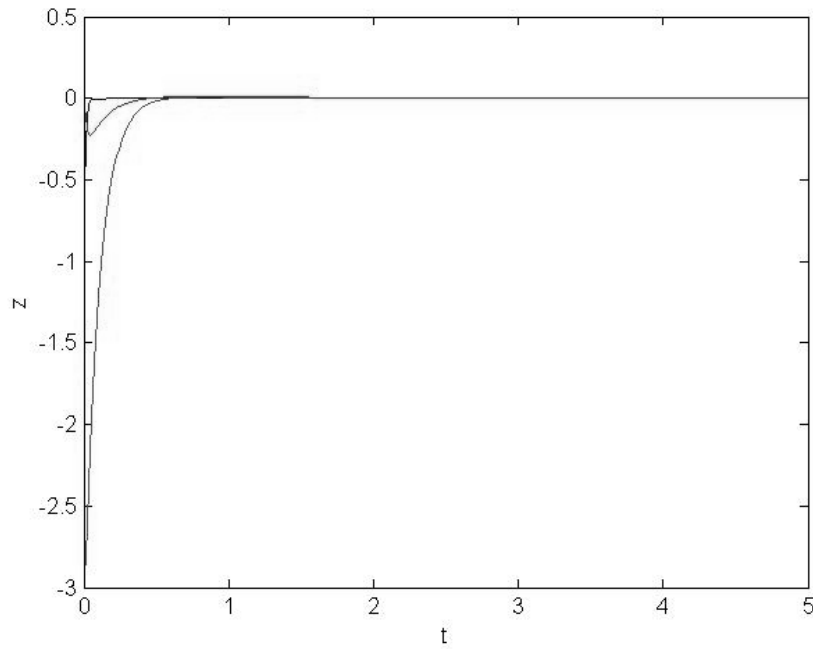


Figura 7. 5-Trajétória em Desvios, Controle SDRE

No segundo exemplo da simulação a trajetória de referência é uma reta na forma (7.10):

$$\tilde{x}(t) = \begin{bmatrix} \frac{\pi}{4} \\ t \\ t \end{bmatrix} \quad (7.10)$$

O sistema parte de condições iniciais  $\theta_0 = 0, x_0 = -2, y_0 = 1$  e evolui no tempo de 0 a 10 segundos. O gráfico da figura 7.6 mostra a trajetória de fase do sistema, sendo a curva de linha sólida a própria trajetória do sistema e a reta pontilhada, trajetória de referência. Novamente o sistema estabiliza rapidamente em torno da trajetória desejada.

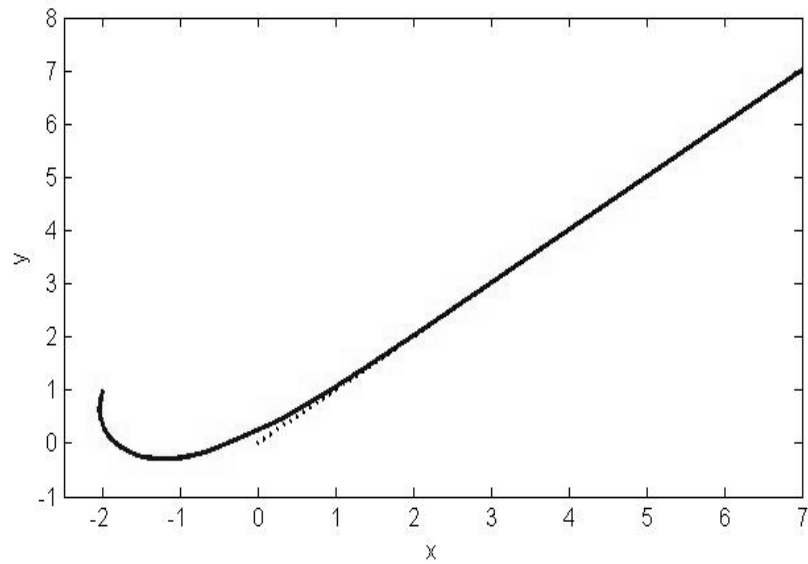


Figura 7. 6- Trajetória do Sistema Evoluindo para Trajetória de Referência Reta, Controle SDRE

A figura 7.7 demonstra a evolução temporal das coordenadas em desvio do sistema que representam as diferenças entre a trajetória do sistema e a trajetória de referência. O gráfico abaixo mostra a minimização desses desvios com tendência a zero após 2 segundos.

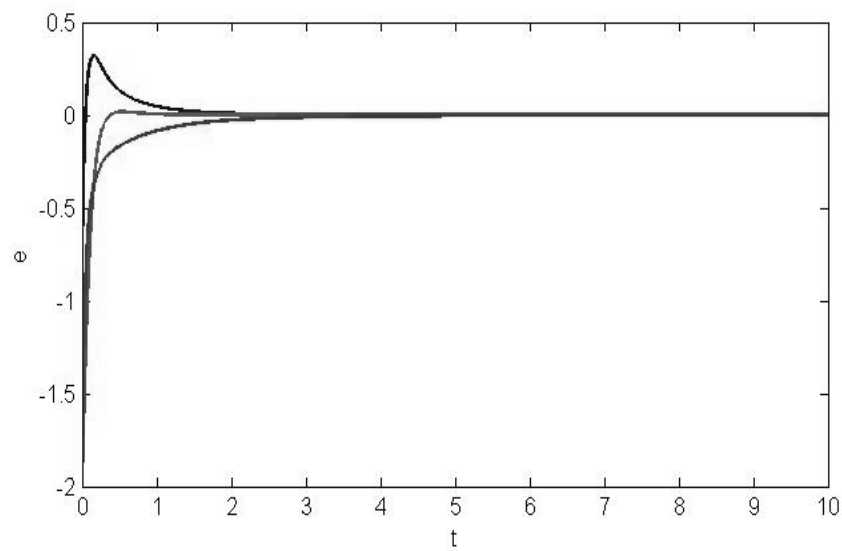


Figura 7. 7- Trajetória em Desvios, Controle SDRE

### 7.1.3. Comparação dos dois métodos

Para apresentar a comparação do funcionamento dos dois métodos escolhe-se a trajetória de referência como:

$$\tilde{z}(t) = \begin{bmatrix} \pi/4 \\ t \\ \pi t/4 \end{bmatrix} \quad (7.11)$$

Os valores das matrizes Q e R são, para ambos os métodos:

$$Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (7.12)$$

A trajetória inicia nas simulações nas condições iniciais:  $x_0 = \begin{bmatrix} -\pi/2 \\ -3\pi/4 - 1/2 \\ 3 \end{bmatrix}$

Na figura 7.8 tem-se o gráfico de coordenadas em desvio para o método de por realimentação de estado ótimo, para os quais as matrizes do sistema têm os seguintes valores.

$$A(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, h(e, u) = \begin{bmatrix} 0 \\ 0 \\ e_1 u_2 - e_2 u_1 \end{bmatrix}$$

e

$$B(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -t & \frac{\pi}{4} \end{bmatrix}. \quad (7.13)$$

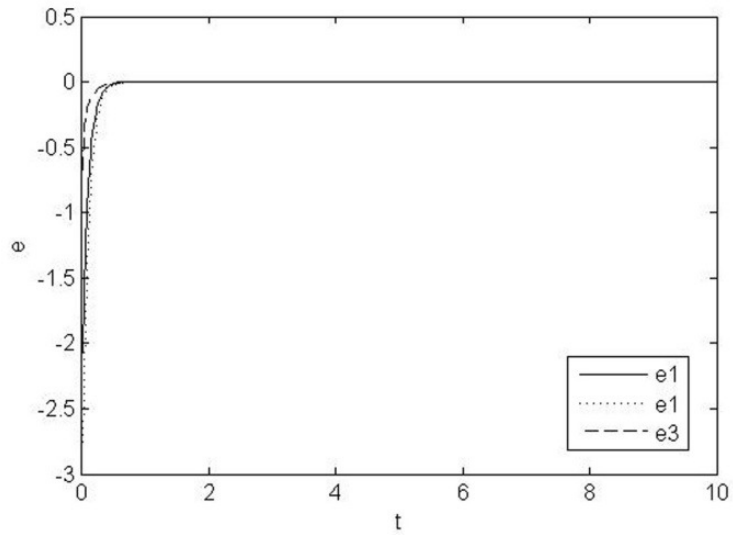


Figura 7. 8- Evolução Temporal das Coordenadas em Desvio, Controle por Realimentação de Estado Linear Ótimo

Para o método SDRE as matrizes do sistema têm os seguintes valores:

$$A(y) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \text{ e } B(y) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -e_2 - t & e_1 + \pi/4 \end{bmatrix}. \quad (7.14)$$

A figura 7.9 mostra a evolução temporal das coordenadas em desvio para o método de controle SDRE.

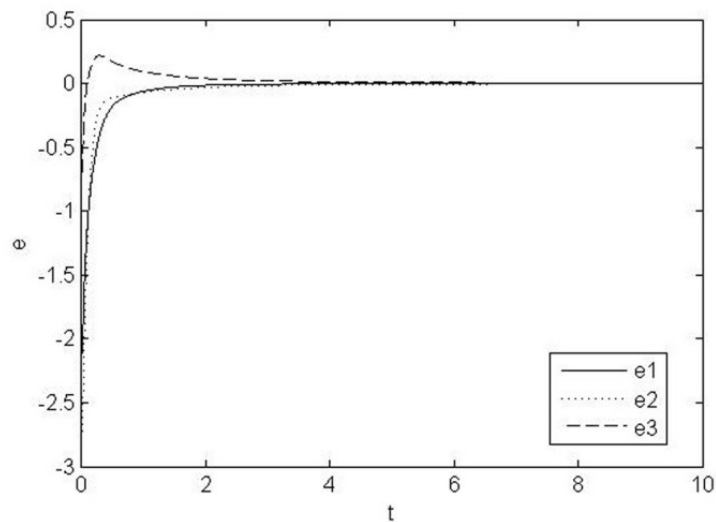


Figura 7. 9- Evolução Temporal das Coordenadas em Desvio, Controle por SDRE

A figura 7.10 mostra trajetórias  $xy$  que o sistema segue controlado pelos dois métodos diferentes. A linha sólida representa a trajetória obtida com o método de por realimentação de estado linear ótimo, enquanto a linha pontilhada representa a trajetória obtida pelo método SDRE.

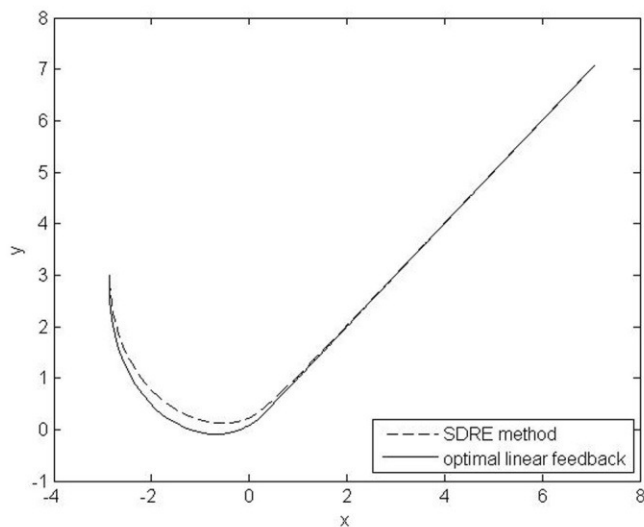


Figura 7. 10- Trajetória do Robô Obtida pelos Dois Métodos

Os dois métodos funcionam adequadamente para o sistema, levando o robô à trajetória de referência, como é ilustrado na figura 7.10. O método de por realimentação de estado ótimo apresenta, nesse caso, uma leve vantagem em relação ao SDRE, em termos de desempenho.

#### 7.1.4. Simulação para o Controle SDRE Discreto

Quando se considera um sistema de localização do robô, as medições de estado proveniente de sensores são disponibilizados de forma discreta no tempo e por vezes com um intervalo significativo entre cada medição. Por isso, é considerado um modelo de robô discreto e um problema de controle não- linear discreto no tempo na seção 5.3, com base na idéia do controle SDRE. A simulação desse problema é descrita a seguir.

O modelo considerado é dado pela equação :

$$\begin{aligned}x_{k+1} &= x_k + v_r \cos \theta_k \\y_{k+1} &= y_k + v_r \sin \theta_k. \\ \theta_{k+1} &= \theta_k + \omega_r\end{aligned}\tag{7.15}$$

A trajetória de referência inicia no ponto  $x_r(0) = [0; 1; \pi/4]$  e tem a forma :

$$\begin{aligned}\tilde{x}_{k+1} &= x_k + v_r \cos \tilde{\theta}_k \\ \tilde{y}_{k+1} &= y_k + v_r \sin \tilde{\theta}_k, \\ \tilde{\theta}_{k+1} &= \tilde{\theta}_k + \omega_r\end{aligned}\tag{7.16}$$

onde  $v_r = 1$  e  $\omega_r = 0$ ; o intervalo de simulação é de 40 segundos. Já a condição inicial da trajetória controlada é de  $x_0 = [2; 5; 0]$

Na figura 7.11 a trajetória controlada é mostrada em azul e a trajetória de referência em verde. A trajetória controlada converge para trajetória de referência, na totalidade, embora não se obtenha convergência imediata em cada a cada iteração, como mostra a figura 7.12. Esta figura ilustra a evolução temporal de cada coordenada em desvio controlado. Percebe-se que a estabilização completa do erro ocorre por volta do instante 30, embora a estabilização da coordenada angular dada por  $e_3$  seja mais rápida.

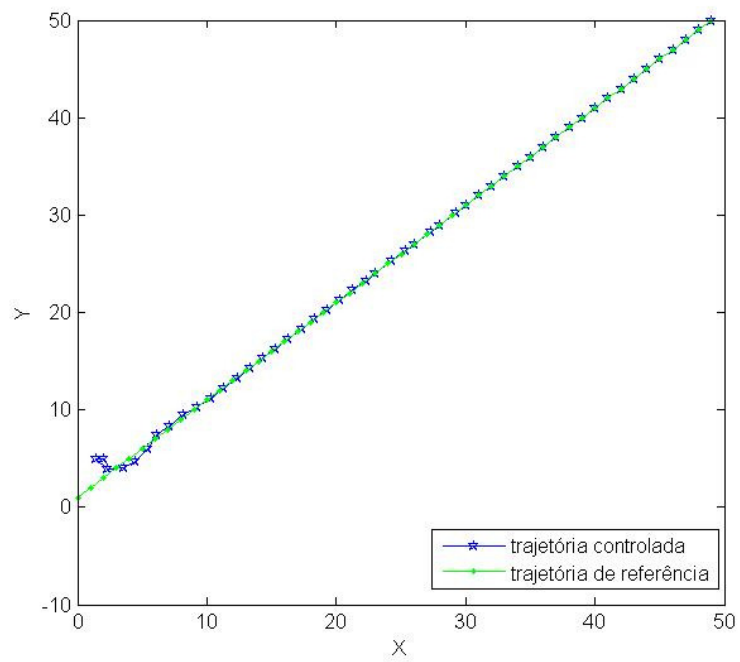


Figura 7. 11- Trajetória Controlada por SDRE Discreto no Tempo

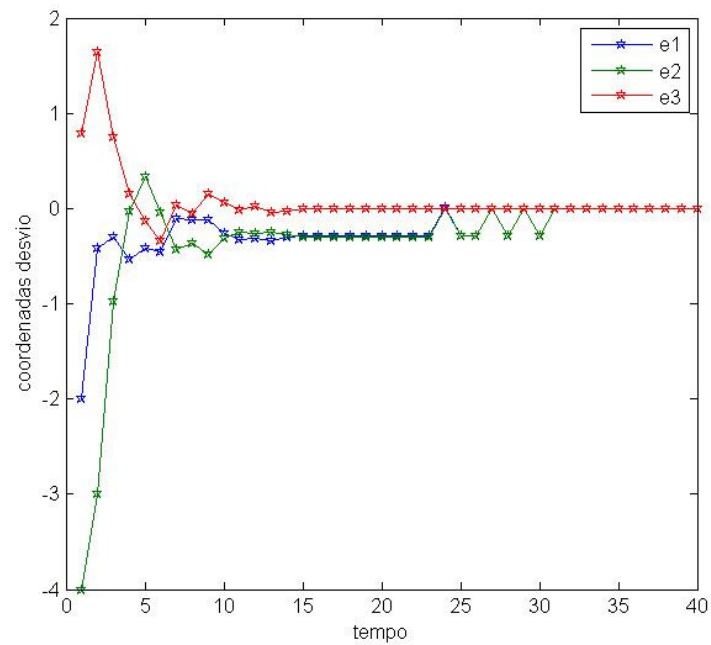


Figura 7. 12- Coordenadas em Desvios Controladas por SDRE Discreto no Tempo

### 7.1.5. Controle SDRE Contínuo no Tempo

Realizam-se, nessa seção, as simulações para o modelo de robô alternativo para o qual é formulado um problema de controle SDRE na seção 5.2. Esse modelo, contínuo no tempo, tem a forma:

$$e = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & \frac{v_r \sin(e_3)}{e_3} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} -e_2 & 1 \\ e_1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (7.17)$$

que é obtido pela transformação (7.18) do modelo cinemático (2.4) dado no capítulo 2:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = T \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} - x \\ \tilde{y} - y \\ \tilde{\theta} - \theta \end{bmatrix} \quad (7.18)$$

A trajetória de referência  $(\tilde{x} \ \tilde{y} \ \tilde{\theta})^T$  aqui considerada tem a forma cúbica, parametrizada pelo tempo e obtida na seção 6.2. Essa trajetória calcula as coordenadas de posição  $x$  e  $y$  de referência, a cada instante de  $t$ , como:

$$\begin{aligned} \tilde{x}(t) &= a_1 t^3 + b_1 t^2 + c_1 t + d_1, \\ \tilde{y}(t) &= a_2 t^3 + b_2 t^2 + c_2 t + d_2, \end{aligned} \quad (7.19)$$

onde  $c_1$  e  $c_2$  são as velocidades iniciais componentes de  $v_r$ , em  $x$  e  $y$ , respectivamente. Já  $d_1$  e  $d_2$  são as coordenadas  $x$  e  $y$  do ponto inicial da trajetória de referencia. As velocidades  $v_x$  e  $v_y$  componentes de  $v$ , a cada instante de  $t$  como:

$$\begin{aligned} v_x &= \dot{\tilde{x}}(t) = 3a_1 t^2 + 2b_1 t + c_1, \\ v_y &= \dot{\tilde{y}}(t) = 3a_2 t^2 + 2b_2 t + c_2. \end{aligned} \quad (7.20)$$

Essas velocidades são utilizadas no cálculo do robô de referência na forma:



$$\begin{aligned}\tilde{x} &= v_r \cos \tilde{\theta}, \\ \tilde{y} &= v_r \sin \tilde{\theta}, \\ \tilde{\theta} &= \omega_r.\end{aligned}\tag{7.21}$$

onde :

$$v_r = \sqrt{v_x^2 + v_y^2} \text{ e } \omega_r = 0.\tag{7.22}$$

O método de controle SDRE é aplicado ao modelo (7.17). A importância dessas simulações deve-se ao fato de que tanto o método SDRE, utilizado para realizar as simulações dessa seção, quanto o modelo (7.17), é escolhidos para a realização de simulações da aplicação do controle para a navegação do robô.

Escolhem-se quatro conjuntos de parâmetros nos quais os valores de  $x$  e  $y$  de posição inicial do robô são fixos e uma orientação inicial do robô variada, para demonstrar a eficácia do controle para todos os ângulos iniciais de posicionamento. Isso porque a estabilização da postura é um ponto crítico de alguns métodos de controle de robô, como a linearização por realimentação de estados. Os parâmetros da simulação encontram-se na tabela abaixo:

Parâmetros	$\tilde{x}_0$ (ponto inicial da traj. de referência )	$\tilde{x}_f$ (ponto final da traj. de referência)	$x_0$ (ponto inicial da traj. do sistema)
P1	$[0; 1; \pi/4]$	$[5; 6; \pi/4]$	$[0; -2; 0]$
P2	$[0; 1; \pi/4]$	$[5; 6; \pi/4]$	$[0; -2; \pi/2]$
P3	$[0; 1; \pi/4]$	$[5; 6; \pi/4]$	$[0; -2; \pi]$
P4	$[0; 1; \pi/4]$	$[5; 6; \pi/4]$	$[0; -2; -\pi/2]$

Tabela 7. 1- Valores de Conjuntos de Parâmetros para Simulação Controle SDRE

A trajetória de referência tem a forma linear. Inicia no ponto  $\tilde{x}_0 = [0; 1; \pi/4]$  e termina no ponto  $\tilde{x}_f = [5; 6; \pi/4]$ . O robô parte do repouso o que significa que a velocidade inicial é nula e a velocidade de chegada ao final da trajetória deve ter nas suas componentes  $x$  e  $y$  os seguintes valores:  $\dot{\tilde{x}}_f = 0,001$  e  $\dot{\tilde{y}}_f = 0,001$ . A trajetória do robô inicia no ponto  $x_0$  indicados pela tabela 7.1.

Na figura 7.13 a ) está representada a trajetória controlada em linha sólida e em hachura a trajetória de referência para os parâmetros P1, ou seja, o ângulo inicial  $\theta_0 = 0$ . Na figura 7.13 b) está representada a evolução das coordenadas em desvios correspondentes às três coordenadas de estado do robô. Dessas figuras é possível verificar que em menos de 4 segundos a trajetória é estabilizada à referência.

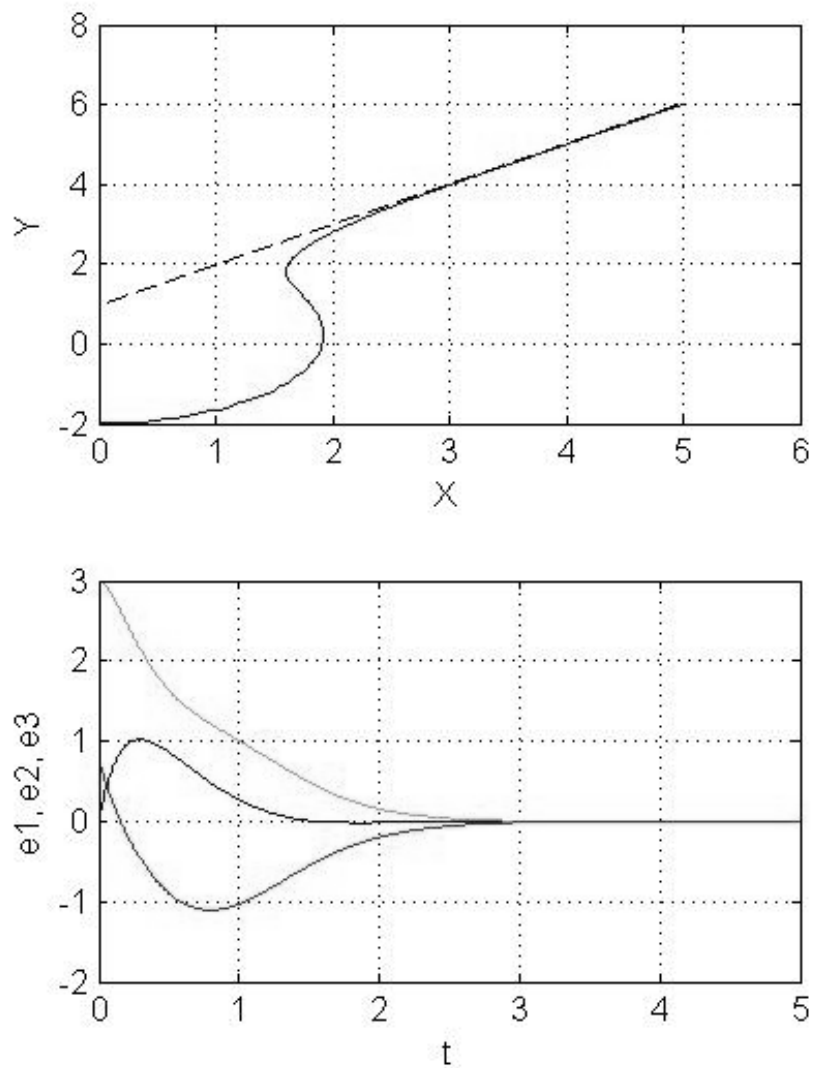


Figura 7. 13- Parâmetros P1 a) Trajetória Controlada SDRE b)Coordenadas em Desvios .

As figuras 7.14 refletem os parâmetros P2, ou seja, quando a coordenada angular inicial do robô é  $\theta_0 = \pi/2$ . Na figura 7.14 a), está a trajetória controlada que converge para a trajetória de referência em hachuras. A figura 7.14 b ) mostra as coordenadas em desvios que são levadas a zero em 2 segundos. A convergência é rápida devido à posição benéfica em relação à referência.

