



Universidade Federal
do Rio de Janeiro

Escola Politécnica

RESULTADOS EXPERIMENTAIS EM CONTROLE DE FORMAÇÃO
BASEADO EM FUNÇÕES POTENCIAIS

Jonathan Fried

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro em Engenharia de Controle e Automação.

Orientador: Fernando Cesar Lizarralde

Rio de Janeiro
Março de 2015

RESULTADOS EXPERIMENTAIS EM CONTROLE DE FORMAÇÃO
BASEADO EM FUNÇÕES POTENCIAIS

Jonathan Fried

PROJETO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO.

Examinado por:

Prof. Fernando Cesar Lizarralde, D.Sc.

Prof. Alessandro Jacoud Peixoto, D.Sc.

Prof. Josiel Alves Gouvêa, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2015

Fried, Jonathan

Resultados experimentais em controle de formação baseado em funções potenciais/Jonathan Fried. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2015.

XV, 121 p.: il.; 29, 7cm.

Orientador: Fernando Cesar Lizarralde

Projeto (graduação) – UFRJ/Escola Politécnica/Curso de Engenharia de Controle e Automação, 2015.

Referências Bibliográficas: p. 116 – 121.

1. Controle de Formação.
 2. Função Potencial.
 3. Robôs não holonômicos.
 4. Servovisão.
- I.
Lizarralde, Fernando Cesar. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Controle e Automação. III. Título.

*Aos meus pais que me apoiaram e
me deram tudo que necessitei para
dar o primeiro passo.*

Agradecimentos

Agradeço aos meus pais, Charles e Rosangela, que me apoiaram por toda a graduação e sem os quais não teria sido possível chegar aqui. Ao meu orientador, Fernando Cesar Lizarralde que foi paciente ao me orientar durante todo este projeto, tantas horas que foram gastas. Ao professor Afonso que tantos problemas me resolveu durante esses anos. Ao Fernando, ao Maurício, ao Matheus e a todos do laboratório, tanto pelo auxílio técnico quanto por tornar tão agradável o ambiente de trabalho. Obrigado.

Resumo do Projeto apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro (de Controle e Automação)

RESULTADOS EXPERIMENTAIS EM CONTROLE DE FORMAÇÃO BASEADO EM FUNÇÕES POTENCIAIS

Jonathan Fried

Março /2015

Orientador: Fernando Cesar Lizarralde

Curso: Engenharia de Controle e Automação

Neste trabalho, são apresentados resultados experimentais em o controle de formação de um sistema composto por um grupo de robôs não-holonômicos. Controle de formação tem uma vasta gama de aplicações e é um tópico muito pesquisado pela sociedade acadêmica nos últimos anos. Este trabalho contribui com uma visão geral sobre controle de formação, antes de apresentar resultados, tanto simulados como experimentais, para um caso específico que é o de um esquema de controle de formação por função potencial artificial saturada e deslocada, que evita colisões entre robôs.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer (de Controle e Automação)

EXPERIMENTAL RESULTS IN FORMATION CONTROL BASED ON
POTENTIAL FUNCTIONS

Jonathan Fried

March/2015

Advisor: Fernando Cesar Lizarralde

Course: Control and Automation Engineering

In this project, experimental results in formation control of a system composed of a group of nonholonomic robots are presented. Formation control has a broad range of applications and is an active research topic in the academic community. This work contributes with an overview about formation control, before presenting results, both simulated and experimental, for a specific case of formation control which is formation control based on an artificial saturated potential function, avoiding collisions between robots.

Sumário

Lista de Figuras	x
Lista de Tabelas	xiv
1 Introdução	1
1.1 Métodos de Controle de Formação	2
1.1.1 Abordagem comportamental	2
1.1.2 Abordagem de algoritmo de consenso	3
1.1.3 Abordagem de estrutura virtual	3
1.1.4 Abordagem de líder-seguidor	4
1.1.5 Abordagem de modelo preditivo	5
1.2 Objetivo	5
1.3 Organização do trabalho	6
2 Descrição do sistema robótico	7
2.1 Manipuladores	8
2.1.1 Cinemática Direta	8
2.1.2 Cinemática Diferencial	11
2.2 Robôs Não-holonômicos	12
2.2.1 Restrição Não-Holonômica	13
2.3 Servo Visão	14
2.4 Controle de formação por função potencial saturada e deslocada . . .	15
3 Modelagem e Simulação	19
3.1 Modelagem	20
3.1.1 Roomba	20

3.1.2	Manipuladores	21
3.1.3	Bloco de Controle	24
3.2	Simulações	26
3.2.1	Um agente buscando a formação	27
3.2.2	Dois agentes buscando a formação	43
3.2.3	Três agentes buscando a formação	58
3.3	Conclusão sobre as simulações	66
4	Resultados Experimentais	74
4.1	Equipamentos	74
4.1.1	Logitech QuickCam Communicate Deluxe	75
4.1.2	iRobot Roomba 530 series	77
4.1.3	Motoman Dia10	79
4.1.4	Raspberry Pi Model B	81
4.1.5	Matlab 2013b	81
4.2	Resultados	84
4.2.1	Um agente buscando a formação	85
4.2.2	Dois agentes buscando a formação	92
4.2.3	Três agentes buscando a formação	100
5	Conclusões	114
5.1	Trabalhos Futuros	115

Lista de Figuras

2.1	Motoman DIA10, braços posicionados em configuração planar.	9
2.2	Diagrama do Motoman apenas com as juntas a serem utilizadas.	9
2.3	Diagrama do braço direito, com rotação e posição do efetuator.	10
2.4	Diagrama do braço esquerdo, com rotação e posição do efetuator.	10
2.5	Diagrama de um robô não-holonômico do tipo unicycle.	13
2.6	Diagrama da configuração entre a câmera e o espaço de trabalho	14
2.7	Função potencial artificial saturada e deslocada	16
2.8	Representação das regiões de colisão e vizinhança de um agente i	17
3.1	Modelo geral, com as três plantas e o bloco de controle.	19
3.2	Modelo do Roomba	21
3.3	Modelo do Manipulador Esquerdo	22
3.4	Modelo do Manipulador Direito	23
3.5	Modelo do Bloco de Controle	25
3.6	Trajatória dos agentes - Roomba Caso 1	29
3.7	Orientação dos agentes - Roomba Caso 1	30
3.8	Distância entre agentes - Roomba Caso 1	31
3.9	Trajatória dos agentes - Roomba Caso 2	32
3.10	Orientação dos agentes - Roomba Caso 2	33
3.11	Distância entre agentes - Roomba Caso 2	34
3.12	Trajatória dos agentes - M. Esq. Caso 1	35
3.13	Orientação dos agentes - M. Esq. Caso 1	36
3.14	Distância entre agentes - M. Esq. Caso 1	37
3.15	Trajatória dos agentes - M. Esq. Caso 2	38
3.16	Orientação dos agentes - M. Esq. Caso 2	39