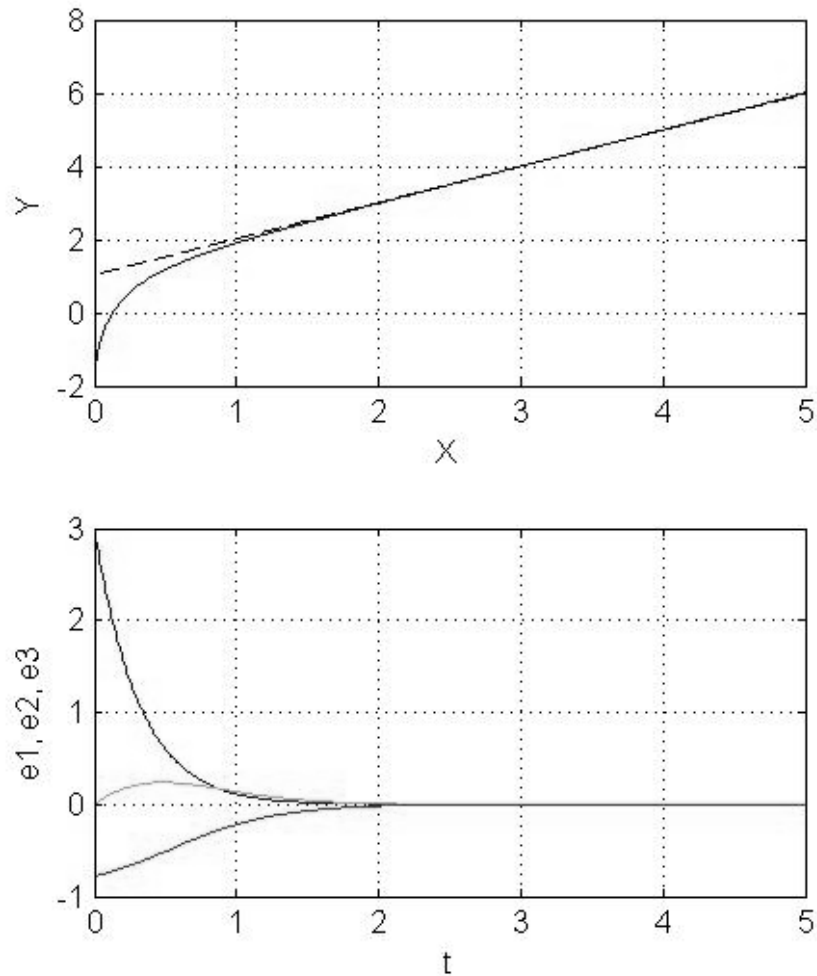


Figura 7. 14- Parâmetros P2 a) Trajetória Controlada SDRE b)Coordenadas em Desvios

As figuras 7.15 representam os parâmetros P3 da tabela , onde o ângulo inicial do sistema é  $\theta_0 = \pi$ . Devido a condição inicial angular a postura inicial do robô está voltado de costas para o sentido da trajetória de referência. Por isso, a estabilização à trajetória demora mais tempo (em torno de 3 segundos).

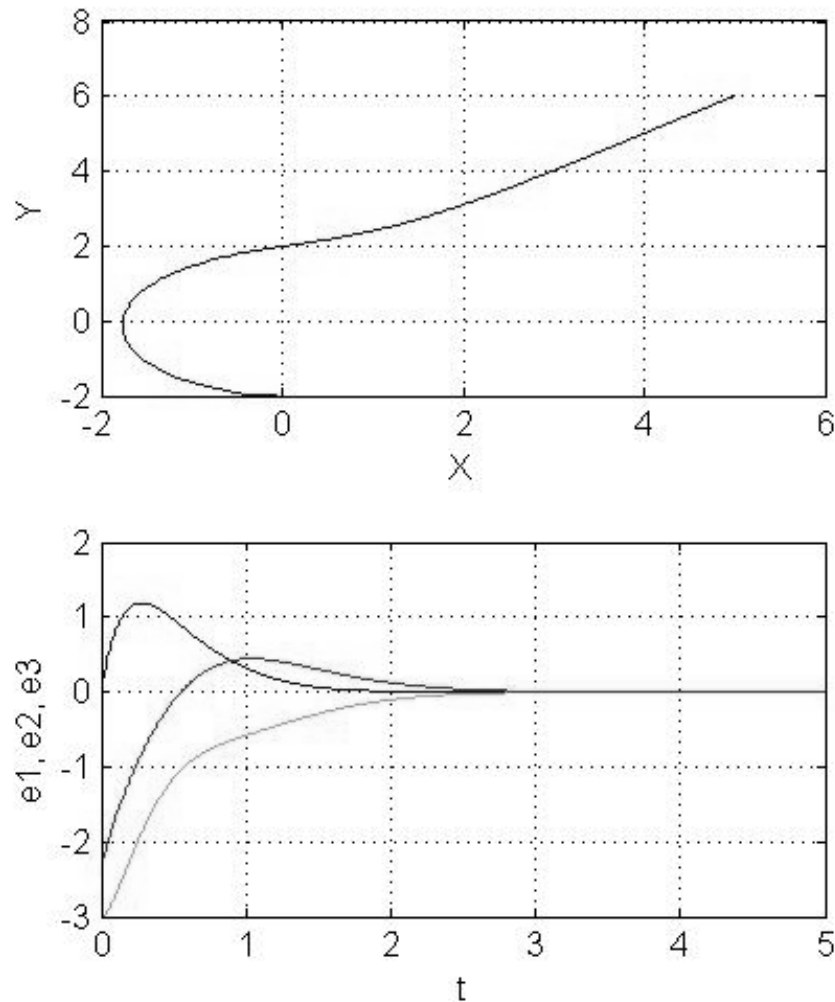


Figura 7. 15- Parâmetros P3 a) Trajetória Controlada SDRE b)Coordenadas em Desvios

As curvas da figura 7.16 são construídas utilizando os parâmetros P4 da tabela 7.1. Na figura 7.16 a) o ângulo inicial é  $\theta_0 = -\pi/2$  , ou seja o robô está rotacionado de um ângulo grande e por isso a estabilização à trajetória de referência leva em torno de 3

segundos (mais do que na figura 7.14) . Entretanto, os desvios na figura 7.16 b) são levados a zero e a trajetória controlada converge impecavelmente para a referência.

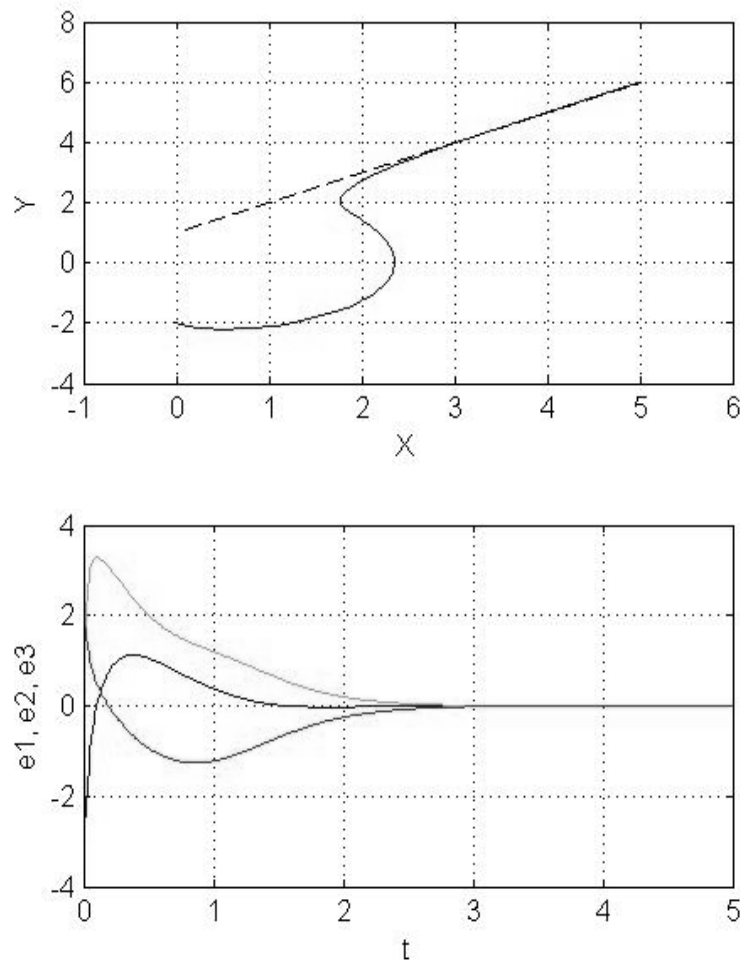


Figura 7. 16- Parâmetros P4 a) Trajetória Controlada SDRE b) Coordenadas em Desvios

Nas próximas simulações é considerada uma trajetória de referência circular, com velocidade angular constante. Têm-se, no mínimo, quatro situações de determinação dessa trajetória e do seu centro. No primeiro caso, como mostra a figura 7.17 deseja-se que o robô realize um movimento circular para esquerda no sentido anti-horário. Nesse caso fixa-se uma velocidade angular positiva e o centro do movimento é dado por :

$$\begin{aligned} c_x &= \tilde{x}_0 - r \cdot \sin(\tilde{\theta}_0) \\ c_y &= \tilde{y}_0 + r \cdot \cos(\tilde{\theta}_0) \end{aligned} \quad (7.23)$$

A figura 7.17 ilustra a convergência da trajetória controlada (em azul) à referência (em vermelho). A evolução temporal das coordenadas em desvio é mostrada na figura 7.18

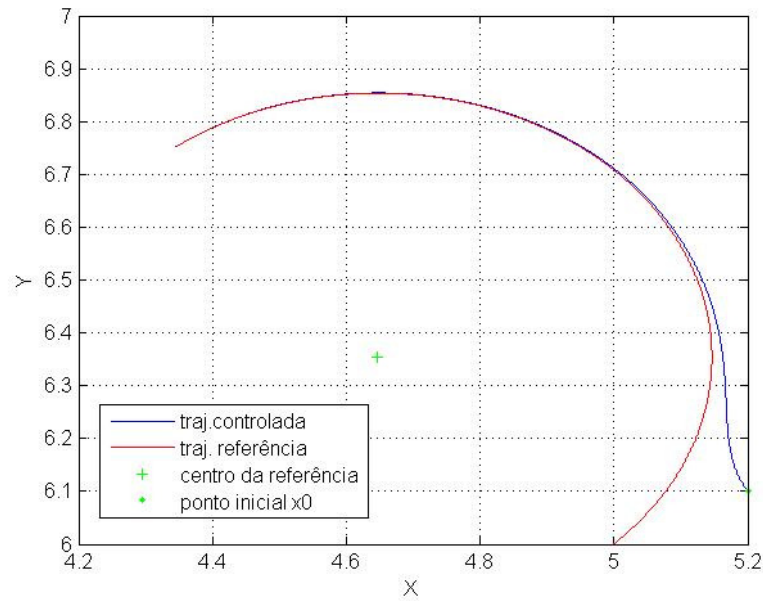


Figura 7. 17- Trajetória Controlada SDRE

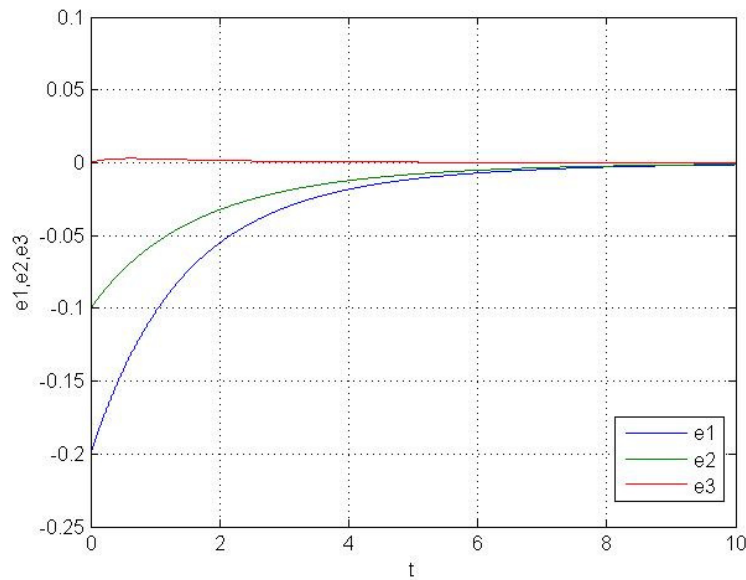


Figura 7. 18-Coordenadas em Desvios Trajetória Circular

No segundo caso, deseja-se um movimento para direita no sentido horário, como mostra a figura 7.18, na qual também se observa uma convergência desejada à trajetória de referência, que demora porém cerca de 8 segundos, como é observado na figura 7.20.

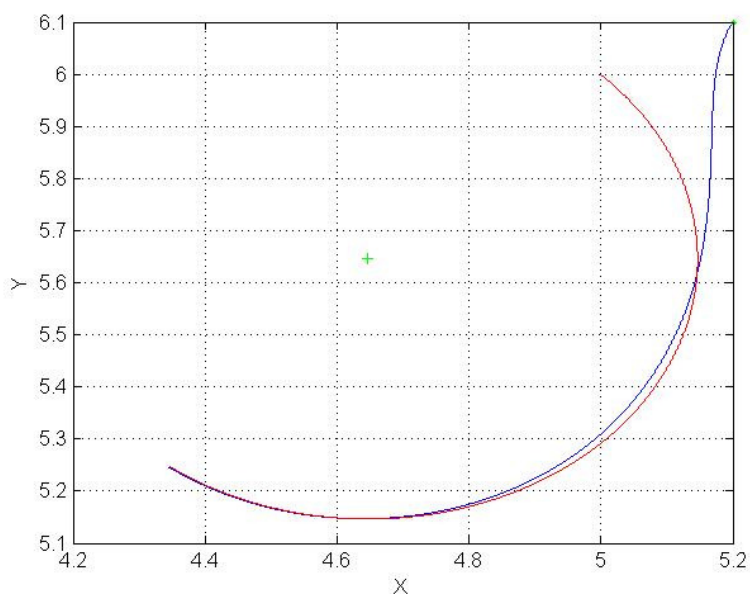


Figura 7. 19- Trajetória Controlada SDRE

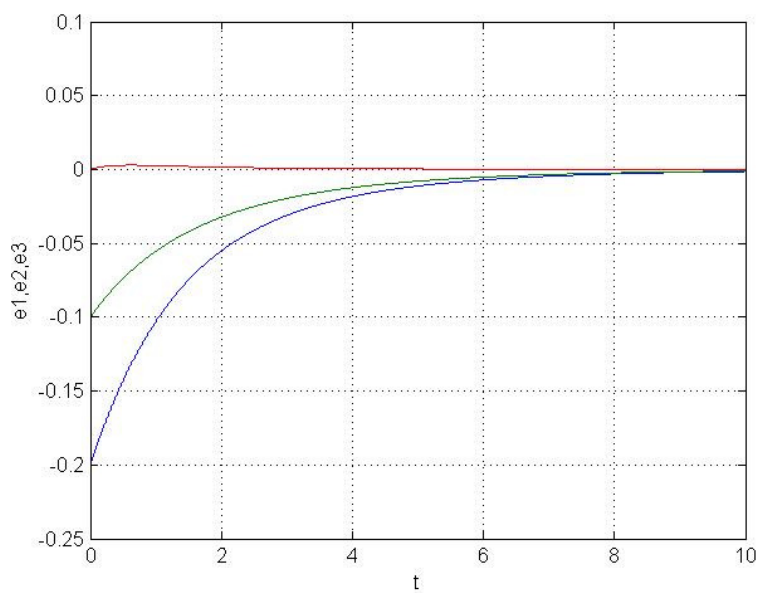


Figura 7. 20- Coordenadas em Desvios Trajetória Circular

No terceiro caso, deseja-se uma curva para frente, pela direita, no sentido anti-horário, conforme ilustrado na figura 7.21. Nesse caso a velocidade angular é positiva. As figuras 7.21 e 7.22 mostram a convergência da trajetória controlada à referência, processo que leva cerca de 8 segundos.

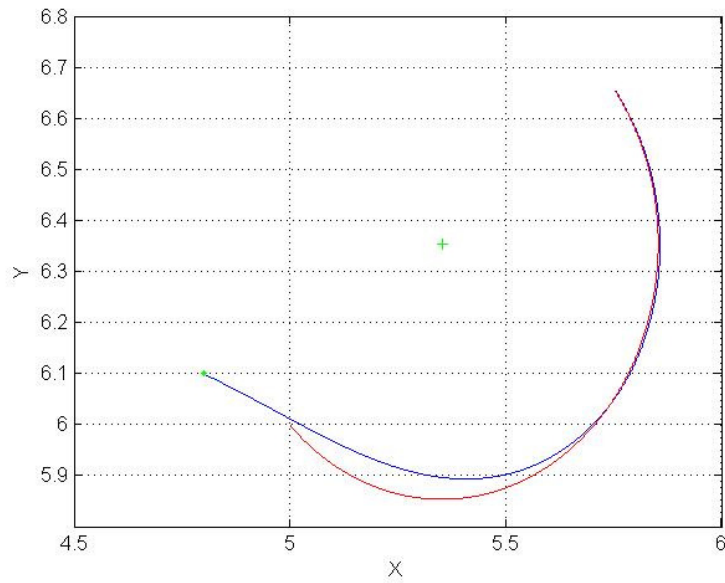


Figura 7. 21- Trajetória Controlada SDRE

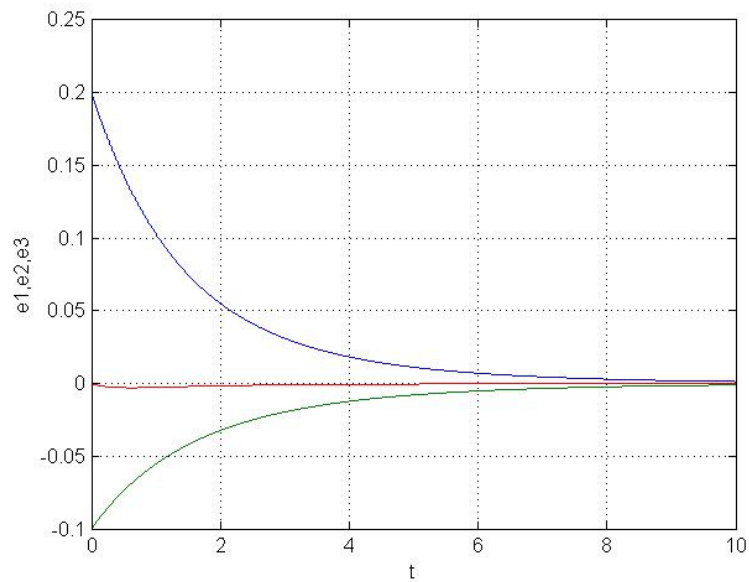


Figura 7. 22- Coordenadas em Desvios Trajetória Circular



No ultimo dos quatro casos, deseja-se uma curva para frente, pela esquerda no sentido horário. Nesse caso, a velocidade angular é negativa. Na figura 7.23 está representado o ultimo caso de trajetória de referência considerada, juntamente com a trajetória controlada obtida pelo controle SDRE. A figura 7.24 demonstra a evolução das coordenadas em desvio para esse caso. Conforme a figura 7.24 nota-se que a regulação demora cerca de 10 segundos.

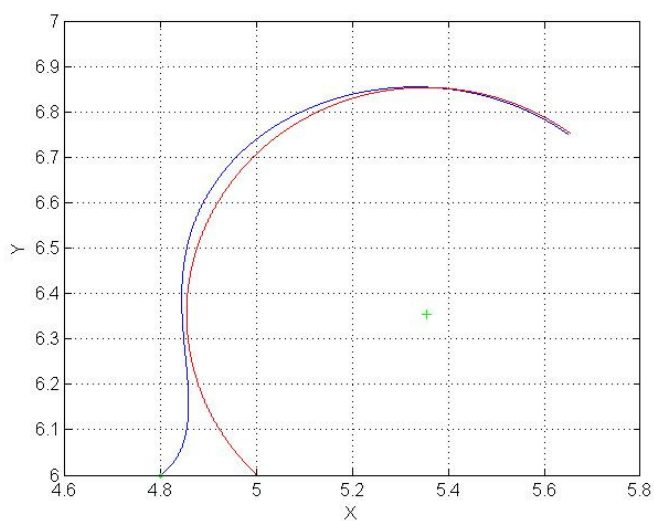


Figura 7. 23- Trajetória Controlada SDRE

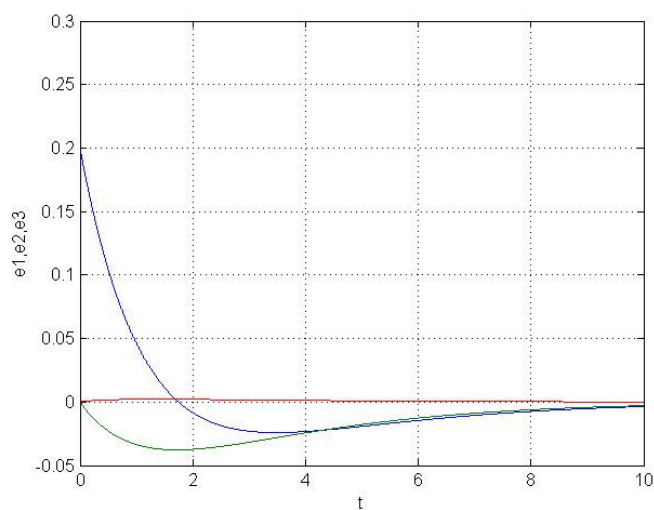


Figura 7. 24- Coordenadas em Desvios Trajetória Circular

### 7.1.6. Simulação de União de Trajetórias Reta e Circulo

Nessa simulação é mostrado o controle de robô para a união de duas trajetória de referência diferentes. A trajetória controlada inicia no ponto  $x_0 = \left[0, -2, \frac{\pi}{4}\right]$ . A trajetória de referência reta inicia em  $\tilde{x}_0 = \left[0, 1, \frac{\pi}{4} - \frac{\pi}{2}\right]$ , com velocidades iniciais  $\dot{\tilde{x}}_0 = \dot{\tilde{y}}_0 = 0$  e termina em  $\tilde{x}_0 = \left[5, 6, \frac{\pi}{4}\right]$ , onde começa uma nova trajetória de referência circular cujo centro é dado por :

$$\begin{aligned} c_x &= 5 - 0.5 \cdot \sin\left(\frac{\pi}{4} - \frac{\pi}{2}\right), \\ c_y &= 6 + 0.5 \cdot \cos\left(\frac{\pi}{4} - \frac{\pi}{2}\right), \end{aligned} \quad (7.24)$$

com velocidade angular  $\omega_r = 0.3$  rad/s e raio de trajetória  $r=0.5$ .

A figura 7.25 mostra essa simulação. Em azul encontra-se a trajetória circular controlada, em vermelho, a trajetória reta. A linha preta representa a trajetória de referência.

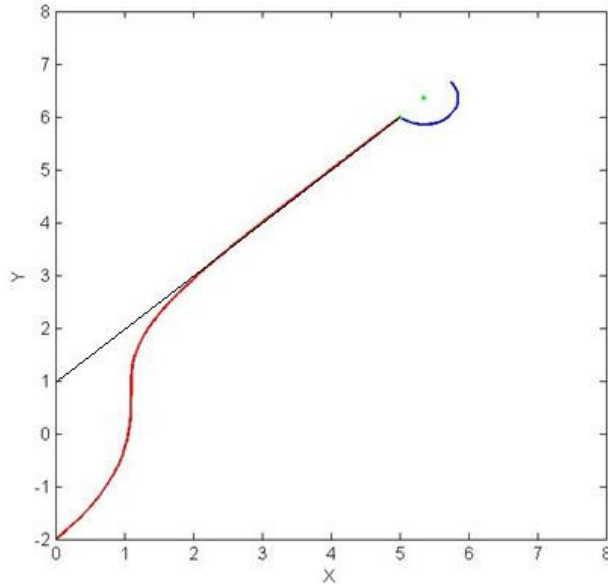


Figura 7. 25-Trajetoária Controlada Reta e Círculo

## **7.2. Simulações de Estimação de Estado Através da Odometria e da Visão**

Esta seção tem o objetivo de aplicar os métodos de medição de estado utilizados na localização do robô. São testados separadamente os esquemas de visão e odometria com parâmetros distintos, a fim de demonstrar a influência de erros gerados por cada um dos métodos.

### **7.2.1. Simulação de Observação de Estado pela Visão**

O objetivo dessas simulações é demonstrar o processo de observação de estado e os erros envolvidos. A simulação consiste em fixar dois pontos : P e A de orientação para a triangulação pelas câmeras e fixar uma trajetória de referência cujo estado é observado e suas coordenadas estimadas pela visão estereoscópica.

. A trajetória de referência evolui do ponto  $\tilde{x}_0 = [0; -2]$ , com velocidade linear  $v_r = 10$ . As coordenadas dos pontos A e P são: A=[350; 250] e P=[200 ; 260].

A respeito aos parâmetros das câmeras, as telas de projeção dos sensores óticos dessas são considerados com uma única linha de comprimento igual a 3,2 mm e resolução em pixels variável conforme o caso (320 ou 640 pixels). A resolução de câmeras disponíveis no mercado é variável ( a resolução mais comum é de 320 pixels), com o custo diretamente proporcional ao aumento dessa resolução. A distância focal das câmeras é 0,27mm. O intervalo de tempo considerado para a simulação é de 44 segundos. As câmeras são alocadas a uma distância entre si de 0,292 m.

Na figura 7.26 está representada a primeira simulação, para uma resolução do sensor das câmeras de 320 pixels. É possível perceber que, principalmente, no início da trajetória há um erro grande na observação da referência, que este diminui com a aproximação do sistema ao alvo.

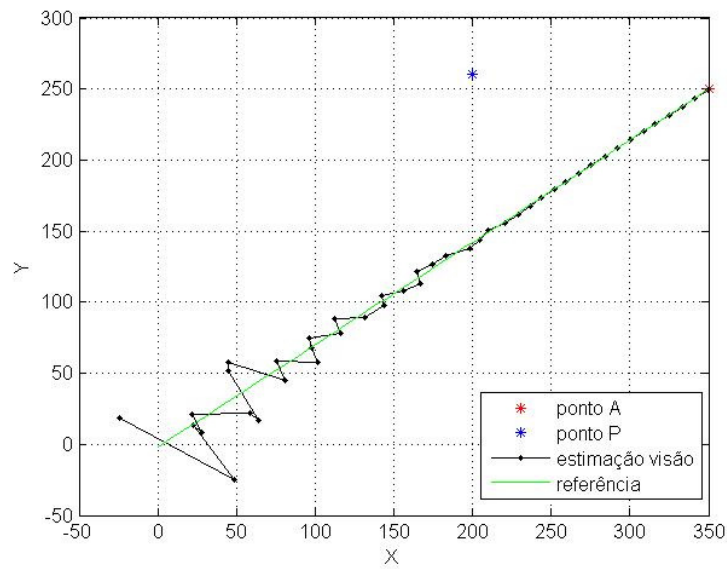


Figura 7. 26- Trajetória Estimada pela Visão. Resolução de 320 pixels.

No outro caso, ilustrado pela figura 7.27 considera-se uma resolução maior (640 pixels) dos sensores das câmeras. O erro, nas observações iniciais, é menor do que no caso anterior e as observações convergem para a trajetória de referência mais rápido

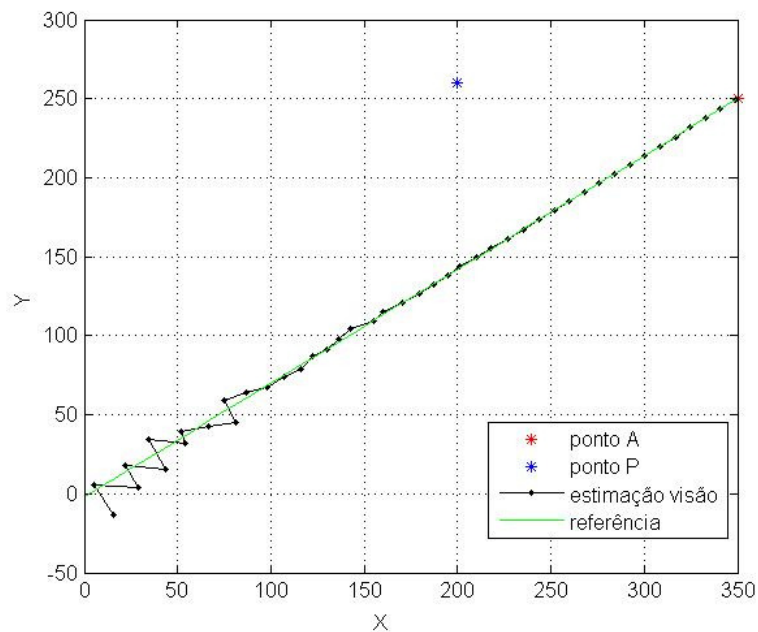


Figura 7. 27- Trajetória Estimada pela Visão . Resolução de 640 pixels.

Conclui-se que a distância do veículo em movimento ao ponto de orientação influencia na qualidade da estimação do seu estado. Quanto mais longe robô estiver dos pontos A e P maior será este erro. O erro diminui a medida que o robô se aproxima do ponto alvo A. Além disso, a resolução do sensor das câmeras tem um papel muito importante nessa estimação de coordenadas. O aumento da resolução reduz o erro de quantização diminuindo sensivelmente o erro de medição de coordenadas.

### **7.2.2. Simulação de Estimação de Estado pela Odometria**

Essas simulações têm a finalidade de demonstrar a influência de alguns parâmetros no erro de estimação de estado pela odometria. Considera-se duas trajetórias, uma calculada pelo modelo de odometria e outra de referência, ambas iniciam no ponto  $\tilde{x}_0 = [0; -2]$ . Testam-se dois valores para o parâmetro de erro pra ilustrar a repercussão desse erro na estimação de coordenada de localização do veículo.

O erro de odometria tem o desvio padrão  $\delta = 0.02$  para primeira simulação. O gráfico da figura 7.28 demonstra este caso. Devido ao erro pequeno a trajetória estimada tem erro mínimo.

Na figura 7.31 o desvio padrão aumenta para  $\delta = 0.2$  o que causa um desvio da estimação que ocorre a partir do meio da trajetória estimada se torna maior com o tempo. Isso ocorre porque os erros de odometria são cumulativos, isto é, ao estado anterior com erro adiciona-se um novo estado com erro, fazendo com que este erro se propague com a distância percorrida.

Por isso, a estimação pela odometria pode ser crítica, mesmo para erro pequeno porque a acumulação desse erro pode levar a desvios significativos quando utilizado esse métodos para percorrer longas distâncias.

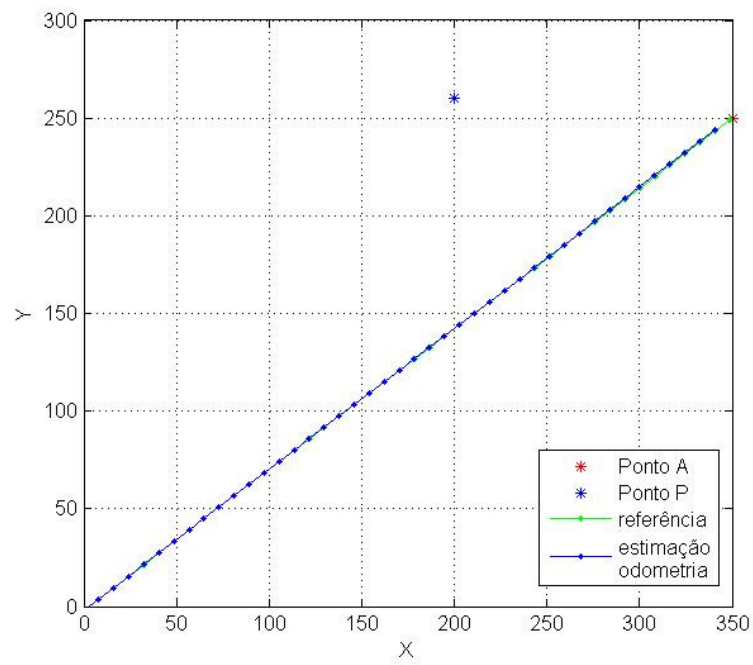


Figura 7. 28- Trajetória Estimada pela Odometria, erro menor.

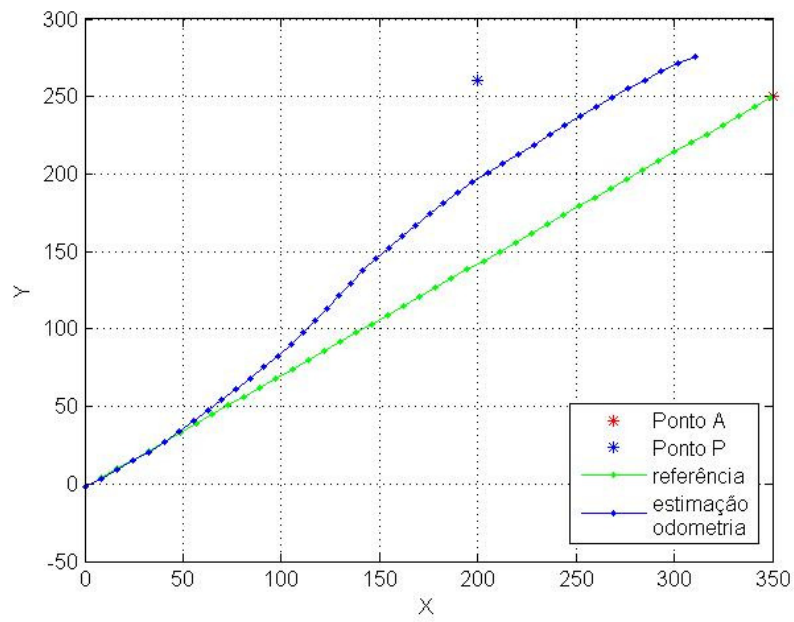


Figura 7. 29- Trajetória Estimada pela Odometria, erro maior.

### 7.3. - Simulações de Navegação do Robô através do controle e filtro de Kalman

Essa seção é dedicada à apresentação de resultados para a navegação de robô através de estratégias de controle descritas no capítulo 6 que utilizam informações de odometria, visão de duas câmeras e controle de trajetória do robô, de formas distintas.

#### 7.3.1. Estratégia de navegação discreta no tempo

Esse conjunto de quatro simulações é realizado para a estratégia de navegação discreta no tempo, descrita detalhadamente na seção 6.4.2.2. O objetivo do robô é realizar uma trajetória retilínea até o alvo A, utilizando a estimação de estado discreta através da fusão de informações da odometria e da visão de duas câmeras, realizada pelo filtro de Kalman. Os pontos de referência A e P utilizados pelas câmeras para a orientação são localizados nas coordenadas do mundo como  $A=[3,50; 2,5]$  e  $P=[2; 2,6]$ . A unidade de distância é  $m$ . O robô tem o seguinte estado inicial em coordenada do mundo:  $x_0 = [0 \quad -0.02 \quad \pi/4]$ .

Simulação	$\tilde{x}_0$ - início da trajetória de referência	$\delta$ – <i>odometria</i>	Resolução das câmeras(pixels)
D1	estimado	0,02	320
D2	estimado	0,2	320
D3	estimado	0,02	570
D4	estimado	0,2	570

Tabela 7. 2-Parâmetros para Navegação Discreta

A tabela 7.2 representa quatro conjuntos de parâmetros utilizados para a simulação da estratégia de navegação discreta. O parâmetro  $\tilde{x}_0$  diz respeito ao ponto inicial da trajetória de referência estimado pelas câmeras. O parâmetro  $\delta$  é desvio padrão do erro de odometria e o último parâmetro é a resolução em pixels do sensor das câmeras que é considerado como uma linha de  $n$  pixels de comprimento.

Nas simulações a seguir são mostradas quatro curvas que representam, respectivamente: em vermelho- trajetória de referência, em preto- a observação de posição do robô pelas câmeras, em verde – as estimações de estado pelo filtro de Kalman e, por fim, em azul – a trajetória verdadeira simulada do robô, ou seja, aquela que ele de fato seguiria se posicionado no ambiente.

A simulação ilustrada na figura 7.30 leva em conta os parâmetros D1 com um erro pequeno de odometria. O início da trajetória de referência é estimado apenas pela visão das câmeras, assim como o início da trajetória estimada. Isso provoca um erro inicial na trajetória estimada. Nesse gráfico a visão apresenta erros grandes na observação do estado (devido à baixa resolução das câmeras), principalmente no início da trajetória

A simulação, dada pela figura 7.31 se refere aos parâmetros D2. Em relação aos parâmetros D1, os erros de odometria são maiores. O erro de visão provoca erro inicial, que é, aos poucos, minimizado convergindo para um valor pequeno, possibilitando, dessa forma, chegar próximo ao alvo e manter uma trajetória regular. Da figura 7.33 percebe-se também que a estimativa de estado é uma trajetória regular que corresponde à trajetória verdadeira do robô.



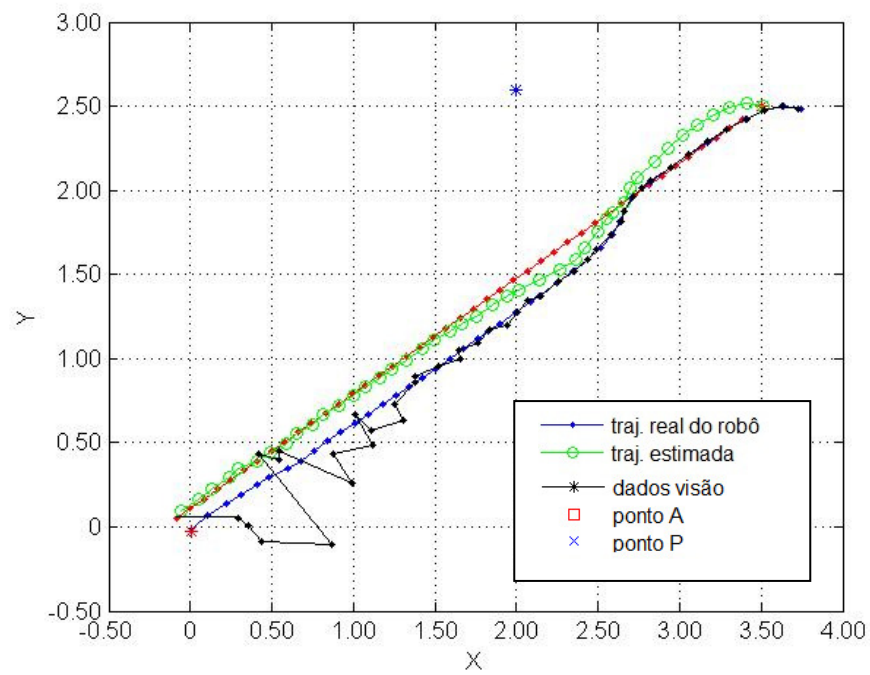


Figura 7. 30- Trajetória de Navegação do Robô com Parâmetros D1

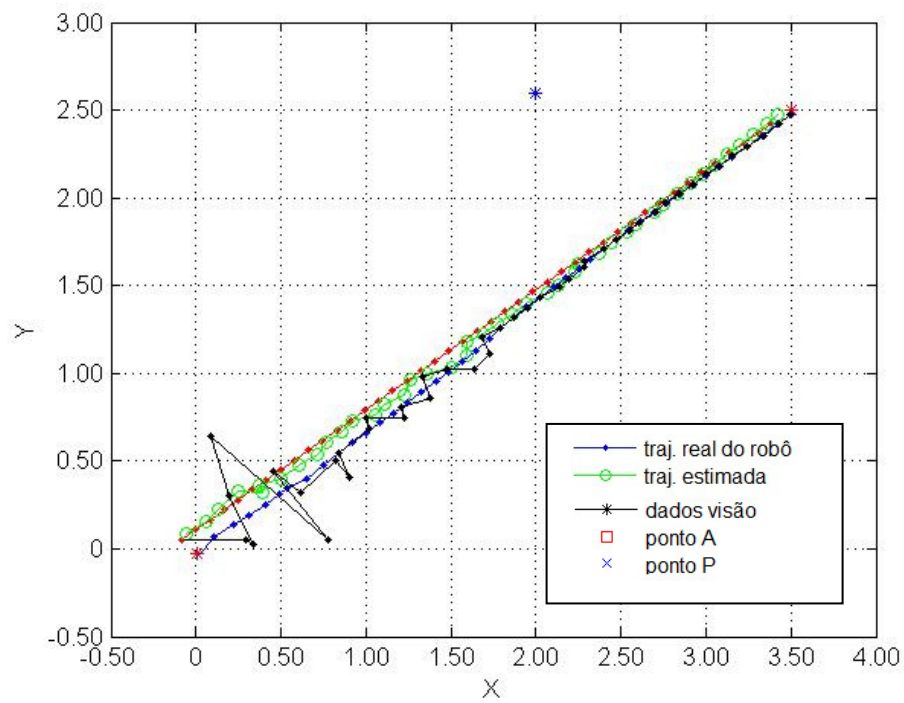


Figura 7. 31- Trajetória de Navegação do Robô com Parâmetros D2

A simulação da figura 7.32, para os parâmetros D3 tem por objetivo testar um erro menor das câmeras aumentando sua resolução. No gráfico dessa figura é possível perceber que há um erro significativamente menor no início da trajetória estimada e também que as observações representadas pelos pontos em preto apresentam mais precisão do que nas simulações anteriores. A trajetória estimada converge à trajetória de referência e ao alvo A. No entanto a trajetória real do robô converge à trajetória de referência com pequeno erro.

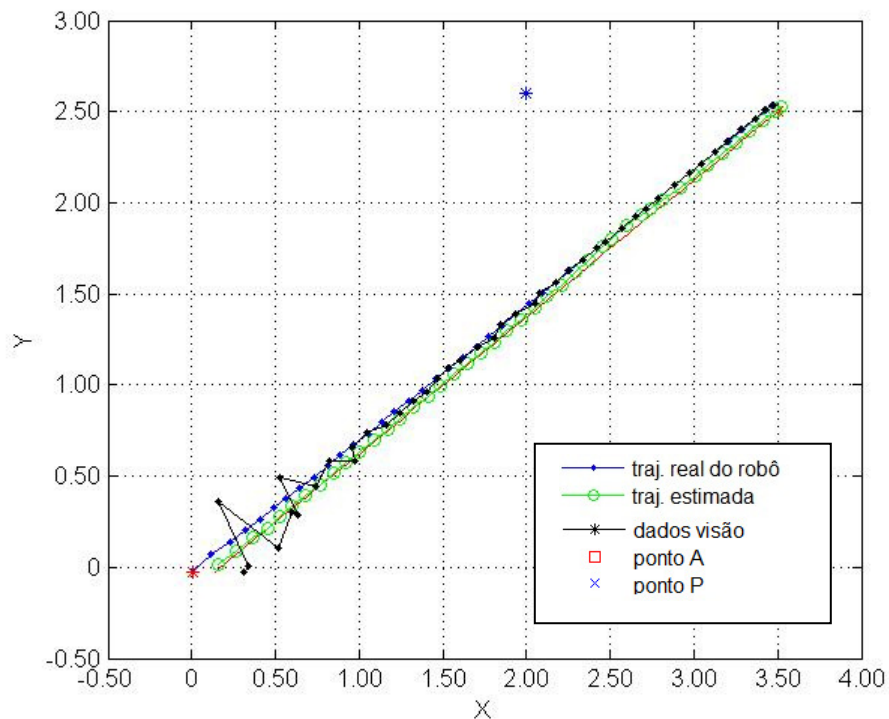


Figura 7. 32-Trajetoória de Navegação do Robô com Parâmetros D3

Na simulação D4 , representada pela figura 7.35 aumenta-se o erro de odometria. Assim como na simulação anterior, a trajetória inicial estimada inicia com um erro em relação a trajetória verdadeira do robô, mas convergindo a essa ao longo do percurso. Ambas as trajetórias convergem à trajetória de referência, aproximadamente, a partir do ponto 1,5 m em  $x$  e 1 m em  $y$ .

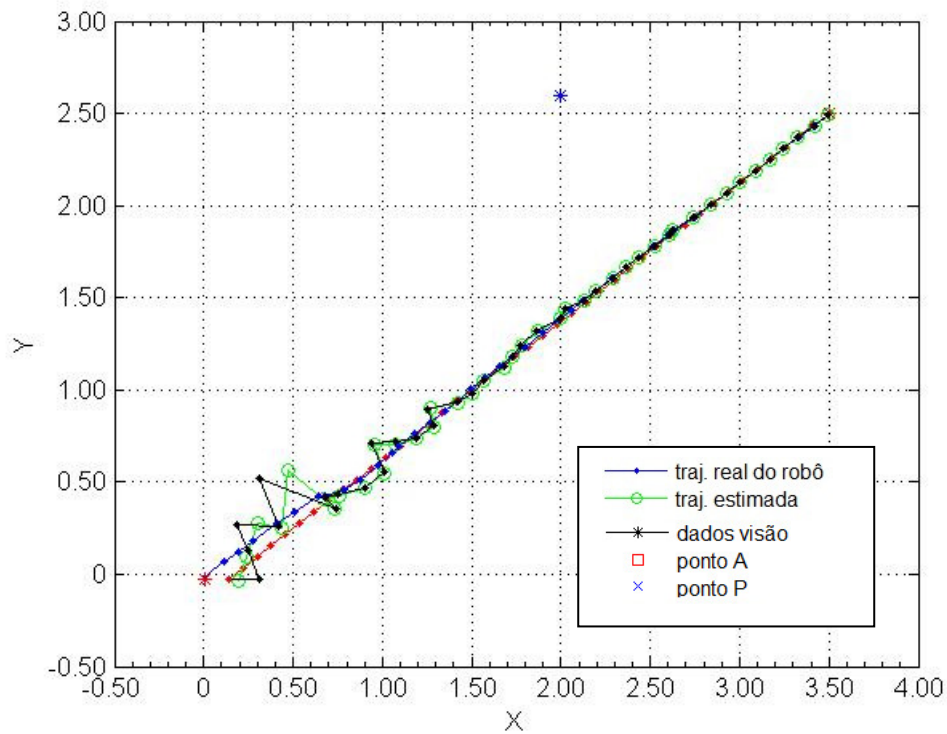


Figura 7. 33- Trajetória de Navegação do Robô com Parâmetros D4

As simulações da estratégia de navegação discreta demonstram a navegabilidade do sistema nessa configuração. No entanto, demonstram, também, que o fator essencial na navegação do robô é determinado pela qualidade inicial de observações pela visão estereoscópica, já que, pela formulação do problema, o ponto inicial de localização do sistema é determinado tão somente pelas câmeras. Erros de observação inicial grandes podem dificultar, como na figura 7.32, a convergência da estimação de estado e até mesmo impossibilitar a navegação.

### 7.3.2. Estratégia de navegação de horizonte recuável (MPC)

Neste conjunto de simulações é utilizada uma estratégia de navegação que une controle contínuo com medições de estado discretas. Por utilizar o controle SDRE que resolve o problema de controle no tempo infinito e utilizar estimações de estados discretas no tempo essa estratégia se enquadra no esquema conhecido como *Model Predictive Control* ou também como *Receding Horizon* (Horizonte Recuável). O desenvolvimento desse controle é detalhado no capítulo 5.

Assim como na seção anterior, o objetivo do robô é realizar uma trajetória retilínea até o alvo A, utilizando estimação de estado discreta no tempo e o controlador SDRE contínuo nos intervalos entre as medições. As simulações para essa estratégia têm quatro conjuntos de parâmetros para uma mesma resolução de sensor das câmeras, simulando diferentes erros de deslocamento, de visão e com condição inicial da referência sendo fixa ou estimada. O robô tem a coordenada inicial verdadeira no mundo  $x_0 = [0; 0.02; \pi/4]$  em todas as simulações dessa estratégia. As medições são disponíveis em 35 momentos discretos. Além disso são apresentados mais quatro conjuntos de parâmetros com as mesmas variações para uma resolução maior do sensor visual das câmeras.

Nos gráficos, linha azul representa a trajetória verdadeira do deslocamento do robô, a linha verde - a trajetória estimada pelo filtro de Kalman e a trajetória em preto representa as medições do estado do robô realizadas pelas câmeras. O símbolo X em azul representa o ponto P utilizado para a orientação visual e o quadrado em vermelho representa a coordenada do ponto A que é o alvo do robô nessa simulação.

Simulação	$[\tilde{x}_0, \tilde{y}_0]$	$\sigma$ erro de deslocamento	d entre as câmeras(m)	Resolução câmeras(pixels)
C1	estimada	0.02	0.146	280
C2	[0,-2]	0.02	0.146	280
C3	estimada	0.2	0.146	280
C4	[0,-2]	0.2	0.146	280

Tabela 7. 3- Parâmetros de Navegação com Estratégia de Horizonte Recuável (MPC) de C1 a C4.

A figura 7.36 retrata o conjunto C1 de parâmetros. Dela percebe-se que inicialmente o erro de medição do estado do robô pelas câmeras é muito grande levando o filtro de Kalman, no momento inicial, a estimar a posição do robô longe da localização real. No entanto, com a evolução da trajetória, as medições das câmeras passam a ser mais confiáveis fazendo com que a trajetória estimada convirja para a real e ambas tendam ao alvo. Isso significa que o robô consegue estimar a sua localização no espaço e portanto navegar com segurança.

Na figura 7.35 é simulado o conjunto de parâmetros C2 que são essencialmente os mesmos da simulação anterior exceto pelo início da trajetória que é conhecido, ao invés de estimado. Essa simulação tem a função de demonstrar que uma localização precisa no início da trajetória do robô proporciona também a estimação mais precisa do seu estado ao longo de toda a trajetória. Devido às imprecisões de medição do estado pelas câmeras, a trajetória estimada não coincide com trajetória verdadeira do robô em um intervalo aproximado entre os pontos  $xy (1; 0,3) \text{ m}$  e  $(2,5; 1,7) \text{ m}$ .

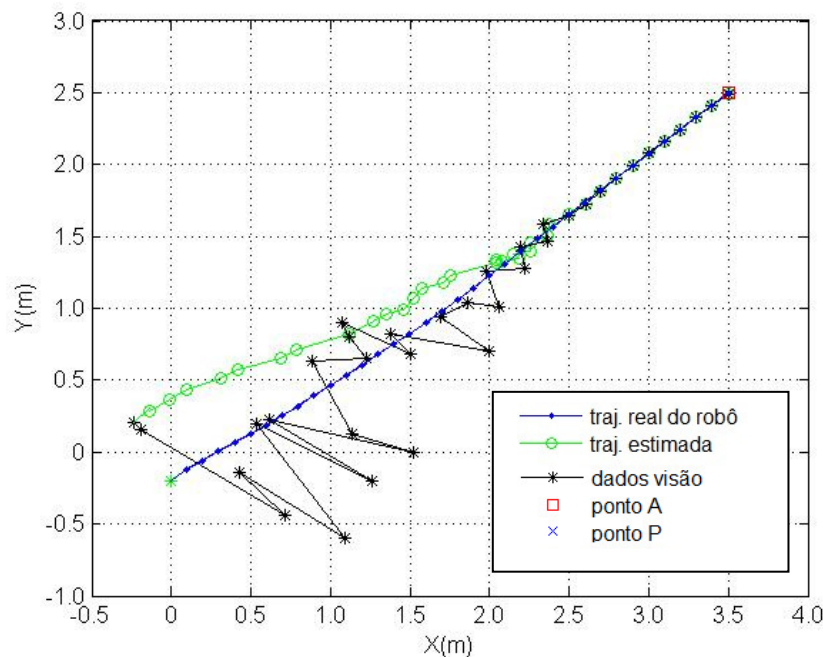


Figura 7. 34- Trajetória de Navegação do Robô com Parâmetros C1

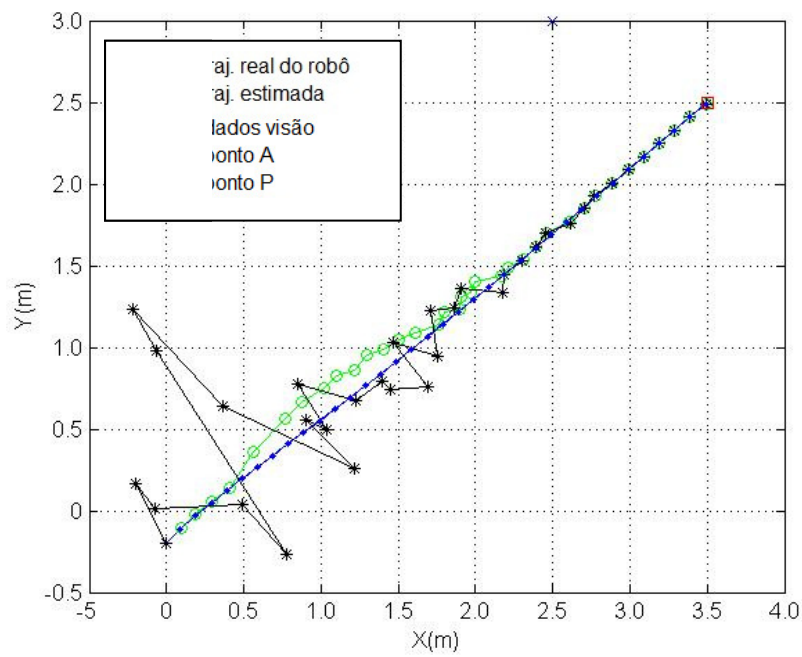


Figura 7. 35- Trajetória de Navegação do Robô com Parâmetros C2

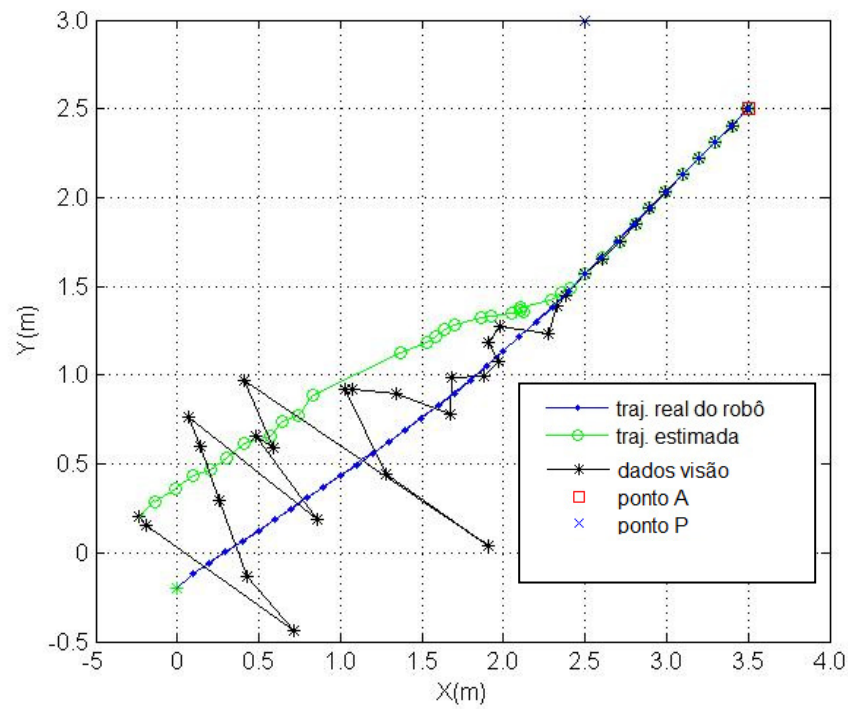


Figura 7. 36- Trajetória de Navegação do Robô com Parâmetros C3

Na simulação da figura 7.36 são utilizados os parâmetros C3 que demonstram a situação com erro de odometria maior que nas simulações anteriores e com o início da trajetória desconhecido e estimado pela visão estereoscópica. Devido a isso a trajetória estimada possui um grande erro caracterizado pelo desvio da trajetória verdadeira. No entanto, quando os erros da visão diminuem a trajetória estimada converge para a verdadeira e a navegação se torna precisa, atingindo o alvo A.