3.17	Distância entre agentes - M. Esq. Caso 2	40
3.18	Trajectoria dos agentes - M. Dir. Caso 1	41
3.19	Orientação dos agentes - M. Dir. Caso 1	42
3.20	Distância entre agentes - M. Dir. Caso 1	43
3.21	Trajectoria dos agentes - M. Dir. Caso 2	44
3.22	Orientação dos agentes - M. Dir. Caso 2	45
3.23	Distância entre agentes - M. Dir. Caso 2	46
3.24	Trajectoria dos agentes - M.Esq./M.Dir. Caso 1	47
3.25	Orientação dos agentes - M. Esq./Dir. Caso 1	48
3.26	Distância entre agentes - M.Esq./M.Dir. Caso 1	49
3.27	Trajectoria dos agentes - M.Esq./M.Dir. Caso 2	50
3.28	Orientação dos agentes - M. Esq./Dir. Caso 2	51
3.29	Distância entre agentes - M.Esq./M.Dir. Caso 2	52
3.30	Trajectoria dos agentes - M.Esq./Rob. Caso 1	53
3.31	Orientação dos agentes - M. Esq./Roomba Caso 1	54
3.32	Distância entre agentes - M.Esq./Rob. Caso 1	55
3.33	Trajectoria dos agentes, com seu $\theta(0)$ - M.Esq./Rob. Caso 2	56
3.34	Orientação dos agentes - M. Esq./Roomba Caso 2	57
3.35	Distância entre agentes - M.Esq./Rob. Caso 2	58
3.36	Trajectoria dos agentes - M.Dir./Rob. Caso 1	59
3.37	Orientação dos agentes - M. Dir./Roomba Caso 1	60
3.38	Distância entre agentes - M.Dir./Rob. Caso 1	61
3.39	Trajectoria dos agentes - M.Dir./Rob. Caso 2	62
3.40	Orientação dos agentes - M. Dir./Roomba Caso 2	63
3.41	Distância entre agentes - M.Dir./Rob. Caso 2	64
3.42	Trajectoria dos agentes - M.Esq./M.Dir./Rob. Caso 1	65
3.43	Orientação dos agentes - M. Esq./Dir./Roomba Caso 1	66
3.44	Distância entre agentes - M.Esq./M.Dir./Rob. Caso 1	67
3.45	Trajectoria dos agentes - M.Esq./M.Dir./Rob. Caso 2	68
3.46	Orientação dos agentes - M. Esq./Dir./Roomba Caso 2	69
3.47	Distância entre agentes - M.Esq./M.Dir./Rob. Caso 2	70
3.48	Trajectoria dos agentes - M.Esq./M.Dir./Rob. Caso 3	71

3.49	Orientação dos agentes - M. Esq./Dir./Roomba Caso 3	72
3.50	Distância entre agentes - M.Esq./M.Dir./Rob. Caso 3	73
4.1	Diagrama ilustrando disposição e comunicação dos equipamentos . . .	75
4.2	Logitech QuickCam Communicate Deluxe	75
4.3	Roomba com haste para elevar o padrão-alvo	76
4.4	Modelo da projeção do padrão-alvo e do roomba na imagem	77
4.5	iRobot Roomba 530 Series	78
4.6	Juntas do motoman, com nome e especificações.	79
4.7	Motoman Dia10 NX100	79
4.8	Limites das juntas do braço do motoman, em configuração planar. . .	81
4.9	Padrões-Alvo, da esquerda para a direita, M. Esq., M. Dir. e Roomba	83
4.10	Posição e orientação inicial dos agentes - Roomba	86
4.11	Trajetória dos agentes - Roomba	87
4.12	Distância entre agentes - Roomba	88
4.13	Posição e orientação final dos agentes - Roomba	89
4.14	Orientação dos agentes - Roomba	90
4.15	Posição e orientação inicial dos agentes - M. Esq.	91
4.16	Trajetória dos agentes - M. Esq.	92
4.17	Distância entre agentes - M. Esq.	93
4.18	Posição e orientação final dos agentes - M. Esq.	94
4.19	Orientação dos agentes - M. Esq.	95
4.20	Posição e orientação inicial dos agentes - M. Dir.	96
4.21	Trajetória dos agentes - M. Dir.	97
4.22	Distância entre agentes - M. Dir.	98
4.23	Posição e orientação final dos agentes - M. Dir.	99
4.24	Orientação dos agentes - M. Dir.	100
4.25	Posição e orientação inicial dos agentes - M. Esq./Dir.	101
4.26	Trajetória dos agentes - M. Esq./Dir.	102
4.27	Distância entre agentes - M. Esq./Dir.	103
4.28	Posição e orientação final dos agentes - M. Esq./Dir.	103
4.29	Orientação dos agentes - M. Esq./Dir.	104
4.30	Posição e orientação inicial dos agentes - M. Esq./Roomba	104

4.31	Trajetória dos agentes - M. Esq./Roomba	105
4.32	Distância entre agentes - M. Esq./Roomba	106
4.33	Posição e orientação final dos agentes - M. Esq./Roomba	106
4.34	Orientação dos agentes - M. Esq./Roomba	107
4.35	Posição e orientação inicial dos agentes - M. Dir./Roomba	107
4.36	Trajetória dos agentes - M. Dir./Roomba	108
4.37	Distância entre agentes - M. Dir./Roomba	109
4.38	Posição e orientação final dos agentes - M. Dir./Roomba	109
4.39	Orientação dos agentes - M. Dir./Roomba	110
4.40	Posição e orientação inicial dos agentes - M. Esq./M. Dir./Roomba . .	110
4.41	Trajetória dos agentes - M. Esq./M. Dir./Roomba	111
4.42	Distância entre agentes - M. Esq./M. Dir./Roomba	112
4.43	Posição e orientação final dos agentes - M. Esq./M. Dir./Roomba . .	113
4.44	Orientação dos agentes - M. Esq./M. Dir./Roomba	113