Na figura seguinte ( figura 7.38) que representa o conjunto de parâmetros C4, o erro de deslocamento e de câmera é o mesmo da simulação C3, no entanto, a posição do ponto inicial da trajetória de referência é fixa, ao invés de estimada. Isso reduz o deslocamento da trajetória real do robô da trajetória ideal no momento inicial, na figura 7.39. Erros aleatórios da visão perturbam a trajetória estimada, provocando erros aproximadamente na metade da trajetória estimada. Porém, esse erro na trajetória estimada não é significativo e há no sistema convergência ao alvo.

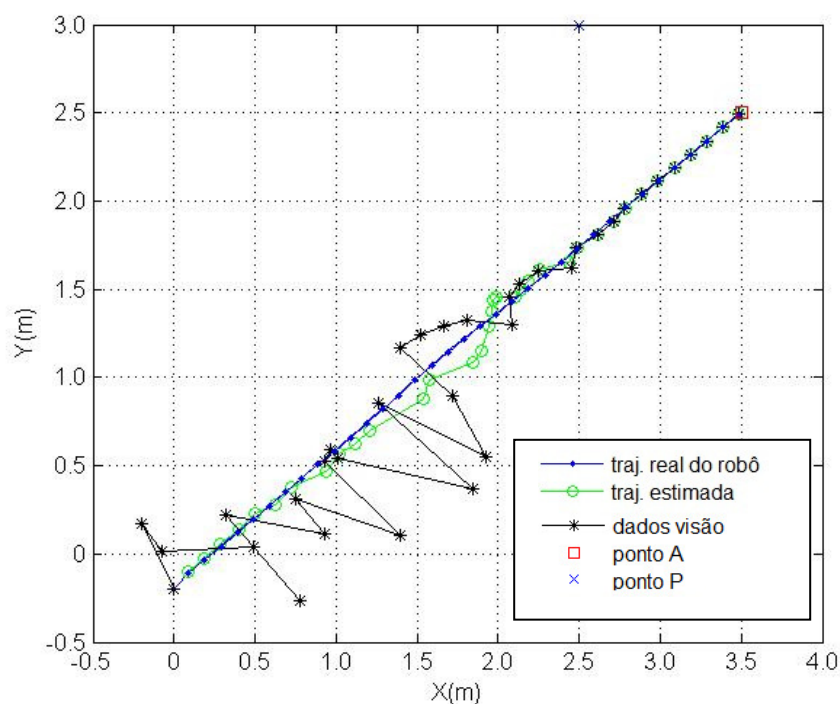


Figura 7. 37- Trajetória de Navegação do Robô com Parâmetros C4

Das quatro simulações anteriores fica claro que a visão estereoscópica é fonte de grandes erros, no início da trajetória, e isso é um ponto crítico da navegação já que o posicionamento inicial é realizado apenas pela visão. Assim há possibilidade de grandes desvios durante a trajetória com risco de não convergência da estimação de estado. Por isso são realizadas quatro simulações correspondentes ao aumentando o parâmetro de resolução do sensor das câmeras. Uma maior resolução significa aumento na precisão de mapeamento das coordenadas do mundo nas câmeras, diminuindo assim o erro de quantização envolvido. A tabela 7.4 apresenta os novos conjuntos de parâmetros.

Simulação	$[\tilde{x}_0, \tilde{y}_0]$	$\sigma$ erro de deslocamento	d entre as câmeras(m)	Resolução câmeras(pixels)
C5	estimada	0.02	0.146	570
C6	[0,-2]	0.02	0.146	570
C7	estimada	0.2	0.146	570
C8	[0,-2]	0.2	0.146	570

Tabela 7. 4- Parâmetros de Navegação com Estratégia de Horizonte Recuável (MPC) de C5 a C8

Na figura 7.38 está representada a simulação do conjunto C5 de parâmetros. Em azul está a trajetória de navegação real do robô , em verde a trajetória estimada pelo filtro de Kalman, em preto as medições de coordenada realizadas pela visão. Observa-se que o erro da visão é sensivelmente menor do que nas simulações anteriores e, especialmente, para esse conjunto de parâmetros a estimação apresenta poucos erros e desvios.



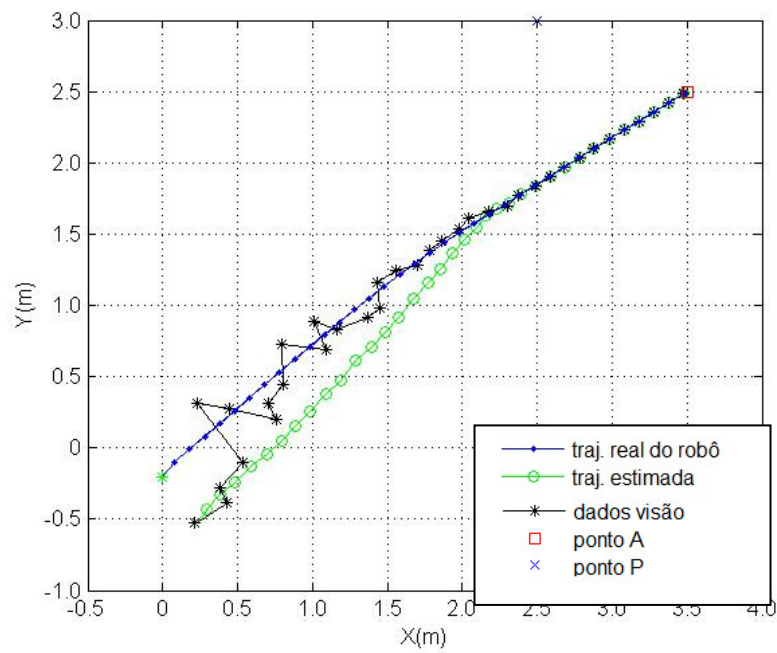


Figura 7. 38- Trajetória de Navegação do Robô com Parâmetros C5

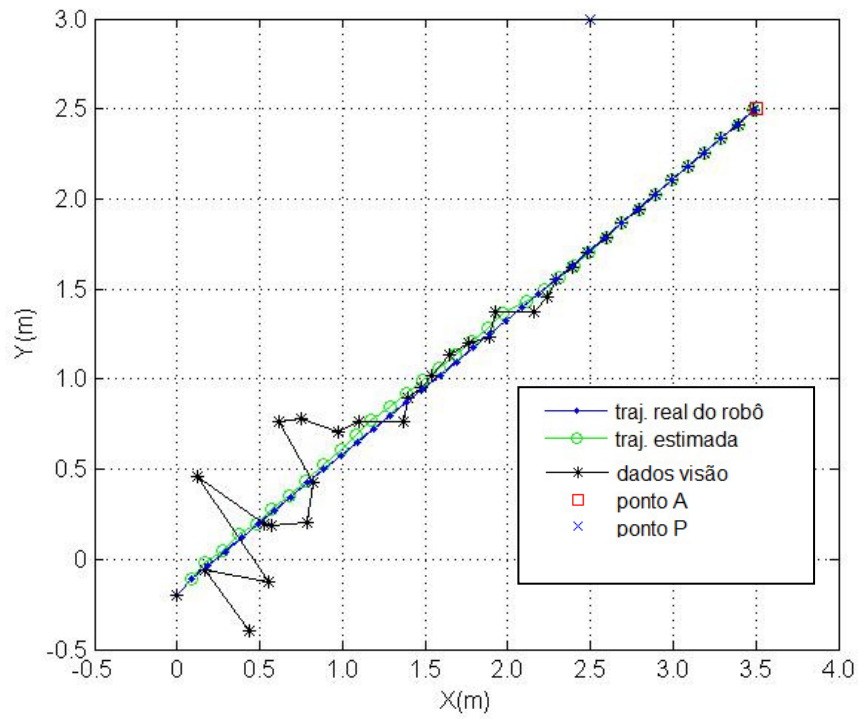


Figura 7. 39- Trajetória de Navegação do Robô com Parâmetros C6

A figura 7.39 representa o conjunto de parâmetros C6 que é equivalente ao conjunto C5 mas com o início da trajetória conhecido ao invés de estimado. Devido ao erro de câmera menor do que nos conjuntos C1-C4 e ao erro de odometria pequeno a trajetória real do robô e a estimada apresentam poucos desvios.

A simulação C7 é representada pela figura 7.40. Nela, são testados: uma resolução mais alta, e maior erro de deslocamento. Embora haja um erro inicial significativo, a estimação converge à trajetória verdadeira. Já a trajetória do robô é regular e atinge com sucesso o alvo A.

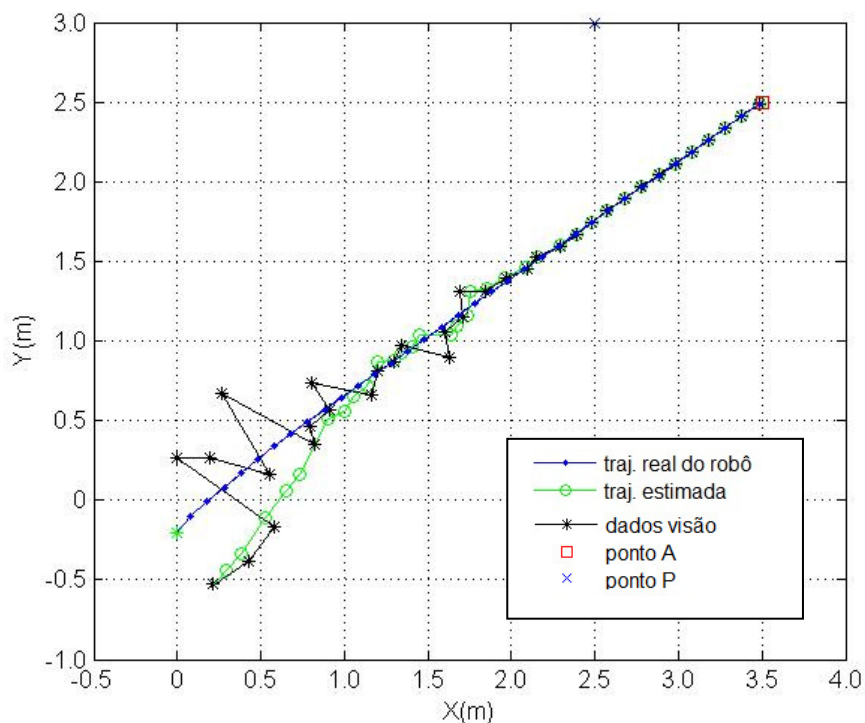


Figura 7. 40- Trajetória de Navegação do Robô com Parâmetros C7

Por último, a figura 7.41 representa o conjunto C8 que difere de C7 pelo ponto inicial da trajetória que é fixo e não estimado pelas câmeras. Devido ao erro maior de deslocamento considerado nesses dois últimos conjuntos de parâmetros, a trajetória estimada sofre pequenos desvios ao longo da primeira metade da trajetória, restabelecendo a estimação precisa na segunda metade do trajeto.

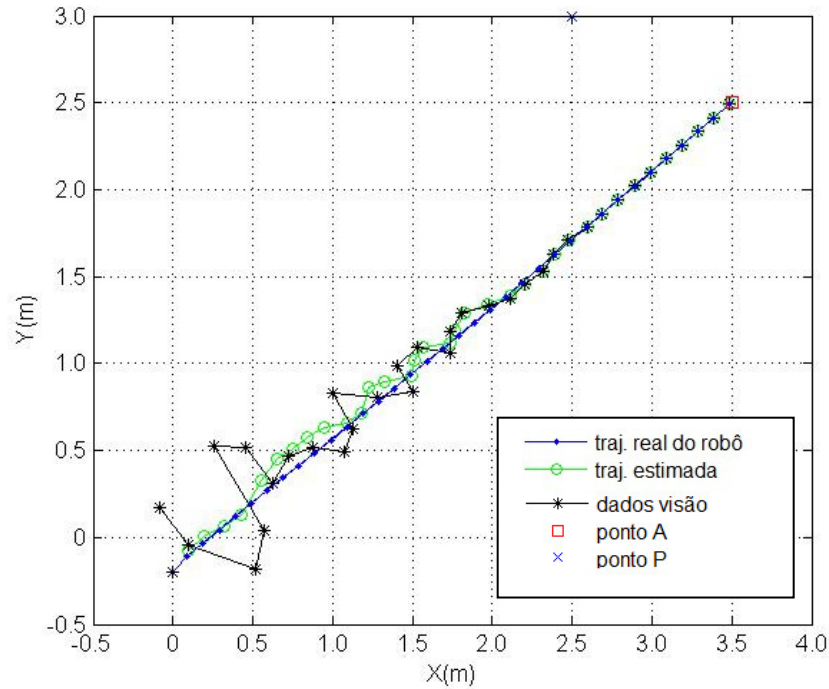


Figura 7. 41- Trajetória de Navegação do Robô com Parâmetros C8

#### 7.4. Discussão dos Resultados

Neste capítulo foram apresentadas simulações realizadas para: métodos de controle, localização através de odometria e visão e estratégias de navegação do robô. Na primeira seção são demonstrados primeiramente simulações de métodos de controle para o modelo de robô na forma de integrador de Brockett. O problema considerado leva o sistema a uma trajetória de referência minimizando um funcional de custo. As figuras 7.1-7.3 mostram simulações desse problema. O método linear *por realimentação de estado*, formulado na

seção 6.1 demonstra eficácia, porém a sua implementação não é possível *on-line*, porque é preciso calcular a solução da equação diferencial de Riccati para todo um intervalo de tempo e somente após, calcular os valores de controle. Isso é adequado em uma aplicação onde as trajetórias de movimento são programadas. Porém para a aplicação de navegação do sistema onde a cada instante do tempo recebe-se *update* de medições de estado, ou seja, uma aplicação *on-line*, isso se torna inviável. Por esse motivo, são apresentadas simulações com controle de SDRE, formulado na seção 5.3, para o modelo de Brockett. As figuras 7.3-7.7 mostram esta situação. O método SDRE apresenta bons resultados, o controle faz o sistema convergir para uma trajetória de referência de forma rápida. Além disso, a comparação de dois métodos, apresentada nas figuras 7.8-7.10 demonstra o funcionamento do controlador linear por realimentação de estado e do controlador SDRE lado ao lado.

O sistema não-holonômico de Brockett, que é equivalente ao modelo cinemático do robô, tem sua relevância por ter sido amplamente estudado, porém para realizar controle deste sistema é preciso transformar o modelo cinemático do robô em sistema de Brockett para depois levar o sistema resultante às coordenadas de desvio. Essas transformações dificultam o controle e a interpretação de resultados. Por isso o modelo dado na seção 2.3 e mostrado neste capítulo nas equações (7.17) é uma alternativa que simplifica o controle do robô móvel. Portanto a partir da seção 7.1.5 esse modelo é considerado em todas as simulações de controle contínuo no tempo.

O fato de que as medições são disponíveis de forma discreta no tempo motiva a formulação do problema do controle discreta na seção 6.4 e as simulações da seção 7.1.4 demonstram o funcionamento desse controle. De uma forma geral, esse funcionamento é adequado. Já as simulações para controlador SDRE, problema de controle não-linear contínuo no tempo e dependente de estado, formulado na seção 6.3, foram apresentadas na seção 7.1.5. As trajetórias de referência consideradas são reta e círculo. Para trajetória de referência linear, foram considerado quatro conjuntos de parâmetros nos quais a coordenada de orientação é variada. As figuras 7.13-7.16 mostram essas simulações. Para trajetória circular apresentam-se também simulações para quatro conjuntos de parâmetros. As figuras 7.17-7.24 demonstram essas simulações. O funcionamento deste método se mostrou bastante eficaz, conduzindo o sistema às trajetórias de referência em pouco tempo.

Na seção 7.2 são realizadas simulações para os métodos de localização odometria e visão de duas câmeras a fim de demonstrar a influência do erro de cada método na estimação de coordenadas de localização. Na seção 7.2.1 os testes de localização simulam o processo de obtenção de coordenadas de uma trajetória de referência através de duas câmeras. Os resultados são mostrados nas figuras 7.26 e 7.27. Constata-se que o erro de estimação é maior quanto mais longe o robô estiver dos pontos de orientação e diminui a medida que este se aproxima destes pontos. Outro parâmetro de influência, é a resolução do sensor das câmeras. Quando utilizada uma resolução típica compatível com a de uma *webcam*, 320 pixels, o erro de estimação de coordenada é grande. Já quando a resolução é dobrada o erro se torna bem menos significativo. Isso ocorre porque há mais precisão na quantização, mapeamento de uma coordenada do mundo real na tela do sensor de cada câmera.

Na seção 7.2.2 os testes de odometria demonstram que quando erro é pequeno a localização ocorre sem grandes desvios (figura 7.28), mas quando o parâmetro de erro é aumentado ocorrem desvios de estimação de trajetória ideal a partir da metade da trajetória (figura 7.29) e têm a tendência de aumentar porque o erro é cumulativo.

O passo seguinte é a elaboração de um sistema simplificado de navegação de robô que une a localização através de odometria, visão de duas câmeras e filtro de Kalman com algoritmos de controle de trajetória. A existência de dois controladores propostos A é uma motivação para se pensar em duas estratégias de navegação do robô. Uma totalmente discreta no tempo e outra mista com controle contínuo e medições discretas, que funciona do esquema de horizonte recuável. As simulações para estratégia discreta são apresentadas na seção 7.3.1, na qual quatro conjuntos de parâmetros simulam o comportamento da navegação do robô diante de valores de erros de medição de estado. As figuras 7.30 a 7.34 demonstram essas simulações. De uma forma geral, a estratégia de navegação é eficaz, mas imprecisões causadas pela visão podem causar desvios dificultando a navegação. O desempenho dessa estratégia é significativamente melhor com resolução maior das câmeras.

A estratégia de horizonte recuável é apresentada na seção 7.3.2 e se mostra bastante adequada para a navegação gerando trajetórias que sempre atingem o alvo, como é mostrado nas figuras 7.35 a 7.42. A leitura da posição inicial pelas câmeras pode provocar grandes erros e tornar imprecisa a estimação do estado do robô. Por esse motivo apresentam-se mais quatro simulações com os mesmos parâmetros de odometria, alterando-se apenas a resolução das câmeras para 570 pixels. Os gráficos 7.39 a 7.42 mostram essas simulações, nas quais independentemente de erros impostos de deslocamento e de câmera a trajetória do robô converge e atinge o alvo.

Em termos de simulações computacionais, as principais contribuições desse trabalho residem nas simulações dos métodos de controle da seção 7.1 e na elaboração e simulação do sistema de navegação, com as simulações da seção 7.3. As simulações da seção 7.2 são auxiliares para demonstrar a influencia dos erros de medição de estado através da odometria e das câmeras nas estratégias de navegação do robô.

## Conclusões e Trabalho Futuro

Nesse trabalho foi realizada, primeiramente uma pesquisa bibliográfica na área de controle de robôs e controle com realimentação de estados. Foi estudado o modelo cinemático do robô de duas rodas com acionamento diferencial e métodos de medição de estado como a odometria e estimação de posição através de visão estereoscópica.

Foram formulados problemas de controle linear ótimo para modelo não-linear , um problema de controle não –linear com o método SDRE- State Dependent Riccati Equation e um problema de controle SDRE para sistemas discretos. Por isso, as contribuições principais desse trabalho consistem em propor e aplicar, no caso do método *por realimentação de estado* linear ótimo e método SDRE e aplicar, no caso do método SDRE, métodos de controle para um sistema não holonômico de robô móvel . Além disso, demonstrar a navegação do robô numa aplicação utilizando visão de duas câmeras e o filtro de Kalman para estimação de estado.

O primeiro método, *por realimentação de estado* linear ótimo de tempo contínuo, é baseado na abordagem semelhante a proposta em Rafikov e Balthazar (2004) e resulta em um controlador que pode ser utilizado para sistemas lineares e também não lineares. Os resultados numéricos foram obtidos para o modelo de Brockett (1983) e são mostrados no capítulo 7 demonstraram. Eles revelam que o controlador é eficaz e eficiente para o problema de regulação a uma trajetória do robô móvel não holonômico.

O método SDRE é uma abordagem atribuída a Cloutier et al. (1996), e tem como base reescrever o sistema na forma dependente de estado, e encontrar a solução da equação

de Riccati dependente de estado o que dá nome ao método. Esse método foi aplicado tanto ao modelo de robô na forma de integrador de Brockett, quanto a um modelo alternativo dado em Kanayama *et al.* (1990). As simulações demonstraram a eficácia e robustez desse método para o problema e ainda a possibilidade da implementação *on line* desse método determinou a sua escolha para a utilização no sistema de navegação.

Duas estratégias de navegação através da localização e controle de trajetória em um ambiente desconhecido foram apresentadas. Elas unem medição de estado através de odometria, visão estereoscópica e controle de trajetória. Uma delas é a estratégia na qual as medições e o controle são discretos no tempo. As simulações computacionais demonstraram que em geral para erros pequenos de medição de estado como acontece em ambientes pequenos a estratégia é eficaz, porém, em certos casos de erros grandes de medição de estado, a trajetória do robô pode sofrer desvios. Outra estratégia é mista, ou seja, as medições de estado são disponíveis de forma discreta e nos intervalos de tempo entre essas medições é realizado um controle contínuo da trajetória do robô. Essa estratégia se mostrou bastante eficaz para a navegação em um ambiente desconhecido.

Em suma, esse trabalho alcançou os objetivos propostos de controlar a trajetória do robô através do controle de realimentação de estado utilizando a visão estereoscópica e a odometria para a estimação de posição. As principais contribuições desse trabalho residem tanto na aplicação dos métodos de controle: linear ótimo por realimentação de estados e não-linear SDRE (State Dependent Riccati Equation), quanto na integração desse controle com informações sobre o posicionamento provenientes da odometria e da visão estereoscópica efetuada através de estratégias propostas de navegação.

No âmbito de trabalhos futuros é preciso destacar a necessidade de utilização de mecanismos de localização no ambiente mais elaborados como correlação de imagens de duas câmeras e outros. Em termos de navegação, é possível programar estratégias mais complexas como localização e mapeamento simultâneo no ambiente. Isso tornará o sistema de navegação do robô mais realístico e mais completo.

E por fim, um trabalho experimental em robô móvel real poderia demonstrar na prática a aplicação do controle e navegação apresentados nesse trabalho.