Lista de Tabelas

3.1	Parâmetros Iniciais da Simulação	27
3.2	Parâmetros para controle de um agente - Roomba Caso 1	28
3.3	Parâmetros para controle de um agente - Roomba Caso 2	31
3.4	Parâmetros para controle de um agente - M. Esq. Caso 1	35
3.5	Parâmetros para controle de um agente - M. Esq. Caso 2	37
3.6	Parâmetros para controle de um agente - M. Dir. Caso 1	40
3.7	Parâmetros para controle de um agente - M. Dir. Caso 2	42
3.8	Parâmetros para controle de 2 agentes - M.Esq./M.Dir. Caso 1	46
3.9	Parâmetros para controle de 2 agentes - M.Esq./M.Dir. Caso 2	48
3.10	Parâmetros para controle de 2 agentes - M.Esq./Rob. Caso 1	50
3.11	Parâmetros para controle de 2 agentes - M.Esq./Rob. Caso 2	53
3.12	Parâmetros para controle de 2 agentes - M.Dir./Rob. Caso 1	55
3.13	Parâmetros para controle de 2 agentes - M.Dir./Rob. Caso 2	58
3.14	Parâmetros iniciais para controle de 3 agentes - Caso 1	61
3.15	Parâmetros iniciais para controle de 3 agentes - Caso 2	63
3.16	Parâmetros iniciais para controle de 3 agentes - Caso 3	66
4.1	Parâmetros comuns a todos os experimentos	85
4.2	Posição e orientação inicial dos agentes - Roomba	86
4.3	Posição e orientação final dos agentes - Roomba	87
4.4	Posição e orientação inicial dos agentes - M. Esq.	88
4.5	Posição e orientação final dos agentes - M. Esq.	90
4.6	Posição e orientação inicial dos agentes - M. Dir.	91
4.7	Posição e orientação final dos agentes - M. Dir.	92
4.8	Posição e orientação inicial dos agentes - M. Esq./Dir.	94

4.9	Posição e orientação final dos agentes - M. Esq./Dir.	95
4.10	Posição e orientação inicial dos agentes - M. Esq./Roomba	96
4.11	Posição e orientação final dos agentes - M. Esq./Roomba	97
4.12	Posição e orientação inicial dos agentes - M. Dir./Roomba	98
4.13	Posição e orientação final dos agentes - M. Dir./Roomba	99
4.14	Posição e orientação inicial dos agentes - M. Esq./M. Dir./Roomba .	101
4.15	Posição e orientação final dos agentes - M. Esq./M. Dir./Roomba . .	112

Capítulo 1

Introdução

Controle cooperativo é questão de suma importância para a coordenação de um grupo de robôs autônomos ou não-tripulados. Um grupo de robôs é capaz de realizar tarefas mais complexas, muitas vezes com custo em escala reduzido. O sistema pode ser tornado mais robusto e eficiente enquanto ganha redundância, eficiência e flexibilidade na sua estrutura. E no caso de tarefas que envolvem a movimentação de um grupo de múltiplos robôs, é, em geral, necessário seguir uma trajetória pré-definida, enquanto se mantém uma formação espacial desejado no espaço de trabalho.

Pode-se ver aplicações disso em diversos ramos da tecnologia, dos quais se citam o controle de formação de veículos espaciais ((SCHARF *et al.*, 2004), (REN e BEARD, 2003) e (CAI e HUANG, 2013)), controle de satélites ((SERRANI, 2003) e (MARTIN *et al.*, 2001)), grupos de veículos sub-aquáticos (STILWELL e BISHOP, 2000), grupos de veículos aéreos não tripulados ((QUINTERO *et al.*, 2013) e (ZHANG e LIU, 2013)), automatização de rodovias ((BENDER, 1991) e (WANG e ZENGJIN, 1998)), mapeamento de regiões (HOWARD *et al.*, 2006) e realização de tarefas em ambientes hostis (LI *et al.*, 2002).

Segundo (REYNOLDS, 1987), o controle de formação consiste em fazer um grupo de agentes convergir para uma formação espacial segundo certos critérios:

- **Separação:** Devem ser evitadas colisões entre os agentes da formação.
- **Alinhamento:** Todos os agentes da formação devem ter a mesma orientação.
- **Coesão:** A formação deve ser mantida mesmo em movimento.

Seguindo esses critérios, diversos trabalhos foram desenvolvidos pela comunidade acadêmica, sobre controle de formação (DAI e LEE, 2012), (NECOARA *et al.*, 2013), (LEE *et al.*, 2013), (CAO *et al.*, 2002), (REN, 2007), (GOUVÊA, 2011), entre outros. As idéias básicas, porém, serão apresentadas na seção a seguir.

1.1 Métodos de Controle de Formação

A seguir, alguns métodos de controle de formação, com citações de trabalhos que os definem ou utilizam.

1.1.1 Abordagem comportamental

Para este controle, cada agente possui uma série de comportamentos desejados onde cada um é representado por uma ação de controle. Então, a ação de controle de cada agente vai ser definida como a média ponderada das ações de resposta aos seus comportamentos (BALCH e ARKIN, 1998). Comportamentos podem incluir evitar colisões, evitar obstáculos, atingir um objetivo e manter uma formação. A mais básica das diferenças entre as diversas abordagens deste tipo de controle na literatura é em como elaborar e ponderar estes comportamentos.

Em (CAO *et al.*, 2002), é utilizado um algoritmo genético para escolher os pesos do controle de forma que seja sempre escolhido o padrão apropriado para manter a formação e evitar obstáculos. Em (YUN *et al.*, 1997) os comportamentos fazem o grupo de agentes convergir para uma linha ou círculo. Já em (MONTEIRO e BICHO, 2002) o controle comportamental é modelado em um sistema de dinâmica não-linear, para gerar trajetória e evitar obstáculos.

Existem ainda muitos outros estudos na literatura sobre essa abordagem, dos quais podemos citar o controle comportamental baseado em espaço nulo em (ANTONELLI *et al.*, 2007) e o controle comportamental por campos potenciais sociais, em (REIF e WANG, 1999). A grande vantagem desta abordagem comportamental, como um todo, é a facilidade de tomada de decisões frente à objetivos conflitantes, visto que basta alterar os parâmetros de ponderação dos comportamentos para dar prioridade a um deles em relação aos demais. Da mesma forma,

como cada agente é responsável pela elaboração do controle do mesmo, o controle de formação por comportamento é descentralizado. A desvantagem é a dificuldade de analisar matematicamente a estabilidade da formação (GOUVÊA, 2011).

1.1.2 Abordagem de algoritmo de consenso

O controle por consenso requer a capacidade de um grupo de agentes de chegar num acordo, baseado na troca de informações dentro do grupo. Para este fim, uma lei de controle baseada nessa troca de informações, dirige cada um dos agentes a uma posição que está de acordo com a decisão anteriormente tomada pelo grupo ((OLFATI-SABER *et al.*, 2007) e (REN *et al.*, 2005)). A idéia é cada agente trocar informações apenas com seus vizinhos locais e ainda assim todos os agentes chegarem a um valor comum. Esses algoritmos em geral tomam a forma de uma dinâmica de primeira ordem, mas extensões para dinâmicas de segunda ordem sob fluxo de informações não-direcionados já foram apresentadas em (OLFATI-SABER e MURRAY, 2003).

Podem ser vistos outros estudos sobre esse controle em (OKOLOKO, 2013), onde o movimento dos agentes está restrito à superfície de uma esfera. Em (ZHOU e LIN, 2013), existe o estudo sistemas lineares de alta ordem com atrasos na rede de comunicação e na entrada do sistema. Já em (CAO *et al.*, 2013) é visto como é possível abordar o algoritmo de consenso para um time de agentes através de interações estocásticas.

1.1.3 Abordagem de estrutura virtual

O controle de formação por estrutura virtual é normalmente utilizado em controle de veículos espaciais e pequenos satélites (BEARD *et al.*, 2000) e nele toda a formação é forçada a se comportar a uma rígida estruturam, esse tipo de controle apareceu pela primeira vez em (TAN e LEWIS, 1996). Nessa abordagem, o controle é determinado em em três etapas. Primeiro, é definida a dinâmica da estrutura virtual. Depois, a trajetória desejada para a estrutura é transformada na trajetória desejada para cada agente. E por último, os controles individuais são derivados para rastreamento de agentes. Em (YUAN e TANG, 2010) vê-se um exemplo direto de tal controle.

Em (ZHANG e LIU, 2013) vemos uma aplicação desse controle para o controle

de um time de veículos voadores não tripulados. Em (LEE *et al.*, 2013), esse controle é aplicado para um conjunto de veículos espaciais. (DONG *et al.*, 2013) fala sobre a teleoperação de um *cluster* de robôs móveis que devem seguir uma trajetória enquanto mantém uma formação rígida. É interessante notar que ambos os casos também fazem uso de um líder virtual, guiando a estrutura ou atuando como centróide da mesma. Como vantagem para esse tipo de controle de formação, têm-se que o movimento do grupo em formação pode ser feito de forma relativamente simples.

1.1.4 Abordagem de líder-seguidor

Um agente qualquer, para este controle, vai escolher um agente vizinho para ser seu agente líder, isto é, ele vai medir sua própria distância e orientação não a partir de uma base fixa, mas relativa a esse agente líder (DESAI *et al.*, 1998). O objetivo aqui é manter uma distância desejada e uma orientação desejada em relação ao robô líder a todo momento. E acima dessa relação de liderança local, existe um líder geral que define a trajetória para a formação, que deve se manter enquanto percorre a mesma. A vantagem dessa abordagem é que somente o comportamento do líder é suficiente para determinar o comportamento do sistema. Entretanto, ela é bastante suscetível a perturbações, uma perturbação em um líder se propaga para seu seguidor em cadeia. Uma segunda falha ocorre se a comunicação com o líder geral for perdida, visto que só ele determina o movimento da formação.

Uma aplicação de tal controle pode ser vista de em (TAKAHASHI *et al.*, 2004), um controlador que considera os limites de cada robô em suas leis de controle. Em (JAFARIAN e DE PERSIS, 2013) é estudado um caso onde a formação é desejada, com seguimento de trajetória, mas a informação que cada agente possui é mínima, os agentes sabem apenas se sua distância atual é maior ou menor do que a atual. (QUINTERO *et al.*, 2013) aplica o controle de formação líder-seguidor para um conjunto de veículos aéreos não tripulados, cujo objetivo é realizar coletivamente a tarefa de detecção. Como um último exemplo, (MAGEE *et al.*, 2013) aplica a idéia de líder seguidor para controle micro veículos aéreos de forma que eles tenham o melhor desempenho no quesito de aproveitamento de energia.

1.1.5 Abordagem de modelo preditivo

Model predictive control (MPC) é uma abordagem de controle que requer um cálculo *online* de um problema otimização para cada passo de tempo do sistema. Uma de suas vantagens é que ele pode ser aplicado em sistemas de larga escala, com um grande número estados. Em contrapartida, como ele precisa de soluções *online*, ele introduz ao sistema um atraso pelo tempo computacional necessário para resolver suas funções, a sua derivada de primeira ou segunda ordem e pelas operações matriciais (NECOARA *et al.*, 2013). Ainda assim, para sistemas onde esse atraso não é um problema, MPC ainda é um controle de grande utilidade, e portanto bem estudado pela comunidade científica, como por exemplo em (NECOARA *et al.*, 2013).

Em (YAN e BITMEAD, 2003), vê-se um exemplo em que o MPC é usado como lei de controle local para melhorar a performance do sistema, quando o sistema está sob efeito de uma comunicação imperfeita entre veículos. Já em (LADINO e PATINO, 2013) é uma otimização que permite usar o MPC em sistemas com atrasos temporais variáveis. Similarmente, (LI e SHI, 2013) apresenta uma alteração que gera a possibilidade MPC ser usado, com sistemas que tem atraso na comunicação. E como mais uma aplicação prática, MPC também pode ser usado para o cerco de múltiplos alvos por veículos aéreos não tripulados (HAFEZ *et al.*, 2013).

1.2 Objetivo

O objetivo deste trabalho é mostrar, na prática, o funcionamento do esquema de controle proposto em (GOUVÊA, 2011), um controle de formação por função potencial saturada e deslocada, que promove a convergência de um conjunto de agentes para uma formação, que minimiza esta função potencial. Para tal, é necessário primeiro definir o grupo de agentes a serem controlados e então definir a ação do controle de formação em si.