## Referencias Bibliográficas

- Aguiar A.P. and A. Pascoal. Stabilization of the extended nonholonomic double integrator via logic-based hybrid control: An application to point stabilization of mobile robots. In: SYROCO'00-6th International IFAC Symposium on Robot control. Vienna, Austria, 2000.
- Alekander J.C. and Maddocks J.H., On the kinematics of wheeled mobile robots, Int. J. Robot. Res., 8 (5) (1989) 15-27.
- Anderson B.D.O. & J.B. Moor, Optimal Control: Linear Quadratic Methods, Prentice-Hall, NY, 1990.
- Astolfi A. Discontinuous control of the Brockett integrator. Proceedings of the 36th Conference on Decision & Control, San Diego, California USA, December 1997
- Astolfi A. Discontinuous control of the brockett integrator. Eur J Control 1998;4(1):49–63.
- Astolfi A., Exponential stabilization of wheeled mobile robot via discontinuous control. Journal of Dynamic systems, Measurements and Control, 121 (1999) 121-126.
- Banks S.P. and K. J. Manha, “Optimal Control and Stabilization for Nonlinear Systems”, IMA Journal of Mathematical Control and Information, vol. 9 pp. 179-196, 1992.
- Bernstein, D.S. Nonquadratic cost and nonlinear por realimentação de estado control. Int. J. Robust Nonlinear Contr. 3 (1993) 211-229

- Bloch A and Drakunov S. Stabilization and tracking in the nonholonomic integrator via sliding modes. *Sys Control Lett* 1996;29:91-9
- Bloch AM. *Nonholonomic Mechanics and Control*. Springer-Verlag, Berlin, 2003.
- Brockett RW. Asymptotic stability and feedback stabilization. In: *Differential Geometric Control Theory*. Birkhäuser, Boston, MA, 1983;181-191
- Borenstein, J., Everett, H. R., Feng, L. e Wehe, D. Mobile robot positioning: sensors and techniques, *Journal of Robotic Systems* 4 (14), pp. 231 -249, 1996.
- Bryson A.E. and Y. Ho. *Applied Optimal Control*. Hemisphere Publ. Corp., Washington D.C., 1975.
- Burghart JA. A technique for suboptimal control of nonlinear systems. *IEEE Transaction on Automatic Control* 1969;14:530-3.
- Canudas de Wit, C., & Sordalen, O. J. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control* 1992;37(11):1791-1797.
- Cheng, H. D.; Xu, H. X.; A novel fuzzy logic approach to contrast enhancement, *Pattern Recognition*, vol.33, pp. 809-819, 2000.
- Çimen T. State-dependent Riccati equation (SDRE) control: A survey. In: *Proc. of the 17<sup>th</sup> World Congress of The International Federation of Automatic Control*, Seoul, Korea, July 6-11, 2008.
- Cloutier, JR, D'Souza CN and C.P. Mracek CP. Nonlinear regulation and nonlinear  $H_{\infty}$  control via the state-dependent Riccati equation technique: Part 1, Theory; Part 2, Examples. In: *Proc. of the First International Conference on Nonlinear Problems in Aviation and Aerospace*, Daytona Beach, FL, 117-141, 1996.
- Cloutier J. R. State-dependent Riccati equation techniques: An overview, presented at the

- American Control Conf., Albuquerque, NM, June 1997.
- Cloutier J. R., Mracek C. P., Ridgely D. B., and Hammett K. D., State dependent Riccati equation techniques: Theory and applications,” presented at the Workshop Notes: Amer. Control Conf., Philadelphia, PA, June 1998.
- Cloutier, J.R., Stansbery, D.T. and Szanaier, M. On the recoverability of nonlinear state feedback laws by extended linearization techniques. In: Proc. Of the American Control Conference, San Diego, CA, 1999, 1515-1519.
- Dahleh M, Peirce A, Rabitz HA and Ramakrishna V. Control of molecular motion. Proc of the IEEE 1996;84(1):7-15
- Dixon W.E, Jiang Z.P, Dawson D.M, Global exponential setpoint control of wheeled mobile robots: a Lyapunov approach, Automatica 36 (2000) 1741-1746
- Erdem, E.B., and Alleyne A.G. Experimental real-time SDRE control of an underactuated robot. In: Proc. of the 40th IEEE Conference on Decision and Control, Piscataway, NJ; 2001:219-224.
- Facon, Jaques; Morfologia Matemática: Teoria e Exemplos. 1996
- Fiore, P. D.; Linear Epipolar Algorithm for Multiframe Orientation; IEEE, pp.557-560, 2000;43
- Forsyth, D. A; Ponce, J.; "Computer Vision. A Modern Approach" Prentice Hall, 2003.
- Freeman, R., Primbs, J. Control Lyapunov functions: New ideas from an old source, em “Proceedings of the 35 IEEE Conference on Decision and Control”, pp. 3926-3931, Kobe, Japan, 1996.
- Garrard WL, McClamroch NH and Clark LG. An approach to sub-optimal feedback control of nonlinear systems. International Journal of Control 1967;5:425-35.
- Godhavn, J., e Egeland, M.. A Lyapunov approach to exponential stabilization of

- nonholonomic systems in power form. *IEEE Transactions on Automatic Control* 42(7) (1997) 1028-1032.
- Gordon, R., Ranganyyan, R. M. Feature enhancement of film mammograms using fixed and adaptive neighborhoods , *Appl. Optics*, vol. 23, no. 4, pp. 560- 564, 1984.
- Haddad, W.M. Chellaboina, V.-S. Discret time nonlinear analysis and feedback control with nonquadratic performance criteria. *J. Franklin Inst.* 333B (6) (1996) 849-860.
- Haddad, W.M. Chellaboina, V.-S, Fausz, J.L. Robust nonlinear feedback control for uncertain linear systems with nonquadratic performance criteria. *Systems Control Lett.* 33 (1998)327-338.
- Hammett KD, Hall CD and Ridgely DB. Controllability issues in nonlinear state-dependent Riccati equation control. *AIAA Journal of Guidance, Control, and Dynamics* 1998; 21:767-73.
- Hasan, Y. M. Y.; Karam, L. J. Morphological reversible contour representation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 227- 240, 2000.
- Hespanha J. P. Stabilization of nonholonomic integrators via logic-based switching. In: *Proc. 13th World congress of IFAC*. Vol. E. S. Francisco, CA, pp.467-472, 1996.
- Huang Y and Lu WM. Nonlinear optimal control: alternatives to Hamilton-Jacobi equation. In: *Proc. of the 35th IEEE Conference on Decision and Control*, Kobe, Japan, 3942-3947, 1996.
- Jayaram A and Tadi M. Synchronization of chaotic systems based on SDRE method. *Chaos, Solitons and Fractals* 2006;28:707-715.
- Jiang ZP and Nijmeijer H. A recursive technique for tracking control of nonholonomic systems in chained form, *IEEE Trans. Autom.Control* 1999;44(2):265–279.
- Jiang ZP. Lyapunov design of global state and output feedback trackers for nonholonomic

- control systems, *Int. J. Control*, 2000;73:744–761.
- Kalman, R.E., Bertram, J.E. Control system analysis and design via the second method of Lyapunov. *Jornal of Basis Engineering*, vol. 82, (1960), pp. 371-393.
- Kanayama, Y., Kimura, Y., Miyazaki, F. e Moguchi, T. A Stable Tracking Control Scheme for an Autonomous Mobile Robot. In: *Proc. IEE Internationac. Conf. on Robotics and Automation*, 1990, pag. 384-389.
- Khalil, H. K. (1996). *Nonlinear Systems*, second ed, Prentice Hall, Upper Saddle River, NJ.
- Kim BM & Tsiotras P, Controllers for unicycle-type wheeled robots: theoretical results and experimental validation, *IEEE Trans. on Robotics and automation* 18(3) (2002) 294-307.
- Klancar G., Skrjanc, I. Tracking error model-based predictive controle for mobile robots in real time. *Robotics and Autonomous Systems*, vol. 55 (2007), pp.460-469
- Kolmanovsky, I., & McClamroch, N. H. Developments in nonholonomic control problems. *IEEE Control Systems Magazine* 1995;15:20-36.
- Krassovskii, N.N., On the theory of optimum control, *Appl. Math. Mech.*, vol. 23 (1959), pp. 899-919.
- Kwakernaak, H. e Sivan R., *Linear Optimal Control systems*, John Wiley & Sons, New York, 1972, 575p.
- Letov, A.M. The analytical design of control systems, *Automation and Remote Control*, 22, (1961), 363-372.
- Li Z and Canny J.F. *Nonholonomic motion planning*. Kluwer Academic Publisher, 1993.
- Morgansen KA and Brockett RW. Optimal regulation and reinforcement learning for the nonholonomic integrator. *Proceedings of the 2000 American Control Conference* 2000;1(6):462-6

- Motta, J.M.S.T.; McMaster, R. S.; Experimental Validation of a 3-D Vision-Based Measurement System Applied to Robot Calibration. Journal of the Braz. Soc. Mechanical Sciences, vol. XXIV, p. 234-238, July 2002.
- Mracek CP and Cloutier JR. (1996). Missile longitudinal autopilot design using the state-dependent Riccati equation method. In: Proc. of the International Conference on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, FL, 387-396.
- Mracek, C.P. and J.R. Cloutier. Full envelope missile longitudinal autopilot design using the state-dependent Riccati equation method. In: Proc. of the AIAA Guidance, Navigation, and Control Conference, New Orleans LA;1996:1697-1705.
- Mracek, C.P. and Cloutier, J.R. Control designs for the nonlinear benchmark problem via the state-dependent Riccati equation method. International Journal of Robust and Nonlinear Control, 8, 1998, 401-433.
- Murray RM and Sastry S. Nonholonomic motion planning: Steering using sinusoids. IEEE Trans Aut Cont 1993;38(5):700
- Naik MS and Singh SN. State-dependent Riccati equation-based robust dive plane control of AUV with control constraints. Ocean Engineering 2007;34:1711-1723.
- Neimark JI and Fufaev NA. Dynamics of Nonholonomic Systems. American Mathematical Society, Providence, RI, 1972.
- Oisel, L.; et al; One Dimensional Dense Disparity Estimation for Three-Dimensional Reconstruction; IEEE Transactions on Image Processing, vol 12, no.9,2003.
- Oppenheim, A. V.; Shafer, R. W. Discrete-time signal processing, Englewood Cliffs, NJ: Prentice Hall, 1989.
- Oriolo G., Luca A. de, Vendittelli, M. WMR Control via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation, IEEE Transaction on Control System Tecnology, 10 (6) 2002.

- Pearson, J. D. "Approximation Methods in Optimal Control", Journal of Electronics and Control, vol.13, pp. 453-465, 1962.
- Pomet, J. -B. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *Systems and Control Letters* 1992;18(2):147-158.
- Qu Z, Wang J, Plaisted CE and Hull RA. Global-stabilizing near-optimal control design for nonholonomic chained systems. *IEEE Trans Aut Cont* 2006;51(9):1440-56
- Quak, E.; Weyrich, N. Decomposition and reconstruction algorithms for spline wavelets on a bounded interval, *Applied and Computational Harmonic Analysis*, vol. 1, no. 3, pp. 217-231, 1994.
- Rafikov, M., J.M Balthazar, On a optimal control design for Rössler system, *Phys. Lett. A* 333 (2004) 241-245.
- Rafikov, M., J.M Balthazar, Optimal linear and nonlinear control design for chaotic systems. Proceedings of ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Long Beach, California, USA, September 24–28, 2005.
- Rafikov M.& Balthazar JM. On control and synchronization in chaotic and hyperchaotic systems via linear feedback control, *Commun. Nonlinear. Sci. Numer. Simul.* 13 (2008) 1246–1255.
- Rafikov M, Balthazar JM. Optimal and suboptimal control of nonlinear systems by State Dependent Riccati Equation (SDRE) method In: Proceedings of 8<sup>th</sup> Brazilian Conference on Dynamics, Control and Applications, 2009.
- Reeds J.A. and Shepp L.A. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.* 145(2) (1990) 367-393
- O.E. Rossler, An equation for continuous chaos. *Phys. Lett. A* 57 (1976) 397.

- Salvi, J.; Armangué, X.; Battle, J.; A Comparative Review of Camera Calibrating Methods With Accuracy Evaluation; *Pattern Recognition*, 35, 1617-1635, 2002.
- Samson, C. Velocity and torque feedback control of a nonholonomic cart. *Proceedings of the international workshop in adaptive and nonlinear control: Issues in robotics*. 1990;
- Samson C., Control of Chained systems: Application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control* 40(1) (1995) 64-77. Sontag, E.D., A 'universal' construction of Artstein's theorem on nonlinear stabilization, *Syst. Contr. Lett.*, 13(2), (1989), 117-123.
- Spitz, S. N.; Rechicha, A. A. G. Accessibility analysis using computer graphics hardware, *IEEE Trans. on Visualization and Computer Graphics*, vol. 6, no. 3, pp. 208-219, 2000.
- Stansbery DT and Cloutier JR. Position and attitude control of a spacecraft using the state-dependent Riccati equation technique. In: *Proc. of the American Control Conference*, Chicago, IL;2000:1867-71.
- Sussmann, H. J. (1991). Optimal control and almost analytic feedback for some nonholonomic systems., Technical report, SYCON -Rutgers Center for Systems
- Teel, A. R., Murray, R. M., & Walsh, C. G. Non-holonomic control systems: From steering to stabilization with sinusoids. *International Journal Control* 1995;62(4):849-870.
- Trucco, E; Verri, A.; "Introductory Techniques for 3-D Computer Vision," Prentice Hall, 1998.
- Tsai, R.; Huang, T.; Uniqueness and estimation of three dimensional motion parameters of rigid objects with curved surface. *IEEE Trans. Pattern Anal. Machine Intell.*, vol 6, pp.13-26, 1984.
- Walsh G, Tilbury D, Sastry S, Murray R and Laumond J. Stabilization of trajectories for systems with nonholonomic constraints, *IEEE Trans Auto. Control* 1994;39(1):216–



- Wang, R.-Z.; Lin, C.-F.; Lin, J.-C. Image hiding by optimal LSB substitution and genetic algorithm, *Pattern Recognition*, vol. 34, pp. 671-683, 2001.
- Wernli, A. and Cook. Suboptimal control for the nonlinear quadratic regulator problem. *Automatica*, 11,1975, 75-84.
- Y. Ma, S. Soatto, J. Kosecká, S. S. Sastry, "An Invitation to 3-D Vision From Images to Geometric Models," Series: Interdisciplinary Applied Mathematics, Vol. 26, Springer. 526 p., 2004.
- Yu J, Jadbabaie A, Primbs J and Huang Y. Comparison of nonlinear control design techniques on a model of the Caltech ducted fan. *Automatica* 2001; 37:1971-1978
- Zhang, B.; Fu, M.; Yan, H. A nonlinear neural network model of mixture of local principal component analysis: application to handwritten digits recognition, *Pattern Recognition*, vol. 34, pp. 203-214, 2001.

## Anexo A :Reconstrução de Estado Linear Ótima através do Filtro de Kalman

Nos problemas de regulação e de *tracking* da trajetória do robô assume-se que o estado atual do robô é exato, ou que a medição desse estado é exata e precisa. Isso, na maioria das vezes não é verdadeiro e a medição do estado do robô é imprecisa e dotada de ruído. Nesse sentido é preciso realizar uma reconstrução do estado, que é uma aproximação do estado real do robô em um dado instante de tempo de modo que um estado reconstruído:

$$\hat{x}_{k+1} = \bar{A}\hat{x}_k + \bar{B}u_k + w_k, \quad (1)$$

onde  $\hat{x}_{k+1}$  é o estado reconstruído e A e B são matrizes do sistema Uma vez reconstruído o estado é possível utilizá-lo no problema de controle para a determinação do vetor de estado controlado.

Para a reconstrução do estado, o primeiro problema que surge é a introdução de um observador que não leva em conta explicitamente de ruído, mas procura reconstrução de estado que implicitamente envolvem um certo grau de filtragem (Kwakernaak e Sivan, 1972).

Num segundo momento uma abordagem estocástica do problema leva a um observador ótimo que lida explicitamente e quantitativamente com os erros sistemáticos e ruído de aquisição de medições. Tal observador ótimo é um estimador de estado conhecido como Filtro de Kalman.

### 1. Observadores de estado

Para um dado sistema:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t). \quad (2)$$

Uma combinação linear do estado  $y(t)$  pode ser medida , sendo um vetor  $l$ -dimensional com  $l \leq n$  , com  $n$  dimensão do estado  $x$ , na forma:

$$y(t) = C(t)x(t). \quad (3)$$

Em seguida introduzem-se as definições de Luenberger ( 1966) para um observador:

### **Definição 1**

O sistema:

$$\dot{q}(t) = F(t)q(t) + G(t)y(t) + H(t)u(t)$$

$$z(t) = K(t)q(t) + L(t)y(t) + M(t)u(t) \quad (4)$$

é um observador para o sistema:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t) \quad (5)$$

Se para cada estado inicial  $x(t_0)$  do sistema (5) existe um estado inicial  $q_0$  para o sistema (4) tal que

$$q(t_0) = q_0 \quad (6)$$

Isso implica que, para todo  $u(t)$  :

$$z(t) = x(t), \quad t \geq t_0, \quad (7)$$

quando o estado observado tem a mesma dimensão que o estado  $x$  o observador é pleno (*full-order*) quando observa-se menos variáveis que a ordem do sistema (1) o observador é de ordem reduzida.

## Definição 2

O sistema n-dimensional;

$$\dot{\hat{x}}(t) = F(t)\hat{x}(t) + G(t)y(t) + H(t)u(t) \quad (8)$$

É um observador de ordem plena para o sistema n-dimensional (5) se:

$$\hat{x}(t_0) = x(t_0) \quad (9)$$

Isso implica em, para todo  $u(t)$  :

$$\hat{x}(t) = x(t), \quad t \geq t_0 \quad (10)$$

**Teorema 1** (Kwaakernak e Sivan , 1972)

O sistema (8) é observador para (5) se e somente se:

$$F(t) = A(t) - K(t)C(t),$$

$$G(t) = K(t) ,$$

$$H(t) = B(t), \quad (11)$$

onde  $K(t)$  é uma matriz arbitrária variável no tempo. Como resultado, observadores de ordem plena tem a forma:

$$\dot{\hat{x}}(t) = A(t)\hat{x} + B(t)u(t) + K(t)[y(t) - C(t)\hat{x}(t)]. \quad (12)$$

A prova desse teorema pode ser encontrada em Kwaakernak e Sivan (1972).

A forma (12) do observador também pode ser escrita como:

$$\dot{\hat{x}}(t) = [A(t) - K(t)C(t)]\hat{x} + B(t)u(t) + K(t)y(t) \quad (13)$$

A expressão acima mostra que a estabilidade do observador é determinada pelo comportamento do termo  $A(t) - K(t)C(t)$  e que essa estabilidade implica na propriedade do erro de reconstrução segundo o teorema a seguir.

**Teorema 2:** Considerando o observador (12) para o sistema (5), o erro de reconstrução de estado é dado por:

$$\hat{e}(t) = x(t) - \hat{x}(t) \quad (14)$$

e este satisfaz a equação diferencial:

$$\dot{\hat{e}}(t) = [A(t) - K(t)C(t)]\hat{e}(t). \quad (15)$$

O erro de reconstrução (14) tem a propriedade de

$$\hat{e}(t) \rightarrow 0 \quad \text{quando } t \rightarrow \infty, \quad (16)$$

Para todo  $e(t_0)$ , se e somente se, o observador é assintoticamente estável. Portanto, a escolha da matriz de ganho  $K(t)$  é essencial no projeto do observador.

## 2. Observador de estado ótimo com abordagem estocástica.

Um problema de construção de estimador ótimo pode ser formulado como a construção de um observador de estado pleno que minimiza o efeito combinado de perturbações que atingem a dinâmica do sistema que o ruído que está contido nas medições, obtendo a estimação ótima do estado observado. Kalman em seu trabalho clássico (Kalman, 1960) analisou esse problema para o caso discreto, já para um sistema contínuo no tempo o estudo foi realizado também por Bucy, dando ao filtro resultante o nome de Filtro de Kalman – Bucy. Outras variantes do filtro de Kalman são utilizados atualmente tais como: Filtro de Kalman Extendido (para o sistema discreto no tempo e não-linear), Filtro de Kalman-Bucy extendido (para o sistema contínuo no tempo e não -linear).

A resolução deste problema exige o conhecimento sobre a natureza dos processos estocásticos envolvidos. Quando se trata de um processo estocástico com ruído Gaussiano, o problema de estimação ótima se torna análogo ao Regulador Linear Quadrático (*LQR-Linear Quadratic Regulator*). Esse problema é também chamado na literatura de *LQE-Linear Quadratic Estimation*.

Consideremos um sistema linear, contínuo no tempo, controlado e observável na seguinte forma:

$$\begin{aligned}\dot{x}(t) &= \bar{A}(t)x(t) + \bar{B}(t)u + w(t) \\ y(t) &= C(t)x(t) + \gamma(t),\end{aligned}\tag{17}$$

onde a dinâmica do sistema está sujeita a perturbações aleatórias  $w(t)$  e as medições estão sujeitas a ruído aleatório  $\gamma(t)$ . Eles caracterizam um processo com ruído branco de intensidade:

$$(18)$$

Além disso, o estado inicial  $x(t_0)$  não é correlacionado com  $w(t)$  e  $\gamma(t)$

O sistema de observador de estado tem a forma:

$$\dot{\hat{x}}(t) = \bar{A}(t)\hat{x} + \bar{B}(t)u(t) + K(t)[y(t) - C(t)\hat{x}(t)].\tag{19}$$

O erro de reconstrução é definido por:

$$\hat{e}(t) = x(t) - \hat{x}(t) \quad .\tag{20}$$

A expectância desse erro é:

$$E\{\hat{e}^T(t)L(t)\hat{e}(t)\},\tag{21}$$

onde  $L(t)$  é uma matriz simétrica positivo-definida que é uma medida de quão bem o observador reconstrói o estado do sistema no tempo  $t$ .

O problema de estimação de estado ótimo, então, trata de encontrar função matricial  $K(\tau), t_0 \leq \tau \leq t$  e a condição inicial  $\hat{x}(t_0)$  a fim de minimizar (21).

Caso em que:

$$V(t) > 0, \quad t \geq t_0 \quad (22)$$

o problema de observador ótimo é não singular. Consideramos aqui um problema onde as perturbações do estado e ruídos de observação são processos não correlacionados e assim temos o problema de estimação ótima, ou, filtro de Kalman-Bucy dado pelo seguinte teorema:

### Teorema 3

Considere um problema de observador ótimo (17)-(22). Supondo esse problema como não singular e as perturbações do estado e ruídos de observação como não são correlacionados, a solução é obtida escolhendo-se uma matriz de ganho:

$$K(t) = \bar{P}(t)C^T(t)V^{-1}(t), \quad (23)$$

onde  $\bar{P}$  é a solução da equação de Riccati:

$$\dot{\bar{P}}(t) = \bar{A}(t)\bar{P}(t) + \bar{P}(t)A^T(t) - \bar{P}(t)C^T(t)V^{-1}(t)C(t)\bar{P}(t) + W(t), \quad (24)$$

com condição inicial:

$$\bar{P}(t_0) = \bar{P}_0 \quad (23)$$

A matriz covariância do erro de reconstrução é dada por:

$$E\{\hat{e}^T(t)L(t)\hat{e}(t)\} = \bar{P}(t) \quad (24)$$

### 3. Filtro de Kalman Discreto

A seguir será explicado o filtro de Kalman clássico, ou discreto no tempo que considera um processo de estimação estático, ou seja, não há movimento do robô entre os instantes  $k$  e  $k+1$  de estimação. Portanto, os passos de estimação e de deslocamento do robô segundo um modelo discreto são os mesmos.

Considera-se um sistema linear discreto controlado e observável na seguinte forma:

$$\hat{x}_{k+1} = \bar{A}\hat{x}_k + \bar{B}u_k + w_k \quad (25)$$

$$z_{k+1} = H\hat{x}_k + \gamma_k \quad ,$$

onde a dinâmica do sistema está sujeita a perturbações aleatórias  $w$  e as medições estão sujeitas a ruído aleatório  $\gamma$ . Essas variáveis aleatórias são assumidas como independentes entre si e de distribuição normal:

$$p(w) \sim N(0, W)$$

e

$$p(\gamma) \sim N(0, V) \quad (26)$$

As matrizes  $W$  e  $V$  são matrizes de covariância das perturbações do sistema e do ruído de medições, respectivamente. No filtro de Kalman clássico essas matrizes são dadas como constantes, ou seja, o processo estocástico é estacionário. Nesse trabalho, no entanto, essas matrizes são consideradas como variáveis dependentes do deslocamento do robô móvel.

Define-se  $\hat{x}_k^- \in \mathbb{R}^n$  como estimação de estado *a priori* no instante  $k$ , e  $\hat{x}_k \in \mathbb{R}^n$  é a estimação de estado *a posteriori* da medição  $z_k$ . Pode-se definir o erro de estimativa *a priori* e *a posteriori* em relação a um estado  $x_k$  verdadeiro como:



$$\hat{e}_k^- = x_k - \hat{x}_k^-$$

e

$$\hat{e}_k = x_k - \hat{x}_k . \quad (27)$$

Levando –se em conta o que foi descrito anteriormente para o filtro de Kalman-Bucy, no teorema 3 tem-se que a covariância do erro *a priori* é:

$$P_k^- = E[\hat{e}_k^- \hat{e}_k^{-T}] \quad (28)$$

e a covariância do erro *a posteriori* é:

$$P_k = E[\hat{e}_k \hat{e}_k] . \quad (29)$$

O objetivo principal do filtro é relacionar o estado estimado a priori com uma diferença ponderada entre a medição  $z_k$  e a predição da medição  $H_k \hat{x}_k^-$ :

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H_k \hat{x}_k^-). \quad (30)$$

A diferença é conhecida como inovação de medição ou residual. O resíduo refere-se à discrepância entre a estimativa  $H_k \hat{x}_k^-$  e a estimação real  $z_k$ . Quanto mais o resíduo se aproxima do zero, mais próximas são a estimação teórica e a medição prática.

A matriz  $K_{n \times m}$  é uma matriz de ganho que minimiza o erro de covariância a posteriori (30) . Essa minimização pode ser obtida substituindo:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} . \quad (31)$$

Observando a equação acima é possível ver que se R se aproxima de zero o ganho K pondera o residual mais pesadamente. Ou seja:

$$\lim_{R_k \rightarrow 0} K_k = H^{-1} \quad (32)$$

Por outro lado, se a covariância do estado *a priori* se aproxima de zero, o ganho  $K$  influencia menos o residual:

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad (33)$$

Os passos do algoritmo computacional do filtro de Kalman podem ser agrupados dois grupos distintos de equações: as de predição e de correção. Enquanto as equações de predição projetam estimativas do estado *a priori* pelo modelo do sistema, o estado estimado *a posteriori* é obtido no estágio da correção levando em conta as medições ponderadas pelo ganho  $K$ .