Dos equipamentos disponíveis, define-se o primeiro agente como o robô cinemático diferencial *iRobot Roomba 530* que pode ser controlado como um robô cinemático do tipo uniciclo. Os outros dois agentes, número minimamente necessário para um teste de controle de formação, serão providos pelos efetadores do robô industrial Motoman DIA10 NX100. Fazer com que estes funcionem como um robô

não-holonômico do tipo uniciclo exigirá uma restrição virtual a ser determinada. Uma *webcam Logitech QuickCam Communicate Deluxe* foi usada para a aquisição de imagens, e resta a um computador com Matlab fazer as demais tarefas, desde o processamento de imagem ao cálculo do sinal de controle.

1.3 Organização do trabalho

No capítulo 2, são revisados os conceitos teóricos necessários para o entendimento do projeto. São apresentados conceitos de cinemática direta e cinemática diferencial, referentes ao Motoman DIA10 NX100. O conceito de robôs não-holonômicos, para o roomba e as restrições de movimento dos manipuladores. O conceito de transformada de uma câmera e finalmente a definição e explanação da lei de controle por função de potencial saturada e deslocada.

No capítulo 3, é apresentado a modelagem do sistema, para ser usado em simulação, conforme os conceitos que serão vistos no capítulo 2. Na segunda parte do mesmo capítulo, serão apresentados os resultados das simulações propriamente ditas, em diversas configurações do espaço de trabalho.

O capítulo seguinte é dividido em duas partes. Na primeira, dá-se maiores detalhes do funcionamento, uso e interação dos equipamentos usados para este projeto, enquanto na segunda, apresenta-se finalmente os resultados práticos que são objetivo último deste trabalho, aproveitando-se das informações obtidas com as simulações.

No capítulo 5, o último capítulo, são feitas as conclusões finais sobre o trabalho e propostos temas para projetos futuros.

Capítulo 2

Descrição do sistema robótico

Como já informado no capítulo anterior, os esquema de controle por função potencial artificial saturada e deslocada será usado para fazer com que três agentes, os manipuladores do motoman e o roomba cujas posições e orientações serão capturados por uma câmera, convirjam para uma formação. Sendo assim é preciso entender os conceitos de cada um dos agentes da formação, da lei de controle de formação por função potencial que os governa, e da transformada da câmera que localiza os agentes no espaço de trabalho. Para os manipuladores robóticos, é fundamental saber o conceito de cinemática direta e cinemática diferencial, usados para modelar o sistema, simulá-o e resolvê-lo. Será explicado o funcionamento da cinemática de robôs não holonômicos do tipo uniciclo, que são o agente base deste trabalho. Esses conceitos mais simples porém não triviais serão todos apresentados, com o formalismo matemático adequado, ressaltando que serão considerados modelos cinemáticos para cada os agentes.

O controle de formação por função potencial saturada e deslocada, por tratar de conceitos mais complexos, será explicado em maiores detalhes. Tal controle foi apresentado em (GOUVÊA, 2011), e propõe uma lei de controle que atua na velocidade linear e angular de um agente, e os faz convergir para uma formação, determinada pelo campo potencial entre cada um desses mesmos agentes.

2.1 Manipuladores

Manipuladores robóticos são cadeias de elos e juntas. Os estados dos elos podem ser relacionados ao estados da juntas através de um modelo cinemático que relaciona posição e orientação do efetuador com ângulo das juntas.

2.1.1 Cinemática Direta

A cinemática direta de um robô é a equação que relaciona a posição e orientação de seu efetuador com as variáveis das juntas, no sistema de coordenada da base. Neste trabalho, será usado uma análise geométrica do manipuladores para inferir diretamente a equação da cinemática direta em relação aos mesmos. Sendo p a posição do efetuador, θ sua orientação e q o ângulo das juntas, a relação entre tais variáveis pode ser representada por P como mostra a equação (2.1).

$$\begin{bmatrix} p \\ \theta \end{bmatrix} = P(q) \quad (2.1)$$

Para fins experimentais, foi usado nesse trabalho o robô Motoman DIA10 (Figura 2.1) para emular dois manipuladores planares de três elos, cujos efetuadores desempenham o papel de agentes a serem controlados.

Em um diagrama da configuração planar do motoman a ser utilizada (Figura 2.2), representa-se apenas as juntas utilizadas no projeto, um total de seis juntas, com as demais mantidas em sua posição zero.

Para o manipulador direito, a junta 2 e a junta 3 rotacionam no sentido horário para incrementos nos ângulos q_{r2} e q_{r3} , devido a configurações de fábrica, relacionadas com o Denavit-Hartenberg completo do Motoman, como esquematizado na Figura 2.3. Levando em consideração tais condições e realizando uma análise geométrica, calcula-se a posição (x_{er}, y_{er}) e a orientação θ_{er} do efetuador desse manipulador conforme a matriz $P(q_r)$ (2.2).

$$P(q_r) = \begin{bmatrix} a_0 + a_1 \cos(q_{r1}) + a_2 \cos(q_{r1} - q_{r2}) + a_3 \cos(q_{r1} - q_{r2} - q_{r3}) \\ a_1 \sin(q_{r1}) + a_2 \sin(q_{r1} - q_{r2}) + a_3 \sin(q_{r1} - q_{r2} - q_{r3}) \\ q_{r1} - q_{r2} - q_{r3} \end{bmatrix} \quad (2.2)$$



Figura 2.1: Motoman DIA10, braços posicionados em configuração planar.

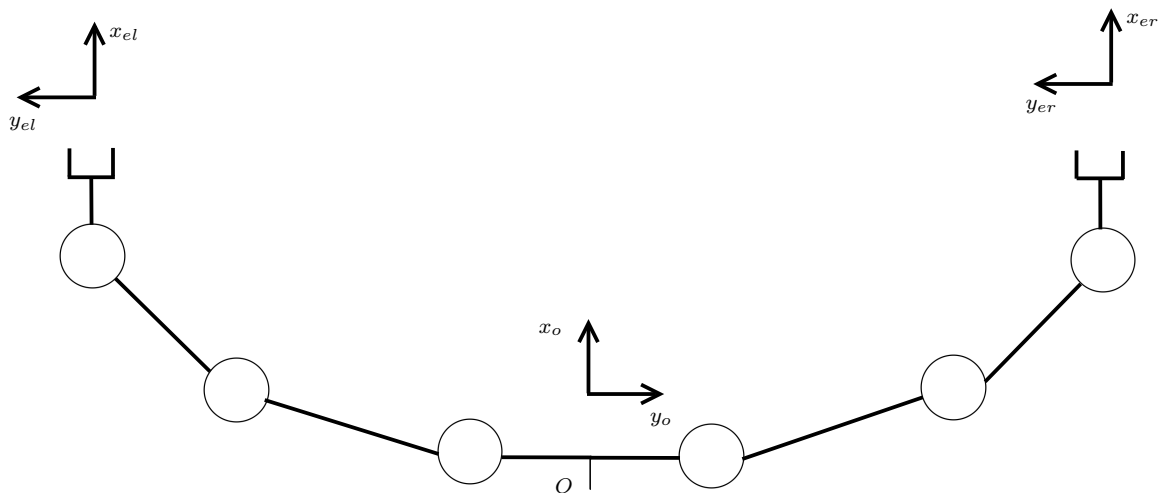


Figura 2.2: Diagrama do Motoman apenas com as juntas a serem utilizadas.

Para o manipulador esquerdo, o pensamento é análogo, mas com especificações de fábrica que determinam que a junta 1 rotaciona no sentido horário para incrementos no ângulo q_{l1} , as juntas 2 e 3 no anti-horário, como esquematizado na Figura 2.4. Com uma análise geométrica similar a anterior, calcula-se a posição (x_{el}, y_{el}) e a orientação θ_{el} do efetuador desse manipulador conforme a matriz $P(q_l)$ (2.3).

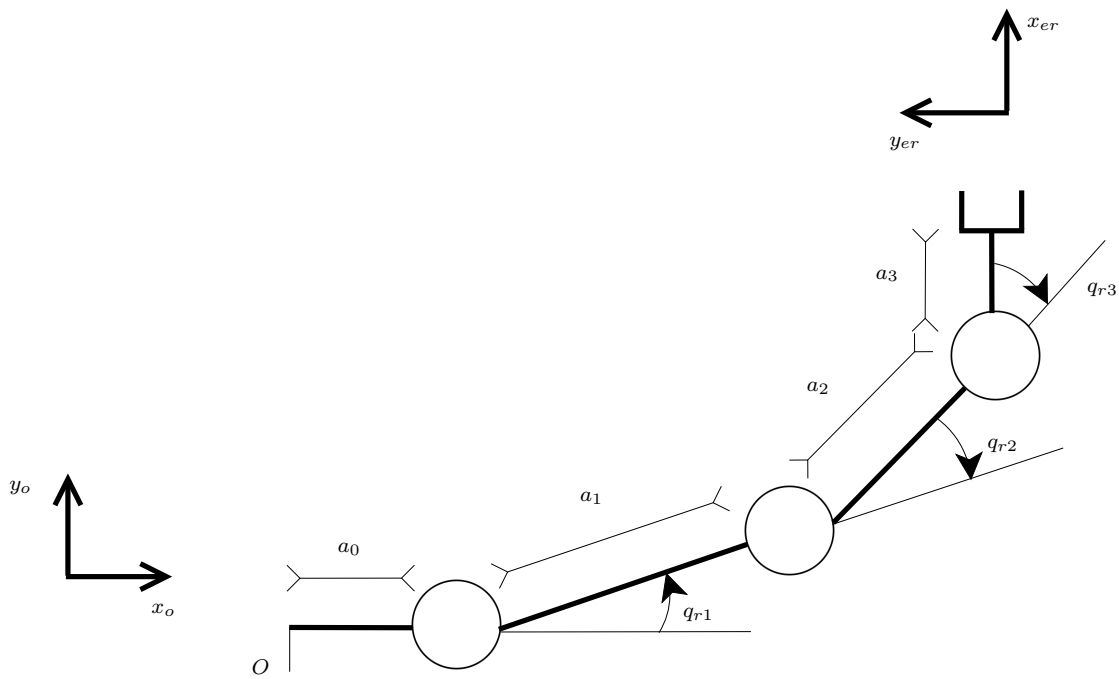


Figura 2.3: Diagrama do braço direito, com rotação e posição do efetuador.

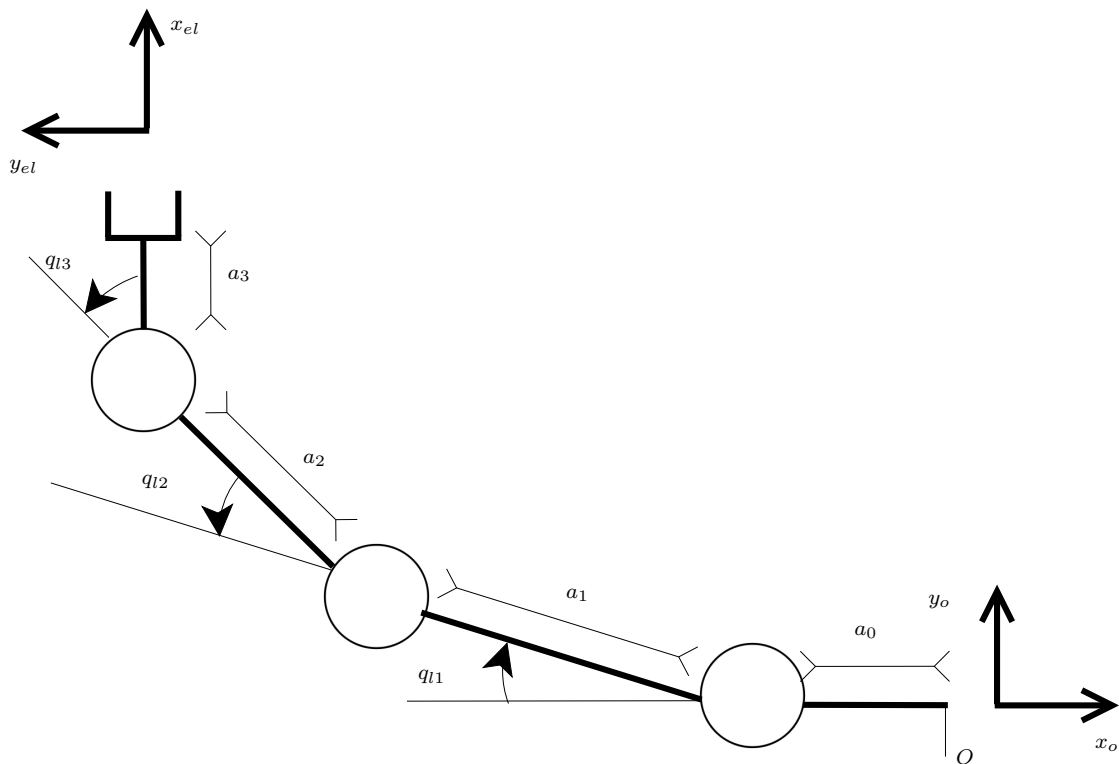


Figura 2.4: Diagrama do braço esquerdo, com rotação e posição do efetuador.