As equações de predição estimativa no tempo projetam o a estimação do estado para o tempo seguinte:

$$\hat{x}_{k+1}^- = \bar{A}\hat{x}_k + \bar{B}u_k, \quad (34)$$

onde  $A$  e  $B$  são matrizes do sistema (5) e  $u_k$  o controle. A covariância do estado estimado *a priori* deve ser calculada por :

$$P_{k+1}^- = \bar{A}P_k A^T + W. \quad (35)$$

A fase de correção inicia com o cálculo da matriz de ganho de Kalman  $K_k$  :

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}. \quad (36)$$

Introduzindo (36) na equação seguinte tem-se o cálculo de estado estimado *a posteriori* que corresponde ao estado estimado pelo filtro de Kalman.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H_k \hat{x}_k^-) \quad (37)$$

A covariância do estado estimado *a posteriori* (37) pode ser calculada de:

$$P_k = (I - K_k H) P_k^- + W. \quad (38)$$

## **Anexos B: Publicações em Anais de Congressos**

# On Linear and Nonlinear Tracking of the Wheeled Mobile Robot

E. Rafikova, *Student Member IEEE*, P.R.G. Kurka, F.L. Pereira, M. Rafikov, *Member IEEE*

**Abstract**—This paper presents two different trajectory tracking control strategies for wheeled mobile robot. The first strategy presents a time-varying linear feedback control law and the second strategy is based on State Dependent Ricatti Equation (SDRE) method. Numerical simulation results indicated that both methods can be successfully used for control of the robot system.

## I. INTRODUCTION

OVER the past two decades the importance of the wheeled mobile robots control has received considerable attention due the growing interest of the robotics use in industry, transportation, exploration. Systems such as mobile robots, robot manipulators, space and underwater robots ([1], [2]), have non integrable constraints and are called nonholonomic systems. As pointed out by Brockett's theorem [3], nonholonomic system cannot be asymptotically stabilized around a fixed point under any smooth (or even continuous) time-independent state feedback control law. Therefore, the growing interest in the investigation of the nonholomic system control (like wheeled mobile robot).

There are several controllers proposed for wheeled mobile robot, most of them based on two main approaches which are posture stabilization and trajectory tracking. The problem of regulation control (or posture stabilization) is to stabilize a robot system at any given point in the state space; while the aim of trajectory tracking is to have the robot system following a reference trajectory.

The stabilization problem has received considerable attention in last decade. (See, for example, paper of Kolmanovsky and McClamroch [4]) The research efforts have been made to develop controllers based on either smooth dynamic feedback or nonsmooth feedback. Astolfi [5], [6], Canudas de Wit and Sordalen [7], Kolmanovsky and McClamroch [4], Morgansen and Brockett [8] and others used discontinuous approaches. In [9] Bloch and Drakunov it is considered a sliding mode approach. Posture stabilization can also be achieved under time-varying

continuous controls. (See, for example, Samson [10], Teel et al. [11], Pomet [12],).

The tracking problem has received less attention. In Walsh et al. [13] a locally exponentially stabilizing control was proposed. A dynamic feedback linearization technique for wheeled mobile robot was presented in Canudas de Wit and Sordalen [7]. Global tracking control laws were proposed in Jiang and Nijmeijer [14], Jiang [15] and Qu [16]. Bloch and Drakunov [9] used sliding mode control for trajectory tracking of Brockett integrator.

In this work two different kinds of trajectory tracking control strategies are presented for the mobile robot system. The first strategy is a time-varying linear feedback control law [17] and the second strategy is based on State Dependent Ricatti Equation (SDRE) method [18]. The wheeled mobile robot considered is the simplest one, of unicycle type. The description of the system and control problem statement is presented in the section 2. The optimal control problem formulation and the description of the optimal linear feedback control method is presented in section 3. The section 4 contains a description of the State Dependent Riccati Equation control method and a suboptimal control problem formulation for robot system. The section 5 is dedicated to the results obtained by numerical simulations of the controlled system for both control strategies. Finally, the concluding remarks follow in the section 6.

## II. MODEL OF THE WHEELED MOBILE ROBOT AND CONTROL PROBLEM FORMULATION

Considering a wheeled mobile robot of the unicycle type, shown in figure 1. The vehicle is equipped with two identical, parallel, nondeformable rear wheels and a front free wheel that have pure rolling and nonslipping contact with the ground contact. This way, the velocity  $v$  of the center of mass of the robot is orthogonal to the rear wheel axis and this center of mass is located in the middle of the axis connecting the rear wheels. It is also assumed that the masses and inertias of the wheels are negligible.

Manuscript received January 22, 2009. This work was supported in part by CAPES and CNPQ

E. Rafikova is graduate student at the UNICAMP- State University of Campinas, Campinas, Brazil (phone: +551930294812 e-mail: elya@fem.unicamp.br)

P. R. G. Kurka is with UNICAMP - UNICAMP- State University of Campinas, Campinas, Brazil email: kurka@fem.unicamp.br

F.L.Pereira is with the University of Porto, Porto, Portugal (email: flp@fe.up.pt).

M. Rafikov is with UFABC- Federal University of ABC, Santo André, São Paulo, Brazil (email: marat9119@yahoo.com.br ).

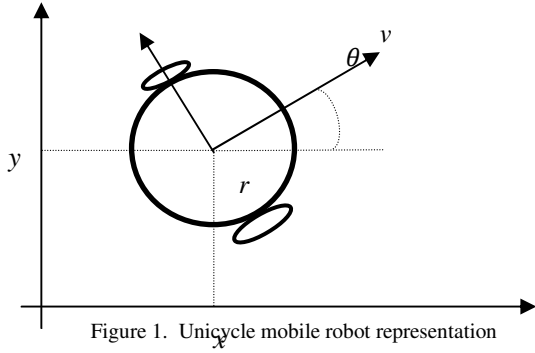


Figure 1. Unicycle mobile robot representation

The kinematics of mobile robot is given by equations:

$$\begin{aligned}\dot{\theta} &= \omega \\ \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta\end{aligned}\quad (1)$$

where  $x$  and  $y$  denote the position of the wheel axis center, and  $\theta$  is the robot orientation with respect to the  $x$ -axis. The symbols  $v$  and  $\omega$  are the linear and angular mobile robot velocities, respectively. A detailed analytical study of the kinematics of wheeled mobile robots is found in [24].

A coordinate change is considered:

$$z = T \begin{bmatrix} \theta \\ x \\ y \end{bmatrix} \quad (2)$$

where  $T \in \mathbb{R}^{3 \times 3}$  is the transformation matrix defined as:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & 2\sin(\theta) - \theta \cos(\theta) & -2\cos(\theta) - \theta \sin(\theta) \end{bmatrix} \quad (3)$$

Leading to the transformed system expressed in coordinates  $z(t) = [z_1(t) \ z_2(t) \ z_3(t)]^T \in \mathbb{R}^3$  with following form:

$$\begin{aligned}\dot{z}_1 &= U_1 \\ \dot{z}_2 &= U_2 \\ \dot{z}_3 &= z_2 U_1 - z_1 U_2\end{aligned}\quad (4)$$

where:

$$\begin{aligned}U_1 &= \omega \\ U_2 &= v - \omega(x \sin \theta - y \cos \theta)\end{aligned}\quad (5)$$

Where  $z \in \mathbb{R}^3$  is a state vector,  $\dot{z} \in \mathbb{R}^3$  is a time derivative of the state vector and  $U \in \mathbb{R}^2$  is a control vector. The model (4) is the Brockett integrator [3], one of the simplest systems with nonholonomic constraint, which is more suitable to subsequent control design. The main goal is to realize the tracking control of this system, by minimizing its deviation from the reference trajectory:

$$\tilde{x} = \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \tilde{x}_3(t) \end{bmatrix} \quad (6)$$

Applying a transformation similar to (3):

$$\tilde{z} = \begin{bmatrix} \tilde{z}_1(t) \\ \tilde{z}_2(t) \\ \tilde{z}_3(t) \end{bmatrix} = \tilde{T} \begin{bmatrix} \tilde{\theta}(t) \\ \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} \quad (7)$$

Where :

$$\tilde{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\tilde{\theta}) & \sin(\tilde{\theta}) \\ 0 & 2\sin(\tilde{\theta}) - \tilde{\theta} \cos(\tilde{\theta}) & -2\cos(\tilde{\theta}) - \tilde{\theta} \sin(\tilde{\theta}) \end{bmatrix} \quad (8)$$

Then, the system (4) can be described by error coordinates  $e(t) \in \mathbb{R}^3$  expressing the difference between the system and a desired trajectory:

$$e = z - \tilde{z} \quad (9)$$

Meanwhile the control vector  $[\tilde{u}_1 \ \tilde{u}_2]^T$  represents a feedforward control which maintains the system at the desired trajectory (7) and satisfies the following equation:

$$\begin{aligned}\dot{\tilde{z}}_1 &= \tilde{u}_1 \\ \dot{\tilde{z}}_2 &= \tilde{u}_2 \\ \dot{\tilde{z}}_3 &= \tilde{z}_1 \tilde{u}_2 - \tilde{z}_2 \tilde{u}_1\end{aligned}\quad (10)$$

The feedback control  $u$  that realizes the tracking control of the system (4) to a trajectory (7) can be expressed as:

$$u = U - \tilde{u} \quad (11)$$

Therefore, error coordinate system is given by

$$\begin{aligned}\dot{e}_1 &= u_1 \\ \dot{e}_2 &= u_2 \\ \dot{e}_3 &= \tilde{u}_2 e_1 - \tilde{u}_1 e_2 + (\tilde{z}_1 + e_1) u_2 - (\tilde{z}_2 + e_2) u_1\end{aligned}\quad (12)$$

### III. OPTIMAL LINEAR STATE FEEDBACK CONTROL PROBLEM FORMULATION

Considering the system :

$$\dot{e} = A(t)e + h(e, u) + B(t)u$$

$$e(0) = e_0 \quad (13)$$

where  $e \in \mathbb{R}^n$  is a state vector,  $A(t) \in \mathbb{R}^{n \times n}$  and  $B(t) \in \mathbb{R}^{n \times m}$  are bounded matrices, whose elements are time dependent,  $u \in \mathbb{R}^m$  is a control vector, and  $h(e, u) \in \mathbb{R}^n$  is a

vector, whose elements are continuous nonlinear functions, with  $h(0,0) = 0$ .

Next, it is presented an important result, concerning a control law that guarantees stability for a nonlinear system and minimizes a nonquadratic performance functional.

*Theorem* : If there exist matrices  $Q(t)$  and  $R(t)$ , both positive definite being  $Q$  symmetric such that there will be a function in form:

$$l(e, u) = e^T Q(e) e - h^T(e, u) P(e) e - e^T P(e) h(e, u), \quad (14)$$

that is positive definite, then the linear feedback control  $u$  :

$$u = -R^{-1} B^T P(t) e \quad (15)$$

is optimal to transfer the system (13) from the initial condition to a final state:

$$e(t_f) = 0, \quad (16)$$

minimizing the functional :

$$J = \int_0^{t_f} [l(e, u) + u^T R(t) u] dt \quad (17)$$

$P(t)$  in eq. (15) is a positive definite symmetric matrix (for all  $t \in [0, t_f]$ ) which is the solution of the matrix differential Riccati equation :

$$\dot{P}(t) + P(t)A(t) + A(t)^T P(t) - P(t)B(t)R(t)^{-1}B(t)^T P(t) + Q(t) = 0, \quad (18)$$

satisfying the final condition:

$$P(t_f) = 0 \quad (19)$$

*Remark 2*: If  $h(e, u) \equiv h(e)$  then the theorem above become the theorem formulated in [17] and their proofs are similar. In the case of Brockett integrator, the system (12) can also be expressed in the form (13), with following matrices values:

$$A(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{u}_2 & -\tilde{u}_1 & 0 \end{bmatrix}, h(e, u) = \begin{bmatrix} 0 \\ 0 \\ e_1 u_2 - e_2 u_1 \end{bmatrix},$$

$$B(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\tilde{z}_2 & \tilde{z}_1 \end{bmatrix} \quad (20)$$

#### IV. SDRE CONTROL PROBLEM FORMULATION

The State-Dependent Riccati Equation (SDRE) strategy has become very popular within the control community over the last decade, providing a very effective algorithm for synthesizing nonlinear feedback controls by allowing nonlinearities in the system states while

additionally offering great design flexibility through state-dependent weighting matrices [19]

This method, first proposed by Pearson [20] and later expanded by Wernli & Cook [21], was independently studied by Mracek & Cloutier [22]. Cloutier, Stansbery & Szaier [23] derived the necessary condition for existence of any gain matrix of SDRE feedback controller. Mracek & Cloutier [22] proved local asymptotic stability for a multivariable case ( $n > 1$ ) system with SDRE feedback controller.

The explanation of the main idea of the method follows ahead.

Consider the general infinite-horizon, input-affine, autonomous, nonlinear regulator problem of the form:

Minimize:

$$J = \frac{1}{2} \int_0^\infty [e^T Q(e) e + u^T R(e) u] dt \quad (21)$$

with respect to the state  $e$  and control  $u$  subject to the nonlinear system constraints

$$\dot{e} = f(e) + g(e)u \quad (22)$$

where  $e \in \mathcal{R}^n$  is a state vector,  $f(e)$ ,  $g(e)$ ,  $u \in \mathcal{R}^n$  and matrices  $Q(e)$  and  $R(e)$  are positive definite for all  $e$ . We assume that  $f(0) = 0$  and that  $g(e) \neq 0$  in a neighborhood of the origin. The SDRE method requires following steps to obtain the suboptimal solution for the control problem (21)-(22) [22]:

- i) Transformation of model (6) to a state dependent coefficient form so it become linear with state dependent coefficients:

$$\dot{e} = A(e)e + B(e)u, \quad (23)$$

where  $f(e) = A(e)e$  and  $B(e) = g(e)$ .

- ii) Solution of the state dependent Riccati equation

$$P(e)A(e) + A(e)^T P(e) - P(e)B(e)R^{-1}(e)B(e)^T P(e) + Q(e) = 0 \quad (24)$$

to obtain  $P \geq 0$ , where  $P$  is a function of  $e$ .

- iii) Construction the nonlinear feedback controller

$$u = -R^{-1}(e)B^T(e)P(e)e \quad (25)$$

The system (4) placed in the state dependent coefficient form yields:

$$\dot{e} = A(e)e + B(e)u, \quad (26)$$

where  $A(e) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{u}_2 & -\tilde{u}_1 & 0 \end{bmatrix}$  and

$$B(e) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -y_2 - \tilde{z}_2 & y_1 + \tilde{z}_1 \end{bmatrix}.$$

## V. NUMERICAL SIMULATION RESULTS

Numerical simulations were made in order to demonstrate the results of both linear and nonlinear trajectory tracking, first, using the optimal time varying control formulation described in section 3 and then, using SDRE control formulation described in section 4. The numerical simulations were performed using the Runge-Kutta fourth order integration method with variable step to solve the system differential equations.

The desired trajectory was chosen as (27) for simulation of both control methods:

$$\tilde{z}(t) = \begin{bmatrix} \pi/4 \\ t \\ \pi t/4 \end{bmatrix} \quad (27)$$

Also the values for matrices  $Q$  and  $R$  were constant for both methods:

$$Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (28)$$

The trajectory started in all simulations from initial

$$\text{conditions : } x_0 = \begin{bmatrix} -\pi/2 \\ -3\pi/4 - 1/2 \\ 3 \end{bmatrix}$$

In figure 2 the there is a time evolution of the error coordinates for the optimal linear regulator method. The system matrices have following values:

$$A(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, h(e, u) = \begin{bmatrix} 0 \\ 0 \\ e_1 u_2 - e_2 u_1 \end{bmatrix},$$

$$B(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -t & \pi/4 \end{bmatrix} \quad (29)$$

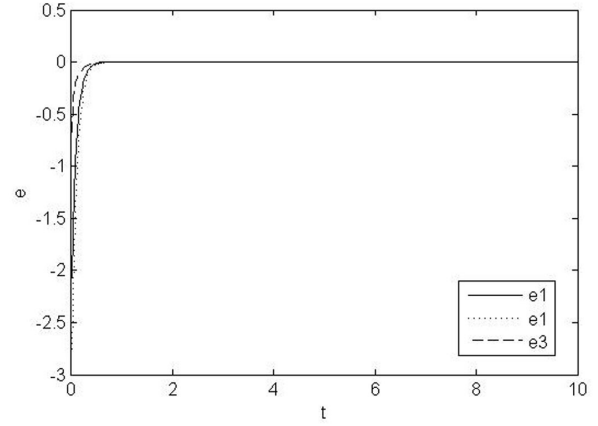


Figure 2. Time evolution of the error system controlled by Optimal Linear Feedback

The figure 3 shows the time evolution of the error system for the second presented method, the nonlinear regulator (SDRE control), where the system matrices have following values:

$$A(y) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, B(y) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -e_2 - t & e_1 + \pi/4 \end{bmatrix} \quad (30)$$

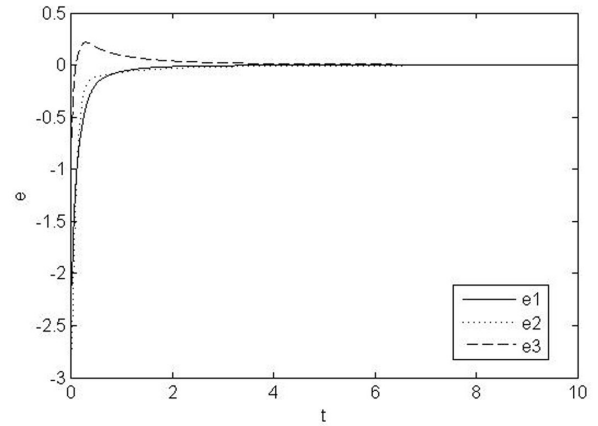


Figure 3. Time evolution of the error system controlled by SDRE method

The figure 4 presents a trajectory of the robot for both presented methods. The dashed line represents the robot trajectory controlled with SDRE method while the solid line in the graphic represents the optimal linear feedback controlled trajectory.

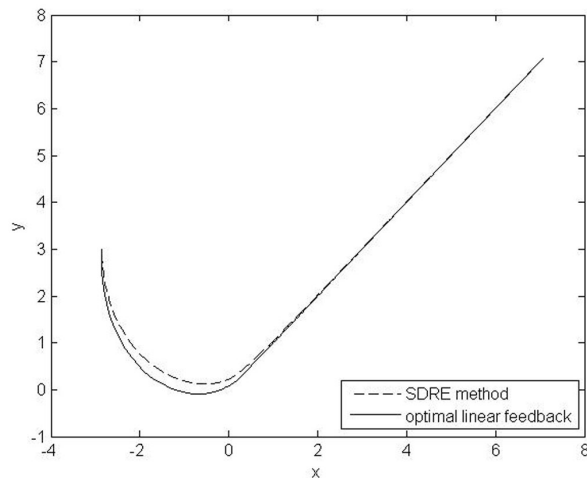


Figure 4. Robot trajectory controlled by different methods

## VI. CONCLUSION

This paper has presented two tracking control strategies for the wheeled mobile robot. The first strategy is an optimal time-varying linear feedback control and the second strategy is based on State Dependent Riccati Equation (SDRE), the suboptimal method. Numerical simulation results have demonstrated that both methods track the system to a chosen reference trajectory in the small amount of time, therefore, both can be successfully applied for trajectory tracking control of the robot system considered. However, the SDRE method has an advantage of an online implementation.

## ACKNOWLEDGMENT

The authors would like to thank CAPES and CNPQ for support on this research

## REFERENCES

- [1] Li Z and Canny J.F. Nonholonomic motion planning. Kluwer Academic Publisher, 1993.
- [2] Murray RM and Sastry S. Nonholonomic motion planning: Steering using sinusoids. *IEEE Trans Aut Cont* 1993;38(5):700
- [3] Brockett RW. Asymptotic stability and feedback stabilization. In: *Differential Geometric Control Theory*. Birkhäuser, Boston, MA, 1983;181-191
- [4] Kolmanovsky, I., & McClamroch, N. H. Developments in nonholonomic control problems. *IEEE Control Systems Magazine* 1995;15:20-36.
- [5] Astolfi A. Discontinuous control of the brockett integrator. *Eur J Control* 1998;4(1):49-63.
- [6] Astolfi A. Discontinuous control of the Brockett integrator. *Proceedings of the 36<sup>th</sup> Conference on Decision & Control*, San Diego, California USA, Desember 1997

- [7] Canudas de Wit, C., & Sordalen, O. J. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control* 1992;37(11):1791-1797.
- [8] Morgansen KA and Brockett RW. Optimal regulation and reinforcement learning for the nonholonomic integrator. *Proceedings of the 2000 American Control Conference* 2000;1(6):462-6
- [9] Bloch A and Drakunov S. Stabilization and tracking in the nonholonomic integrator via sliding modes. *Sys Control Lett* 1996;29:91-9
- [10] Samson, C. Velocity and torque feedback control of a nonholonomic cart. *Proceedings of the international workshop in adaptive and nonlinear control: Issues in robotics*. 1990;
- [11] Teel, A. R., Murray, R. M., & Walsh, C. G. Non-holonomic control systems: From steering to stabilization with sinusoids. *International Journal Control* 1995;62(4):849-870.
- [12] Pomet, J. -B. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *Systems and Control Letters* 1992;18(2):147-158.
- [13] Walsh G, Tilbury D, Sastry S, Murray R and Laumond J. Stabilization of trajectories for systems with nonholonomic constraints, *IEEE Trans Auto. Control* 1994;39(1):216-222.
- [14] Jiang ZP and Nijmeijer H. A recursive technique for tracking control of nonholonomic systems in chained form, *IEEE Trans. Autom. Control* 1999;44(2):265-279.
- [15] Jiang ZP. Lyapunov design of global state and output feedback trackers for nonholonomic control systems, *Int. J. Control*, 2000;73:744-761.
- [16] Qu Z, Wang J, Plaisted CE and Hull RA. Global-stabilizing near-optimal control design for nonholonomic chained systems. *IEEE Trans Aut Cont* 2006;51(9):1440-56
- [17] Rafikov M. & Balthazar JM. On control and synchronization in chaotic and hyperchaotic systems via linear feedback control, *Commun. Nonlinear. Sci. Numer. Simul.* 13 (2008) 1246-1255.
- [18] Cloutier J. R. State-dependent Riccati equation techniques: An overview, presented at the American Control Conf., Albuquerque, NM, June 1997.
- [19] Cimen, T. State Dependent Riccati Equation (SDRE) Control : a Survey, in: *Proceedings of the 17th World Congress The International Federation of Automatic Control* Seoul, Korea, July 6-11, 2008
- [20] Pearson, J.D. Approximation methods in optimal control. *Journal of Electronics and Control*, 13, 1962, 453-469.
- [21] Wernli, A. and G. Cook. Suboptimal control for the nonlinear quadratic regulator problem. *Automatica*, 11, 1975, 75-84
- [22] Mracek, C.P. and J.R. Cloutier. Control designs for the nonlinear benchmark problem via the state-dependent Riccati equation method. *International Journal of Robust and Nonlinear Control*, 8, 1998, 401-433.
- [23] Cloutier, J.R., D.T. Stansbery and M. Sznajder. On the recoverability of nonlinear state feedback laws by extended



linearization techniques. In: Proc. of the American Control Conference, San Diego, CA, 1999, 1515-1519.

[24] Alekander J.C. and Maddocks J.H., On the kinematics of wheeled mobile robots, Int. J. Robot. Res., 8 (5) (1989) 15-27.

**ESDA2008-59167**

## Optimal Linear State feedback Time-Varying Regulator for a Unicycle Mobile Robot

**Elvira Rafikova**

UNICAMP/FEM, Campinas, Brazil,  
 elya@fem.unicamp.br

**Paulo R.G. Kurka**

UNICAMP/FEM, Campinas, Brazil,  
 kurka@fem.unicamp.br

**Marat Rafikov**

UNIJUI/DEFEM, Ijuí, Brazil,  
 rafikov@unijui.edu.br

### ABSTRACT

This paper proposes an optimal time-varying linear state feedback control for wheeled mobile robot of the unicycle type. The control law that stabilizes exponentially the motion of the robot to a given desired trajectory is found, after transformation of the cinematic model of the robot into a well-known Brockett integrator [1]. Numerical simulations are presented in order to demonstrate the effectiveness of the proposed control design.

### INTRODUCTION

Over the past twenty years the importance of wheeled mobile robots increased significantly due the growing interest of its use in industry, transportation, exploration etc... A large number of mechanical systems such as mobile robots, robot manipulators and underwater robots, are of the nonholonomic type, having non-integrable constraints. Despite Brockett's postulation [1] that "nonholonomic systems cannot be stabilized by continuously differentiable time invariant state feedback control laws", efforts are made in order to find a control laws that may work properly with this kind of dynamic systems, in order to bring them to a desired regime. A review on the control of nonholonomic systems is found in [2]. References [3] and [4] present an approach to an exponent stabilization of a wheeled mobile robot to a desired set point. Reference [5] suggests that in nonholonomic systems, regulation to a trajectory is a more appropriate problem to consider than that of stabilization to a point. The linear time-varying feedback law that stabilizes the nonholonomic system locally, in a uniform and asymptotic way, is considered in reference [5].

This paper proposes an optimal time-varying linear state feedback control of a nonholonomic, unicycle wheeled mobile robot. The fact that the wheeled mobile robot can be globally stabilized by a time-varying linear-state feedback controller is a novelty of such an approach.

The organization of the paper is as follows. Section 2 presents a linear design for nonlinear system, as well as a generalization of the control theorem which is found originally in reference [6]. A description of the first-order cinematic model of the wheeled mobile robot, used by many researchers (see references [7], [4], [3], [8], [9]), is presented in section 3. In the same section, the model is transformed into the Brockett integrator (reference [1]), which is suitable to subsequent control design. Section 4 presents the formulation of a linear feedback control problem. Section 5 presents numerical simulation where the system is tracked to a desired trajectory in order to verify the effectiveness of the control design. Concluding remarks are presented in section 6.