$$P(q_l) = \begin{bmatrix} -a_0 - a_1 \cos(q_{l1}) - a_2 \cos(q_{l1} - q_{l2}) - a_3 \cos(q_{l1} - q_{l2} - q_{l3}) \\ a_1 \sin(q_{l1}) + a_2 \sin(q_{l1} - q_{l2}) + a_3 \sin(q_{l1} - q_{l2} - q_{l3}) \\ \pi - (q_{l1} - q_{l2} - q_{l3}) \end{bmatrix} \quad (2.3)$$

2.1.2 Cinemática Diferencial

Assim como na cinemática direta existe uma matriz de transformação homogênea que relaciona a posição e orientação do efetuador com os ângulos das juntas, na cinemática diferencial existe uma matriz jacobiana que relaciona a velocidade linear e angular do efetuador com a velocidade angular das juntas do manipulador. O Jacobiano é definido como a matriz de todas as derivadas parciais de primeira ordem de uma função vetorial (Equação 2.4). Neste projeto, tal função vetorial é a equação de cinemática direta.

$$J = \frac{dF}{dx} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \dots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \dots & \frac{\partial F_m}{\partial x_n} \end{bmatrix} \quad (2.4)$$

Para o presente projeto, tal matriz será calculada de forma geométrica, devido a simplicidade dos manipuladores utilizados. É possível calcular então, o jacobiano de posição J_p (equação (2.5)) e o jacobiano de orientação J_o (equação (2.6)) com base na equação de cinemática direta e com os dois, enfim, calcular o jacobiano J (equação (2.8)).

$$\dot{p} = J_p(q)\dot{q} \quad (2.5)$$

$$\omega = J_o(q)\dot{q} \quad (2.6)$$

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q)\dot{q} \quad (2.7)$$

$$J(q) = \begin{bmatrix} J_p(q) \\ J_o(q) \end{bmatrix} \quad (2.8)$$

No caso deste trabalho, onde o manipulador é planar e tem apenas três juntas, aplicando as equações (2.2) e (2.3), é possível obter o Jacobiano do manipulador direito e esquerdo deste projeto, dados pelas equações (2.9) e (2.10) respectivamente.

$$J_r(q) = \begin{bmatrix} -a_1 s_{r1} - a_2 s_{r12} - a_3 s_{r123} & a_2 s_{r12} + a_3 s_{r123} & a_3 s_{r123} \\ a_1 c_{r1} + a_2 c_{r12} + a_3 c_{r123} & -a_2 c_{r12} - a_3 c_{r123} & -a_3 c_{r123} \\ 1 & -1 & -1 \end{bmatrix} \quad (2.9)$$

$$J_l(q) = \begin{bmatrix} a_1 s_{l1} + a_2 s_{l12} + a_3 s_{l123} & -a_2 s_{l12} - a_3 s_{l123} & -a_3 s_{l123} \\ a_1 c_{l1} + a_2 c_{l12} + a_3 c_{l123} & -a_2 c_{l12} - a_3 c_{l123} & -a_3 c_{l123} \\ -1 & 1 & 1 \end{bmatrix} \quad (2.10)$$

onde $s_{r1} = \text{sen}(q_{r1})$, $s_{r12} = \text{sen}(q_{r1} - q_{r2})$, $s_{r123} = \text{sen}(q_{r1} - q_{r2} - q_{r3})$, $c_{r1} = \text{cos}(q_{r1})$, $c_{r12} = \text{cos}(q_{r1} - q_{r2})$, $c_{r123} = \text{cos}(q_{r1} - q_{r2} - q_{r3})$ com resultado análogo para s_{l1} , s_{l12} , s_{l123} , c_{l1} , c_{l12} , c_{l123} .

2.2 Robôs Não-holonômicos

Sistemas robóticos não-holonômicos apresentam movimentos limitados, embora possam atingir qualquer posição no espaço onde estão definidos se forem globalmente controláveis e alcançáveis (MURRAY *et al.*, 1994). O estado do sistema depende da trajetória feita pra chegar até ele, mas qualquer estado é alcançável desde que o sistema seja controlável globalmente. Considerando nosso espaço de trabalho em \mathbb{R}^2 , o robô do tipo unicycle (Figura 2.5) que é usado nesse trabalho é não-holonômico. Um robô desse tipo pode ser descrito pelo sistema apresentado na equação (2.11), onde x_{rob} e y_{rob} são coordenadas cartesianas do robô, θ_{rob} é sua orientação em relação a origem, v é sua velocidade linear e ω é sua velocidade angular.

$$\begin{bmatrix} \dot{x}_{rob} \\ \dot{y}_{rob} \\ \dot{\theta}_{rob} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{rob}) & 0 \\ \text{sen}(\theta_{rob}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.11)$$

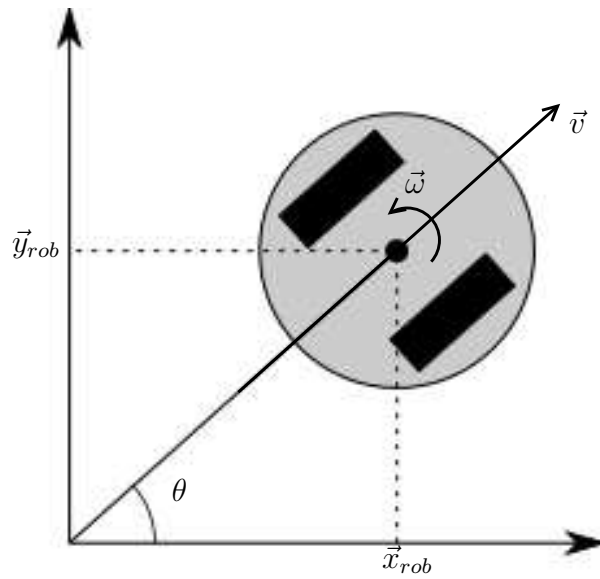


Figura 2.5: Diagrama de um robô não-holonômico do tipo uniciclo.