### LINEAR DESIGN FOR NON LINEAR SYSTEM

The nonlinear controlled system considered is given by

$$\dot{\mathbf{y}} = \mathbf{A}(t) \mathbf{y} + \mathbf{h}(\mathbf{y}, \mathbf{u}) + \mathbf{B}(t)\mathbf{u}, \quad \mathbf{y}(0) = \mathbf{y}_0, \quad (1)$$

where  $\mathbf{y} \in \mathbb{R}^n$  is a state vector,  $\mathbf{A}(t) \in \mathbb{R}^{n \times n}$  and  $\mathbf{B}(t) \in \mathbb{R}^{n \times m}$  are bounded matrices, whose elements are time dependent,  $\mathbf{u} \in \mathbb{R}^m$  is a control vector, and  $\mathbf{h}(\mathbf{y}, \mathbf{u}) \in \mathbb{R}^n$  is a vector, whose elements are continuous nonlinear functions, with  $\mathbf{h}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$ .

In many problems of physics and engineering, a control law that leads the perturbed system to a desired regimen is sought. These kinds of regimes may be an equilibrium point, a periodic or a non-periodic orbit. Vector  $\mathbf{y}$  described in eq.(1) can be considered as the deviation of the actual trajectory of the system from the desired one.

Next, a control law is presented, that guarantees stability for a nonlinear system and minimizes a non quadratic performance functional.

**Theorem 1.** If there exist matrices  $\mathbf{Q}(t)$ , and  $\mathbf{R}(t)$ , both positive definite, being  $\mathbf{Q}$  symmetric, such that the function

$$l(\mathbf{y}, \mathbf{u}) = \mathbf{y}^T \mathbf{Q} \mathbf{y} - \mathbf{h}^T(\mathbf{y}, \mathbf{u}) \mathbf{P} \mathbf{y} - \mathbf{y}^T \mathbf{P} \mathbf{h}(\mathbf{y}, \mathbf{u}), \quad (2)$$

is positive definite, then the linear feedback control

$$\mathbf{u} = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}(t) \mathbf{y} \quad (3)$$

is optimal, in order to transfer the nonlinear system Eq.(1) from an initial to a final state

$$\mathbf{y}(t_f) = \mathbf{0}, \quad (4)$$

minimizing the functional

$$J = \int_0^{t_f} [l(\mathbf{y}) + \mathbf{u}^T \mathbf{R} \mathbf{u}] dt, \quad (5)$$

where  $\mathbf{P}(t)$ , the positive definite symmetric matrix (for all  $t \in [0, t_f]$ ), is the solution of the matrix differential Riccati equation:

$$\dot{\mathbf{P}} + \mathbf{P} \mathbf{A} + \mathbf{A}^T \mathbf{P} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = \mathbf{0}, \quad (6)$$

satisfying the final condition

$$\mathbf{P}(t_f) = \mathbf{0}. \quad (7)$$

Further to the feedback control Eq.(3), there exists a neighborhood  $\mathbf{\Gamma}_0 \subset \mathbf{\Gamma}$ ,  $\mathbf{\Gamma} \subset \mathfrak{R}^n$ , of the origin, such that if  $\mathbf{y}_0 \in \mathbf{\Gamma}_0$ , then the solution  $\mathbf{y}(t) = \mathbf{0}$ ,  $t \geq 0$ , of the controlled system (1) is locally asymptotically stable, and  $J_{\min} = \mathbf{y}_0^T \mathbf{P}(0) \mathbf{y}_0$ .

Finally, if  $\mathbf{\Gamma} = \mathfrak{R}^n$  then the solution  $\mathbf{y}(t) = \mathbf{0}$ ,  $t \geq 0$ , of the controlled system Eq.(1) is globally asymptotically stable.

**Proof.** Consider the linear feedback control Eq.(3) with matrix  $\mathbf{P}$  determined by equation (6) which transfers the nonlinear system Eq.(1) from an initial to a final state Eq.(4), minimizing the functional Eq.(5) where the function  $l(\mathbf{y}, \mathbf{u})$  needs to be determined.

It is known, according to Dynamic Programming rules [10], that if the minimum of functional (5) exists, and if  $V$  is a smooth function of the initial conditions, then it satisfies the following Hamilton – Jacobi – Bellman equation:

$$\min_u \left( \frac{dV}{dt} + \mathbf{y}^T \tilde{\mathbf{Q}} \mathbf{y} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right) = 0. \quad (8)$$

Considering a Lyapunov function

$$V = \mathbf{y}^T \mathbf{P}(t) \mathbf{y}, \quad (9)$$

where  $\mathbf{P}(t)$  is a symmetric positive definite matrix that satisfies the differential Riccati equation (6).

Note that the derivative of function  $V$ , evaluated in the optimal trajectory with control given by Eq.(3) is :

$$\begin{aligned} \dot{V} = \dot{\mathbf{y}}^T \mathbf{P} \mathbf{y} + \mathbf{y}^T \dot{\mathbf{P}} \mathbf{y} + \mathbf{y}^T \mathbf{P} \dot{\mathbf{y}} = & [\mathbf{y}^T \mathbf{A}^T + \\ & + \mathbf{h}^T(\mathbf{y}, \mathbf{u}) - \mathbf{y}^T \mathbf{P} \mathbf{B} (\mathbf{R}^{-1})^T \mathbf{B}^T] \mathbf{P} \mathbf{y} + \mathbf{y}^T \dot{\mathbf{P}} \mathbf{y} + \\ & + \mathbf{y}^T \mathbf{P} [\mathbf{A} \mathbf{y} + \mathbf{h}(\mathbf{y}, \mathbf{u}) - \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \mathbf{y}]. \end{aligned} \quad (10)$$

Substituting  $\dot{V}$  in the Hamilton – Jacobi – Bellman equation (8) leads to:

$$\begin{aligned} \mathbf{y}^T [\dot{\mathbf{P}} + \mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P}] \mathbf{y} + \mathbf{h}^T(\mathbf{y}, \mathbf{u}) \mathbf{P} \mathbf{y} + \\ + \mathbf{y}^T \mathbf{P} \mathbf{h}(\mathbf{y}, \mathbf{u}) + l(\mathbf{y}, \mathbf{u}) = 0. \end{aligned} \quad (11)$$

Then ,

$$l(\mathbf{y}, \mathbf{u}) = \mathbf{y}^T \mathbf{Q} \mathbf{y} - \mathbf{h}^T(\mathbf{y}, \mathbf{u}) \mathbf{P} \mathbf{y} - \mathbf{y}^T \mathbf{P} \mathbf{h}(\mathbf{y}, \mathbf{u}). \quad (12)$$

Note that for the positive definite function  $l(\mathbf{y}, \mathbf{u})$  and positive definite matrix  $\mathbf{R}$ , the derivative of function (9), evaluated in optimal trajectories of system Eq. (1), is given by

$$\dot{V} = -l(\mathbf{y}, \mathbf{u}) - \mathbf{u}^T \mathbf{R} \mathbf{u}, \quad (13)$$

which is negative definite. This way, expression (9) is a Lyapunov function, and the controlled system Eq.(1) is locally asymptotically stable. Integrating the derivative of the Lyapunov function (9) given by

$$\dot{V} = -\mathbf{y}^T \tilde{\mathbf{Q}} \mathbf{y} - \mathbf{u}^T \mathbf{R} \mathbf{u}$$

along the optimal trajectory, leads to  $J_{\min} = \mathbf{y}_0^T \mathbf{P}(0) \mathbf{y}_0$ .

Finally, if  $\mathbf{\Gamma} = \mathfrak{R}^n$ , the global asymptotic stability follows as a direct consequence of the radial unboundness condition for the Lyapunov function (9), that is,  $V(\mathbf{y}) \rightarrow \infty$  as  $\|\mathbf{y}\| \rightarrow \infty$ .

According to the optimal control theory of linear systems with quadratic functional [11], it can be remarked that the solution of the nonlinear differential Riccati equation (6) is a positive definite symmetric matrix for all  $t \in [0, t_f]$ , given the positive definite matrix  $\mathbf{R}$  and the positive semidefinite matrix

**Q.** This concludes the Theorem proof. When the function  $h(\mathbf{y}, \mathbf{u}) = h(\mathbf{y})$ , is not dependent on  $\mathbf{u}$ , then, theorem 1 is equivalent to the theorem presented in reference[6].

## A MODEL OF WHEELED MOBILE ROBOT

Consider the wheeled mobile robot of the unicycle type, shown in Fig. 1. The vehicle is equipped with two identical, parallel, and nondeformable rear wheels which are controlled independently by motors, and a front free wheel. It is assumed that the plane of each wheel is perpendicular to the ground and that the contact between wheels and ground is pure rolling and nonslipping. This way, the velocity  $v$  of the center of mass of the robot is orthogonal to the rear wheel axis and this center of mass is located in the middle of the axis connecting the rear wheels. It is also assumed that the masses and inertias of the wheels are negligible.

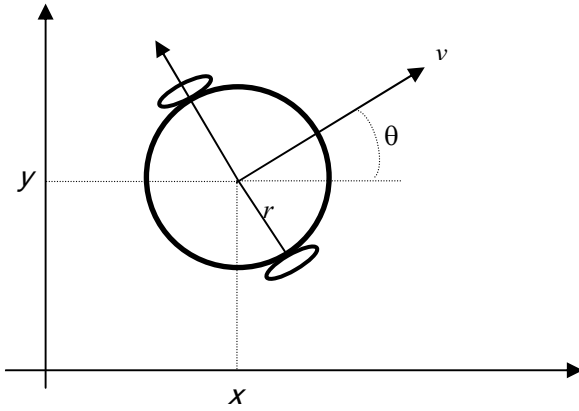


Fig 1.- A wheeled mobile robot of unicycle type

The kinematics of the mobile robot is given by the equations

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega\end{aligned}\quad (14)$$

where  $x$  and  $y$  denote the position of the wheel axis center, and  $\theta$  is the robot orientation with respect to the  $x$ -axis. The symbols  $v$  and  $\omega$  are the linear and angular mobile robot velocities, respectively. This model is the first-order kinematics model of the wheeled mobile robot, used by many researchers (see references [7], [4], [3], [8], [9]). A detailed analytical study of the kinematics of wheeled mobile robots is found in [12].

A coordinate change is considered, where:

$$\mathbf{z} = \mathbf{T} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (15)$$

with  $\mathbf{T} \in \mathcal{R}^{3 \times 3}$  being the transformation matrix defined as:

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 1 \\ \cos \theta & \sin \theta & 0 \\ -\theta \cos \theta + 2 \sin \theta & -\theta \sin \theta - 2 \cos \theta & 0 \end{bmatrix}. \quad (16)$$

It follows that Eq.(15) and Eq. (16) can be transformed from the coordinate system described in (14), to a new one, expressed in  $\mathbf{z}(t) = [z_1(t) \ z_2(t) \ z_3(t)]^T \in \mathcal{R}^3$ .

Transformations lead to a chained form system defined by:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ z_2 U_1 - z_1 U_2 \end{bmatrix} \quad (17)$$

Where

$$U_1 = \omega$$

and

$$U_2 = v - (\tilde{x} \sin \theta - \tilde{y} \cos \theta) \quad (18)$$

The model described by Eq. (17) is the Brockett integrator [1] which is more suitable to subsequent control design.

## CONTROL PROBLEM STATEMENT

In this section, we consider an optimal control problem of the system described in Eq.(17) in order to reach a desired trajectory given by the generic state vector

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \\ \tilde{\theta}(t) \end{bmatrix} \quad (19)$$

In terms of the system Eq.(17) the desired trajectory is:

$$\tilde{\mathbf{z}} = \begin{bmatrix} \tilde{z}_1(t) \\ \tilde{z}_2(t) \\ \tilde{z}_3(t) \end{bmatrix} = \tilde{\mathbf{T}} \begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \\ \tilde{\theta}(t) \end{bmatrix} \quad (20)$$

where:

$$\tilde{\mathbf{T}} = \begin{bmatrix} 0 & 0 & 1 \\ \cos(\tilde{\theta}) & \sin(\tilde{\theta}) & 0 \\ -\tilde{\theta}\cos(\tilde{\theta}) + 2\sin(\tilde{\theta}) & -\tilde{\theta}\sin(\tilde{\theta}) - 2\cos(\tilde{\theta}) & 0 \end{bmatrix}$$

The control vector  $\tilde{\mathbf{u}} = [\tilde{u}_1 \ \tilde{u}_2]^T$  which maintains the system described by Eq.(17) at the desired steady state given at Eq.(20), satisfies the following equation:

$$\begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \tilde{z}_3 \end{bmatrix} = \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ z_2\tilde{u}_1 - z_1\tilde{u}_2 \end{bmatrix} \quad (21)$$

The error vector  $\mathbf{e}(t) \in \mathcal{R}^3$  is given as a difference between the transformed system  $\mathbf{z}(t) \in \mathcal{R}^3$  and a desired trajectory  $\tilde{\mathbf{z}}(t) \in \mathcal{R}^2$  that is,

$$\mathbf{e} = \mathbf{z} - \tilde{\mathbf{z}} \quad (22)$$

and

$$\mathbf{u} = \mathbf{U} - \tilde{\mathbf{u}} \quad (23)$$

The error system can be also expressed by subtracting Eq.(21) from Eq.(17) and taking into consideration Eq.(22) and Eq. (23), yielding

$$\dot{\mathbf{e}} = \mathbf{A}(t)\mathbf{e} + \mathbf{h}(\mathbf{e}, \mathbf{u}) + \mathbf{B}(t)\mathbf{u} \quad (24)$$

where:

$$\mathbf{A}(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{u}_2 & -\tilde{u}_1 & 0 \end{bmatrix}, \quad \mathbf{h}(\mathbf{e}, \mathbf{u}) = \begin{bmatrix} 0 \\ 0 \\ e_1 u_2 - e_2 u_1 \end{bmatrix}$$

and

$$\mathbf{B}(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\tilde{z}_2 & \tilde{z}_1 \end{bmatrix}, \quad (25)$$

with  $\dot{\mathbf{e}} \in \mathcal{R}^3$  being the time derivative of  $\mathbf{e} \in \mathcal{R}^3$  and  $\mathbf{u} \in \mathcal{R}^2$  the control vector. An optimal control problem can be established as follows.

Find the optimal control

$$\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}(t)^T \mathbf{P}(t)\mathbf{e} \quad (26)$$

that leads the system Eq.(24) from an initial to the final state  $\mathbf{e}(t_f) = 0$ , minimizing the functional

$$J = \int_0^{t_f} [l(\mathbf{e}, \mathbf{u}) + \mathbf{u}^T \mathbf{R} \mathbf{u}] dt. \quad (27)$$

where  $\mathbf{P}(t)$  the symmetric positive definite matrix is the solution of the matrix differential Riccati equation:

$$\dot{\mathbf{P}} + \mathbf{P}\mathbf{A} + \mathbf{A}^T \mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0 \quad (28)$$

This way, applying Theorem 1, the control vector (Eq.(26)) is obtained from the solution of Eq.(28).

## NUMERICAL SIMULATIONS

Numerical simulations are performed using the Runge-Kutta fourth order integration method with variable step. The desired trajectory of the system (Eq.(12)) is defined as:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{bmatrix} = \begin{bmatrix} t \\ t \\ \frac{\pi}{4} \end{bmatrix} \quad (29)$$

Such a choice is inspired by the work of Reeds and Shepp [13], that shows that time-optimal paths for this types of robot consist of straight-line segments and arcs of circles.

The correspondent desired trajectory of the system given by Eq.(17) is

$$\tilde{\mathbf{z}}(t) = \begin{bmatrix} \frac{\pi}{4} \\ \sqrt{2}t \\ -\frac{\pi\sqrt{2}}{4}t \end{bmatrix}. \quad (30)$$

Besides, it can be said that

$$\tilde{\mathbf{u}} = \begin{bmatrix} 0 \\ \sqrt{2} \end{bmatrix} \quad (31)$$

Matrix  $\mathbf{Q}$  and  $\mathbf{R}$  were chosen as:

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \text{ and } \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Firstly the numerical solution for differential Riccati equation is found. The time-varying Riccati coefficients are present in Fig. 2.

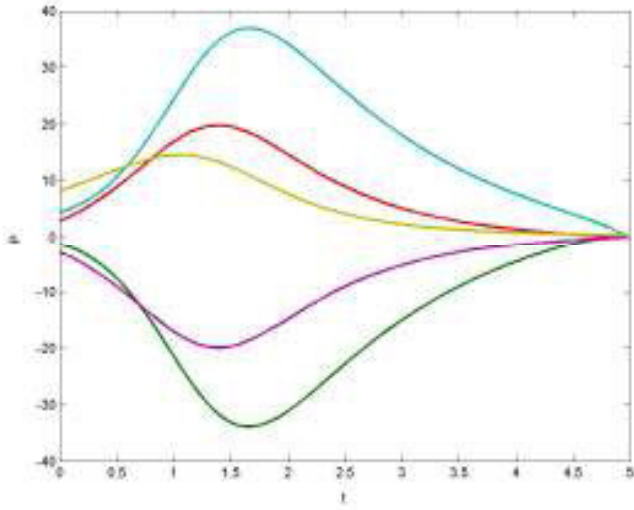


Fig. 2. Time-varying Riccati coefficients

Next, application of control Eq. (25) yields the error trajectory shown in Fig. 3, for the initial conditions  $\theta_0 = -\pi/4$ ,  $x_0 = 1$ ,  $y_0 = 0$ .

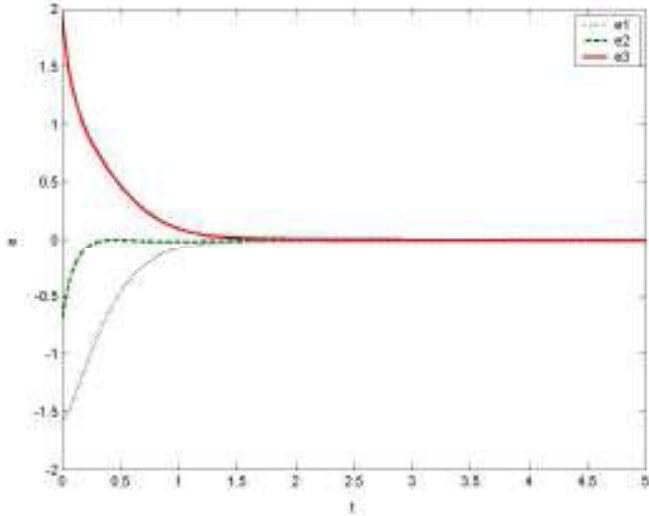


Fig. 3. Time traces of the controlled error system

The error converges to zero after 2 units of time, as seen in Fig. 3.

Integration of the system given by Eq. (17), which is controlled by the relation

$$\mathbf{U} = \mathbf{u} + \tilde{\mathbf{u}}, \quad (32)$$

and admitting the following transformations

$$\theta = z_1,$$

$$x = z_2 \cos \theta + \frac{1}{2}(z_1 z_2 - z_3) \sin \theta$$

and

$$y = z_2 \sin \theta - \frac{1}{2}(z_1 z_2 - z_3) \cos \theta \quad (33)$$

leads to the time trajectories and phase portrait of the original controlled system, shown in Fig.4 and 5, respectively.

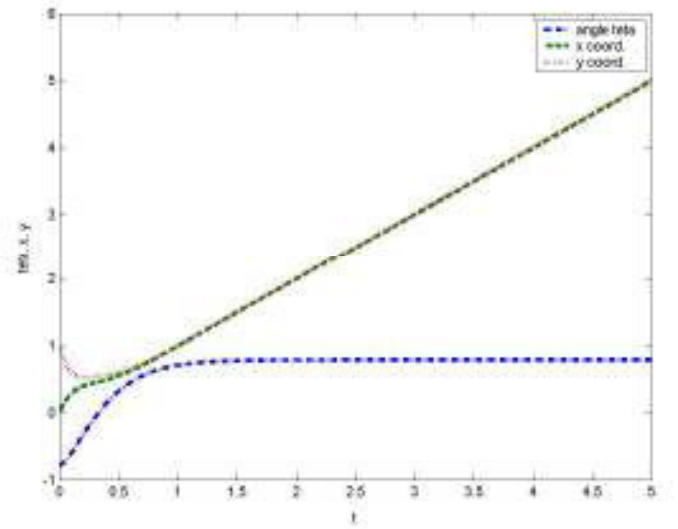


Fig. 4. Time trajectories of the wheeled mobile robot

In the above figure, orientation of the vehicle is represented by the dash-dotted line. The dotted and dashed lines represent the  $x$  and  $y$  coordinates, respectively. It can be seen from this figure that  $x$  and  $y$  coordinate converge to a desired trajectory, and the angular orientation of the robot ( $\theta$ ) remains constant.

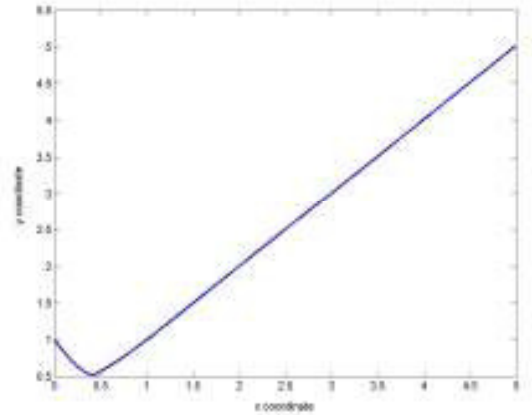


Fig. 5. Phase portrait of the wheeled mobile robot

This control method also presents a satisfactory result in the difficult case when robots orientation is set backwards which correspond to the following initial conditions:

$\theta_o = \pi, x_o = 0, y_o = 1$ . Figure 6 illustrates such a situation.

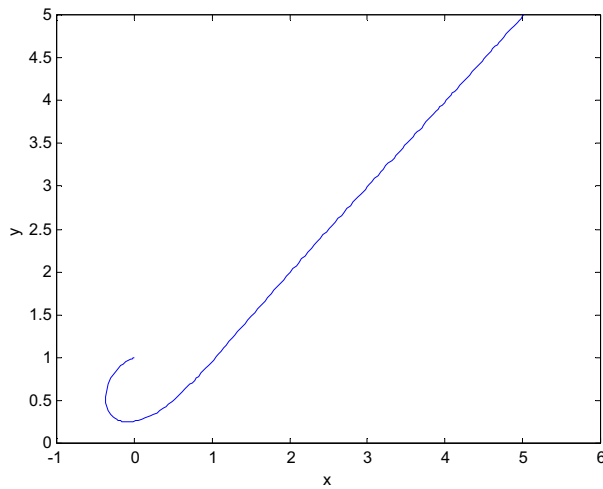


Fig. 6. Phase portrait of the wheeled mobile robot

## CONCLUSIONS

In this paper, an optimal linear feedback control approach to a nonholonomic wheeled mobile robot is considered. the wheeled mobile robot model is reduced to a well-known Brockett problem [1], by the transformation given by Eq. (15), for which no stabilizing static state-feedbacks exists. Reference [8] remarks that the time varying controllers proposed in [14] have typically slow rates of convergence. The proposed optimal linear time-varying feedback control, however, shows through the numerical results, fast rates of convergence of the system to a desired trajectory. This is observed even in a case where the robot orientation is set backwards. The control law presented in this paper is robust to initial conditions because the Lyapunov function (9) provides asymptotic stability when the condition of theorem 1 is satisfied.

According to Walsh et.al. [5], the procedure to establish a composite controller for mobile robots has at least three steps which are: the use of path planners to generate a nominal open-loop trajectory; stabilization of the system to this nominal trajectory; the use of sensor data to avoid collisions and to plan new feasible nominal trajectory during operation of the robot. This way, the control law presented in this paper can be applied in second step of controller described above which is the stabilization the mobile robot system to a nominal trajectory.

## ACKNOWLEDGMENTS

The first author would like to thank CNPQ and CAPES for the support on this research.

## NOMENCLATURE

- $\mathbf{A}(t)$  – system matrix
- $\mathbf{B}(t)$  - control matrix
- $J$  – cost functional
- $\mathbf{P}(t)$ - symmetric positive definite matrix of differential Riccati equation solutions
- $V(\mathbf{y})$ - Lyapunov function
- $\mathbf{e}(t)$  - error signal
- $\mathbf{h}(\mathbf{y}, \mathbf{u})$  - vector of continuous nonlinear functions
- $r$ - radius of robot
- $t$  – time
- $\mathbf{u}$ - control vector,
- $\mathbf{x}$ - position coordinate
- $\tilde{\mathbf{x}}$  - desired trajectory position coordinate
- $\mathbf{y}$  - a state vector
- $\tilde{\mathbf{y}}$  - desired trajectory position coordinate
- $\tilde{\mathbf{z}}$  - desired trajectory in transformed coordinates
- $\omega$  - angular velocity of the robot
- $\theta$  - robot's orientation coordinate
- $v$  - linear velocity of the robot

## REFERENCES

- [1] Brockett, R. Asymptotic stability and feedback stabilization. In: R. Brockett, R. Millman & H. Sussmann, eds, Differential Geometric Control Theory, Boston, Birkhauser, 1983, pp.181-191.
- [2] Kolmanovsky I. & N.H. McClamroch, Developments in nonholonomic control problems. IEEE Control Systems Magazine 15 (1995) 20-36.
- [3] Dixon W.E, Jiang Z.P, Dawson D.M, Global exponential setpoint control of wheeled mobile robots: a Lyapunov approach, Automatica 36 (2000) 1741-1746.
- [4] Canudas de Wit C., Sördaalen, Exponential Stabilization of mobile robots with nonholonomic constraints. IEEE Transactions on Automatic Control 37 (11) (1992) 1791-1797.
- [5] Walsh G., Tilbury D., Sastry S., Murray R. and Laumond J.P. Stabilization of trajectories for systems with nonholonomic constraints. IEEE Transactions on Automatic Control 39(1) (1994) 216-222.
- [6] Rafikov M, Balthazar JM, On control and synchronization in chaotic and hyperchaotic systems via linear feedback control, Communications in Nonlinear Science and Numerical Simulation, 13 (2008) 1246–1255

- [7] Astolfi A., Exponential stabilization of wheeled mobile robot via discontinuous control. *Journal of Dynamic systems, Measurements and Control*, 121 (1999) 121-126.
- [8] Kim BM & Tsiotras P, Controllers for unicycle-type wheeled robots: theoretical results and experimental validation, *IEEE Trans. on Robotics and automation* 18(3) (2002) 294-307.
- [9] Oriolo G., Luca A. de, WMR Control via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation, *IEEE Transaction on Control System Tecnology*, 10 (6) 2002.
- [10] Bryson A.E. and Y. Ho. *Applied Optimal Control*. Hemisphere Publ. Corp., Washington D.C., 1975
- [11] Anderson B.D.O. & J.B. Moor, *Optimal Control : Linear Quadratic Methods*, Prentice-Hall, NY, 1990.
- [12] Alekander J.C. and Maddocks J.H., On the kinematics of wheeled mobile robots, *Int. J. Robot. Res.*, 8 (5) (1989) 15-27.
- [13] Reeds J.A. and Shepp L.A. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.* 145(2) (1990) 367-393
- [14] Samson C., Control of Chained systems: Application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control* 40(1) (1995) 64-77.
- [15] Godhavn J. M. and Egeland. A Lyapunov approach to exponential stabilization of nonholonomic systems in power form. *IEEE Transactions on Automatic Control* 42(7) (1997) 1028-1032.





## OPTIMAL LINEAR STATE FEEDBACK TIME-VARYING CONTROL OF A WHEELED MOBILE ROBOT

*Elvira Rafikova*<sup>1</sup>, *Paulo R.G. Kurka*<sup>1</sup>, *Marat Rafikov*<sup>2</sup>

<sup>1</sup>UNICAMP/FEM, Campinas, Brazil, [elviraelya@yahoo.com.br](mailto:elviraelya@yahoo.com.br),  
[kurka@fem.unicamp.br](mailto:kurka@fem.unicamp.br)

<sup>2</sup>UNIJUI/DEFEM, Ijuí, Brazil, [rafikov@unijui.edu.br](mailto:rafikov@unijui.edu.br)

### 1. INTRODUCTION

Over the past twenty years the importance of wheeled mobile robot (WMR) increased significantly due its intense the growing interest of its use in industry, transportation, exploration. However large number of mechanical systems such as mobile robots, robot manipulators, underwater robots have non-integrable constraints, so called nonholonomic systems.

Great efforts are made on the research on control of non-holonomic systems due their challenging nature: non-integrable constraints, non-transformable into linear control problem systems and Brockett limitation [5]: *nonholonomic systems cannot be stabilized by continuously differentiable time invariant state feedback control laws*. To overcome this limitation there are several approaches for stabilization of nonholonomic control systems to an equilibrium points, among them are smooth time-varying controllers [9], [16], discontinuous or piecewise smooth control laws [4], [7] and hybrid controllers [1], [10]. A review if achievements on control of nonholonomic systems can be found in [12]. In [7], [8] is presented a global approach to an exponent stabilization of WMR to a desired set point. In [1] an exponential stabilization is done by time-varying state-feedback controls which have oscillatory functions in its forms. However this control does not provide robustness to data noise and model uncertainties which may prevent the system from converging.

In this paper it is proposed an optimal time-varying linear state feedback control of the nonholonomic system consisting of unicycle wheeled mobile robot. The fact that the wheeled mobile robot can be stabilized by time-varying linear state feedback controller is a novelty of this approach.

The organization of this paper is as follows. It is presented, in section 2 a linear design for nonlinear system as well as a generalization of the control theorem which can be originally found in [15]. Then, in section 3, there is a description of the so-called first-order kinematic model of the wheeled mobile robot, which was used by many researcher (see, for example, papers [4], [7], [8], [11], [14]). In same section, this model was transformed to famous Brockett integrator [5] that is more comfortable to subsequent control design. In section 4, there is a formulation of a linear feedback control problem. Section 5 presents numerical simulation where the system is tracked to

a desired trajectory in order to verify the effectiveness of the control design. Finally concluding remarks are given in 6.

### 2. LINEAR DESIGN FOR NONLINEAR SYSTEM

We consider the nonlinear controlled system

$$\dot{y} = A(t)y + h(y, u) + B(t)u, \quad y(0) = y_0 \quad (1)$$

where  $y \in \mathbb{R}^n$  is a state vector,  $A(t) \in \mathbb{R}^{n \times n}$  and  $B \in \mathbb{R}^{n \times m}$  are bounded matrices, which elements are time dependent,  $u \in \mathbb{R}^m$  is a control vector, and  $h(y, u) \in \mathbb{R}^n$  is a vector, which elements are continuous nonlinear functions,  $h(0, 0) = 0$ .

We know that in several problems of physics and engineering science the goal is to find a control law  $u$  that leads the perturbed system to a desired one. These kinds of regimes may be an equilibrium point or a periodic or non-periodic orbit. The vector  $y$  of the system (1) can be considered as the deviation of the perturbed trajectory of system from the desired one.

Next, we present an important result, concerning a control law that guarantees stability for a nonlinear system and minimizes a nonquadratic performance functional.

**Theorem 1.** If there exist matrices  $Q(t)$ , and  $R(t)$ , positive definite, being  $Q$  symmetric, such that the function

$$l(y, u) = y^T Q y - h^T(y, u) P y - y^T P h(y, u), \quad (2)$$

is positive definite then the linear feedback control

$$u = -R^{-1} B^T P(t) y \quad (3)$$

is optimal, in order to transfer the nonlinear system (1) from an initial to final state

$$y(t_f) = 0 \quad (4)$$

minimizing the functional

$$(5)$$

where  $P(t)$  the symmetric, positive definite matrix (for all  $t \in [0, t_f]$ ), is the solution of the matrix differential Riccati equation

$$\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (6)$$

satisfying the final condition

$$P(t_f) = 0 \quad (7)$$

In addition, with the feedback control (3), there exists a neighborhood  $\Gamma_0 \subset \Gamma$ ,  $\Gamma \subset \mathfrak{R}^n$ , of the origin such that if  $y_0 \in \Gamma_0$ , the solution  $y(t) = 0$ ,  $t \geq 0$ , of the controlled system (1) is locally asymptotically stable, and  $J_{\min} = y_0^T P(0) y_0$ .

Finally, if  $\Gamma = \mathfrak{R}^n$  then the solution  $y(t) = 0$ ,  $t \geq 0$ , of the controlled system (1) is globally asymptotically stable. *Proof.* Let us consider the linear feedback control (3) with matrix  $P$  determined by equation (6) which transfers the nonlinear system (1) from an initial to a final state (4), minimizing the functional (5) where the function  $l(y, u)$  need to be determined.

According to the Dynamic Programming rules [6] one knows that if the minimum of functional (5) exists and if  $V$  is a smooth function of the initial conditions, then it satisfies the following Hamilton – Jacobi – Bellman equation:

$$\min_u \left( \frac{dV}{dt} + y^T \tilde{Q} y + u^T R u \right) = 0 \quad (8)$$

Considering a Lyapunov function

$$V = y^T P(t) y, \quad (9)$$

where  $P(t)$  is a symmetric positive definite matrix and it satisfies the differential Riccati equation (6).

Note that the derivative of the function  $V$ , evaluated in the optimal trajectory with control given by (3) is

$$\begin{aligned} \dot{V} &= \dot{y}^T P y + y^T \dot{P} y + y^T P \dot{y} = [y^T A^T + \\ &+ h^T(y, u) - y^T P B (R^{-1})^T B^T] P y + y^T \dot{P} y + \\ &+ y^T P [A y + h(y, u) - B R^{-1} B^T P y] \end{aligned} \quad (10)$$

Then substituting  $\dot{V}$  in the Hamilton – Jacobi – Bellman equation (8) one obtains,

$$\begin{aligned} y^T [\dot{P} + A^T P + PA - PBR^{-1}B^T P] y + h^T(y, u) P y + \\ + y^T P h(y, u) + l(y, u) = 0 \end{aligned} \quad (11)$$

Then

$$l(y, u) = y^T \tilde{Q} y - h^T(y, u) P y - y^T P h(y, u) \quad (12)$$

Note that for positive definite function  $l(y, u)$  and positive definite matrix  $R$ , the derivative of the function (9), evaluated in optimal trajectories of system (1), is given by

$$\dot{V} = -l(y, u) - u^T R u, \quad (13)$$

and, it is negative definite. Then, the function (9) is Lyapunov function, and the controlled system (1) is locally asymptotically stable. Integrating the derivative of the Lyapunov function (9) given by

$$\dot{V} = -y^T \tilde{Q} y - u^T R u$$

along the optimal trajectory, we obtain  $J_{\min} = y_0^T P(0) y_0$ .

Finally, if  $\Gamma = \mathfrak{R}^n$ , global asymptotic stability follows as a direct consequence of the radial unboundedness condition for the Lyapunov function (9)  $V(y) \rightarrow \infty$  as  $\|y\| \rightarrow \infty$ .

We remark that according to the optimal control theory of linear systems with quadratic functional [3] the solution of the nonlinear differential Riccati equation (6) is positive definite and symmetric matrix for all  $t \in [0, t_f]$  and positive definite matrix  $R$  and positive semidefinite matrix  $Q$  given, one can conclude the Theorem proof. When the function  $h(y, u) = h(y)$ , i.e. it is not depend on  $u$ , the Theorem 1 is equivalent to the Theorem related in [15].

### 3. A MODEL OF WHEELED MOBILE ROBOT

Consider the wheeled mobile robot of unicycle type, shown in Figure 1. The vehicle is equipped with two identical, parallel, and nondeformable rear wheels which are controlled independently by motors, and a front free wheel. It is assumed that the plane of each wheel is perpendicular to the ground and that the contact between the wheels and ground is pure rolling and nonslipping, i.e., the velocity of the center of mass of the robot  $v$  is orthogonal to the rear wheel axis. It is also assumed that the masses and inertias of the wheels are negligible and that the center of mass of the mobile robot is located in the middle of the axis connecting the rear wheels.

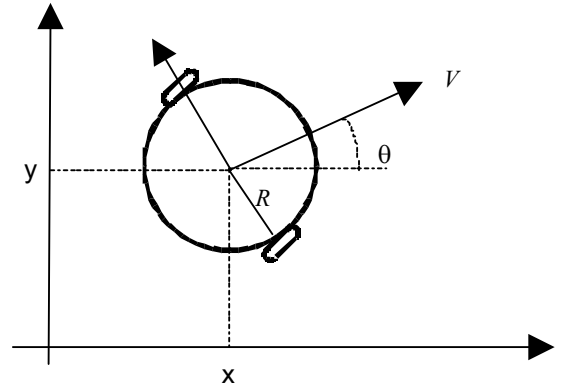


Fig 1.- A wheeled mobile robot of unicycle type

The kinematics of the mobile robot is modeled by the equations

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega\end{aligned}\quad (14)$$

where  $x$  and  $y$  denote the position of the wheel axis center, and  $\theta$  is the robot orientation with respect to the x-axis. The symbols  $v$  and  $\omega$  are the linear and angular mobile robot velocities, respectively. This model is so-called first-order kinematics model of the wheeled mobile robot, which was used by many researcher (see, for example, papers [4], [7], [8], [11], [14]). A detailed analytical study of the kinematics of wheeled mobile robots is found in [2].

Consider transformation

$$z = T \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (15)$$

where  $T \in \mathbb{R}^{3 \times 3}$  is the transformation matrix defined as:

$$T = \begin{bmatrix} 0 & 0 & 1 \\ \cos \theta & \sin \theta & 0 \\ -\theta \cos \theta + 2 \sin \theta & -\theta \sin \theta - 2 \cos \theta & 0 \end{bmatrix} \quad (16)$$

We relate, via transformation (15), (16), the system (14) to a new one expressed in coordinates  $z(t) = [z_1(t) \ z_2(t) \ z_3(t)]^T \in \mathbb{R}^3$  and which is more comfortable for a posterior control design. After transformations we have a system in so called chained form:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ z_2 U_1 - z_1 U_2 \end{bmatrix} \quad (17)$$

where  $U(t) = [U_1(t) \ U_2(t)]^T \in \mathbb{R}^2$  is defined as:

$$\begin{aligned}U_1 &= \omega \\ U_2 &= v - \omega(\tilde{x} \sin \theta - \tilde{y} \cos \theta)\end{aligned}\quad (18)$$

The model (17) is famous Brockett integrator [5] that is more comfortable to subsequent control design.

#### 4. CONTROL PROBLEM STATEMENT

In this section, we consider an optimal control problem of the system (17) in order to reach the desired trajectory:

$$\tilde{z} = \begin{bmatrix} \tilde{z}_1(t) \\ \tilde{z}_2(t) \\ \tilde{z}_3(t) \end{bmatrix} \quad (19)$$

The control vector  $\tilde{u}$  which maintains the system (17) at the desired steady state (19) satisfies the following equation:

$$\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \dot{\tilde{z}}_3 \end{bmatrix} = \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ z_2 \tilde{u}_1 - z_1 \tilde{u}_2 \end{bmatrix} \quad (20)$$

Defining error signal  $e(t) \in \mathbb{R}^3$  as a difference between transformed system  $z(t) \in \mathbb{R}^3$  and a desired trajectory  $\tilde{z}(t) \in \mathbb{R}^3$  we have:

$$e = z - \tilde{z} \quad (21)$$

and

$$u = U - \tilde{u} \quad (22)$$

Subtracting (20) from (17) and admitting (21) and (22), we get the error system in following form:

$$\dot{e} = A(t)e + h(e, u) + B(t)u \quad (23)$$

where:

$$\begin{aligned}A(t) &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tilde{u}_2 & -\tilde{u}_1 & 0 \end{bmatrix}; \quad h(e, u) = \begin{bmatrix} 0 \\ 0 \\ e_1 u_2 - e_2 u_1 \end{bmatrix}; \\ B(t) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\tilde{z}_2 & \tilde{z}_1 \end{bmatrix} u\end{aligned} \quad (24)$$

and  $\dot{e} \in \mathbb{R}^3$  is a time derivative of  $e \in \mathbb{R}^3$  and  $u \in \mathbb{R}^2$  is control vector. Then we can formulate an optimal control problem as:

Find the optimal control

$$u = -R^{-1} B(t)^T P(t) e \quad (25)$$

that leads the system (23) from initial to final state  $e(t_f) = 0$  minimizing functional

$$J = \int_0^{t_f} [\ell(e, u) + u^T R u] dt \quad (26)$$

where  $P(t)$  the symmetric positive definite matrix is the solution of the matrix differential Riccati equation:

$$\dot{P} + PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (27)$$

This way, applying the Theorem (if the condition (2) hold) and finding a solution to (27) we obtain control vector (25).

## 5. NUMERICAL SIMULATIONS

Numerical simulations were performed using Runge-Kutta fourth order integration method with variable step. We set the desired trajectory of the system (12) as follows:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{bmatrix} = \begin{bmatrix} t \\ t \\ \frac{\pi}{4} \end{bmatrix} \quad (28)$$

The correspondent desired trajectory of the system (17) is

$$\tilde{z}(t) = \begin{bmatrix} \frac{\pi}{4} \\ \frac{\sqrt{2}t}{4} \\ \frac{\pi\sqrt{2}}{4}t \end{bmatrix} \quad (29)$$

It is clear that

$$\tilde{u} = \begin{bmatrix} 0 \\ \sqrt{2} \end{bmatrix} \quad (30)$$

Matrix  $Q$  was chosen as:

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \text{ and } R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

First we have found the numerical solution for differential Riccati equation. The time-varying Riccati coefficients are present in Fig. 2. Then applying control (25) we obtain error trajectory (Fig. 3) for initial conditions  $\theta_0 = -\pi/4, x_0 = 1, y_0 = 0$ .

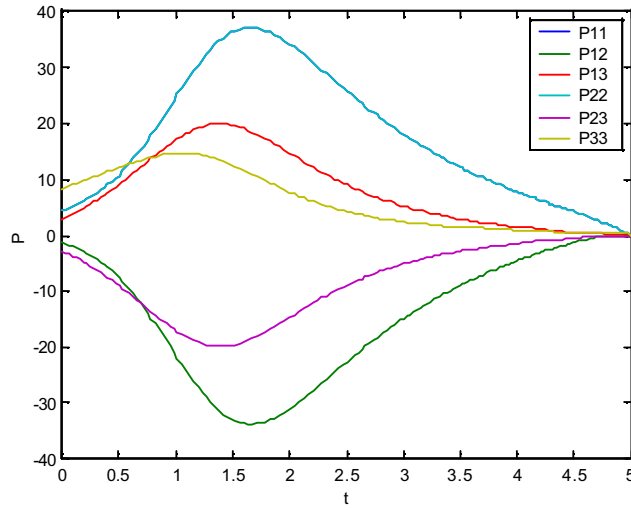


Fig. 2. Time-varying Riccati coefficients

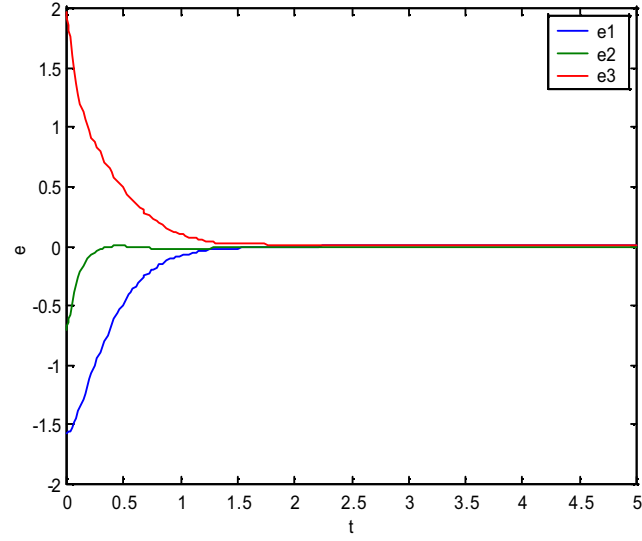


Fig. 3. Time traces of the controlled error system

and admitting the following transformations

As we see in Fig.3, the error converges to zero after 2 units of time.

Integrating the system (17) controlled by control

$$U = u + \tilde{u} \quad (31)$$

$$\begin{aligned} \theta &= z_1 \\ x &= z_2 \cos \theta + z_3 \sin \theta \\ y &= z_2 \sin \theta - z_3 \cos \theta \end{aligned} \quad (32)$$

we obtain the time trajectories of the original controlled system shown in Fig.4.

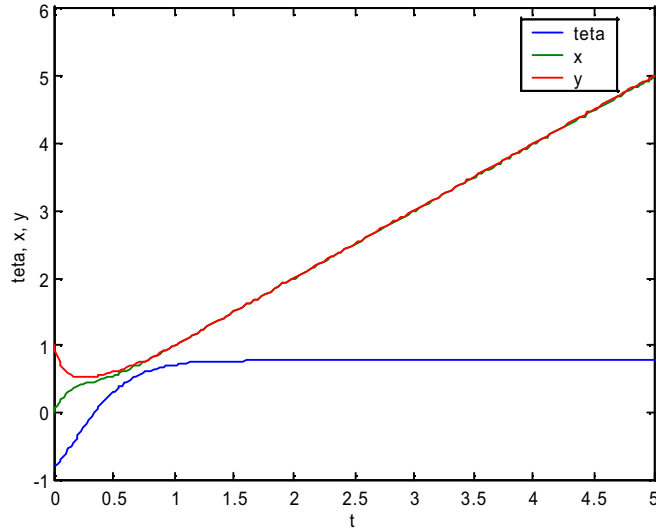


Fig. 4. Time trajectories of the wheeled mobile robot

In the figure above are represented in blue the orientation of the vehicle, in red the  $x$  coordinate and in green the  $y$  coordinate. As it's clear from this figure that  $x$  and  $y$

coordinate converge to a desired trajectory and  $\theta$  orientation of the robot remains constant.

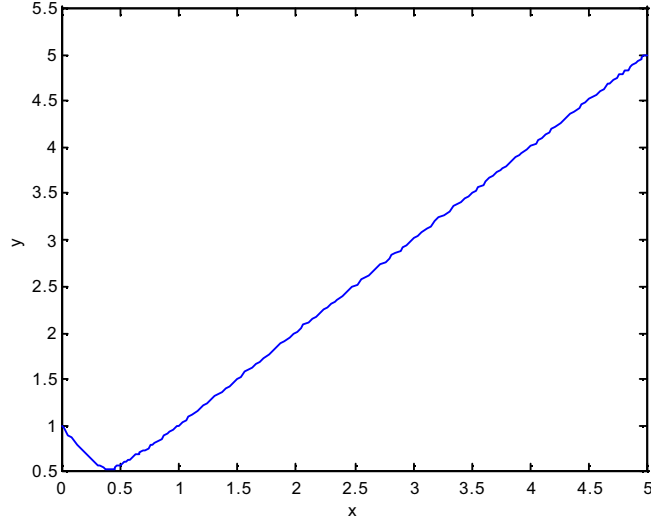


Fig. 5. Phase portrait of the wheeled mobile robot

## 7. CONCLUSIONS

In this paper we considered a nonholonomic wheeled mobile robot from the optimal linear feedback control approach. By transformation (15) the wheeled mobile robot control problem was reduced to a well-known Brockett problem [5] for which no stabilizing static state-

feedbacks exist. In [11] it is remarked that time varying controllers proposed in [16] have typically slow rates of convergence. We applied the optimal linear time-varying feedback control for nonlinear system proposed in [15]. Numerical simulations of this control applied to wheeled mobile robot show rapid rates of convergence to desired motion. One of these simulations of a difficult case  $\theta_o = \pi, x_o = 0, y_o = 1$  is shown in Fig. 6.

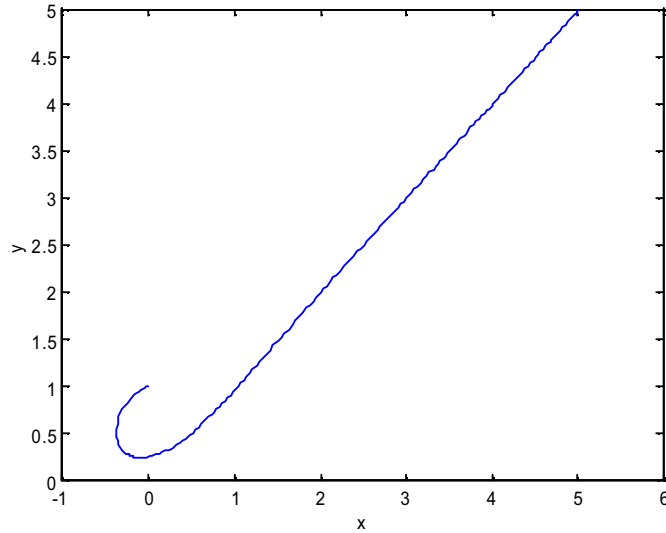


Fig. 6. Phase portrait of the wheeled mobile robot for initial conditions  $\theta_o = \pi, x_o = 0, y_o = 1$

## ACKNOWLEDGEMENTS

The authors thank CNPQ for the support on this research.

## REFERENCES

- [1] Aguiar A.P. and A. Pascoal. Stabilization of the extended nonholonomic double integrator via logic-based hybrid control: An application to point stabilization of mobile robots. In: SYROCO'00-6<sup>th</sup> International IFAC Symposium on Robot control. Vienna, Austria, 2000.
- [2] Alexander J.C. and Maddocks J.H., On the kinematics of wheeled mobile robots, *Int. J. Robot. Res.*, 8 (5) 15-27, 1989.
- [3] Anderson B.D.O. & J.B. Moor, *Optimal Control : Linear Quadratic Methods*, Prentice-Hall, NY, 1990.
- [4] Astolfi A., Exponential stabilization of wheeled mobile robot via discontinuous control. *Journal of Dynamic systems, Measurements and Control*, 121, 121-126, 1999.
- [5] Brockett, R. Asymptotic stability and feedback stabilization. In: R. Brockett, R. Millman & H. Sussmann, *Differential Geometric Control Theory* (pp.181-191), 1983.
- [6] Bryson A.E. and Y. Ho. *Applied Optimal Control*. Hemisphere Publ.corp., Washington D.C., 1975.
- [7] Canudas de Wit C., Sördalen, Exponential Stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on automatic Control* 37 (11), 1791-1797, 1992.
- [8] Dixon WE, Jiang ZP, Dawson DM, Global exponential setpoint control of wheeled mobile robots: a Lyapunov approach, *Automatica* 36 (2000) 1741-1746.
- [9] Godhavn J. M. and Egeland. A Lyapunov approach to exponential stabilization of nonholonomic systems in power form. *IEEE Transactions on Automatic Control* 42(7), 1028-1032, 1997.
- [10] Hespanha J. P. Stabilization of nonholonomic integrators via logic-based switching. In: *Proc. 13<sup>th</sup> World congress of IFAC*. Vol. E. S. Francisco, CA, pp.467-472, 1996.
- [11] Kim BM & Tsiotras P, Controllers for unicycle-type wheeled robots: theoretical results and experimental validation, *IEEE Trans. on Robotics and automation*, vol.18, no 3, 2002.
- [12] Kolmanovsky I. & N.H. McClamroch, Developments in nonholonomic control problems. *IEEE Control Systems Magazine*, 1995, 15, 20-36
- [13] P. Morin and C. Samson, *Application of Backstepping Techniques to the Time-Varying Exponential Stabilization of Chained Form Systems*, INRIA Research Report No. 2792, Sophia-Antipolis, 1996.
- [14] Oriolo G., Luca A. de, WMR Control via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation, *IEEE Transaction on Control System Technology*, 10 (6) 2002.
- [15] Rafikov M, Balthazar JM, On control and synchronization in chaotic and hyperchaotic systems via linear feedback control, *Communications in Nonlinear Science and Numerical Simulation* (2007), doi:10.1016/j.cnsns.2006.12.011
- [16] Samson C., Control of Chained systems: Application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control* 40(1), 64-77, 1995