2.2.1 Restrição Não-Holonômica

O manipulador planar de três elos, visto na seção anterior, é um robô-holonômico. Entretanto, visto que este trabalho se trata sobre a aplicação real de uma lei de controle de formação para agentes não-holonômicos, será imposta, virtualmente, uma restrição não-holonômica ao movimento de tais manipuladores.

Com a equação de cinemática diferencial de um manipulador descrita em (2.7), é possível inferir a equação (2.12) para o manipulador planar de três elos estudado.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \omega \end{bmatrix} = J(q)\dot{q} \quad (2.12)$$

O efetador deste manipulador pode então ser controlado por uma variável \vec{u} (2.13) conforme a equação (2.14). Uma restrição não-holonômica pode ser aplicada na formação de \vec{u} (2.15) e usando esta restrição com a equação (2.14) obtendo o novo modelo (2.16) que é não-holonômico.

$$\vec{u} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\omega} \end{bmatrix} \quad (2.13)$$

$$\dot{q} = J^{-1}(q)\vec{u} \quad (2.14)$$

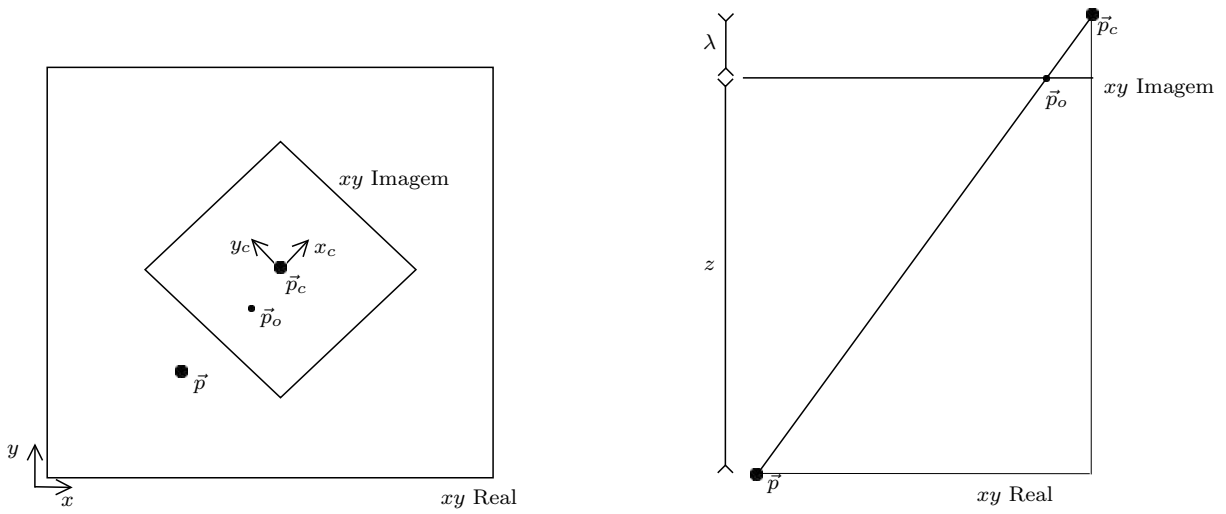


Figura 2.6: Diagrama da configuração entre a câmera e o espaço de trabalho

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.15)$$

$$\dot{q} = J^{-1}(q) \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.16)$$

2.3 Servo Visão

Considere um ambiente de trabalho em \mathbb{R}^2 e uma câmera captando tal espaço de trabalho, perpendicular ao mesmo. A Figura 2.6 mostra, respectivamente, uma visão superior e uma visão lateral deste ambiente de trabalho. Sendo assim a transformada entre um objeto no plano de trabalho real e o mesmo objeto no plano de trabalho da imagem pode ser obtida através de conceitos básicos de geometria e rotação, como mostra a equação (2.17).

$$p_o = \alpha \frac{\lambda}{z} R^T (\vec{p} - \vec{p}_c) \quad (2.17)$$

onde \vec{p} é a posição do objeto no espaço real, \vec{p}_c é a posição da câmera no espaço real, α é o ganho de transformação de metros para *pixels*, z é a distância entre a câmera e o espaço de trabalho, λ é a distância focal e R é a matriz de rotação entre

o plano xy da imagem e o plano xy real.

$$R = \begin{bmatrix} \cos(\phi) & -\text{sen}(\phi) \\ \text{sen}(\phi) & \cos(\phi) \end{bmatrix}, \quad (2.18)$$

onde ϕ é o ângulo entre o plano xy da imagem e o plano xy real

De forma análoga, como se trata de uma rotação planar, o ângulo de orientação do objeto no plano da imagem pode ser trivialmente inferido pela equação (2.19).

$$\theta_{cam} = \theta - \phi \quad (2.19)$$

2.4 Controle de formação por função potencial saturada e deslocada

Deseja-se implementar uma lei de controle de formação descentralizada, ou seja não existe autoridade central controlando toda a formação, que evita colisões entre os agentes e os faz convergir a uma formação desejada, para um conjunto de robôs não-holonômicos, cinemáticos, do tipo unicycle.

Aplicando uma função potencial saturada em cada um dos agentes, é desejado conduzi-los ao mínimo global dessas funções, alcançando a formação desejada enquanto evita-se colisões entre os mesmos.

A função potencial saturada usada para a lei de controle deve atender certos requerimentos para o controle da formação ser estável e evitar colisão (Figura 2.7).

- A função potencial $V_{ij}(\|r_{ij}\|)$ entre dois agentes deve ser uma função não negativa da norma da distância $\|r_{ij}\|$ entre os agentes i e j .
- $V_{ij}(\|r_{ij}\|)$ deve tender a infinito quando $\|r_{ij}\|$ tende a c_i , sendo c_i o diâmetro da zona de colisão do agente i .
- V_{ij} atinge seu mínimo global quando a distância entre os agentes i e j é a distância r_d desejada.
- O gradiente da função V_{ij} vale zero quando o módulo da distância r_{ij} entre os agentes i e j for maior que um determinado valor R_s , que é o raio do campo de vizinhança do agente i

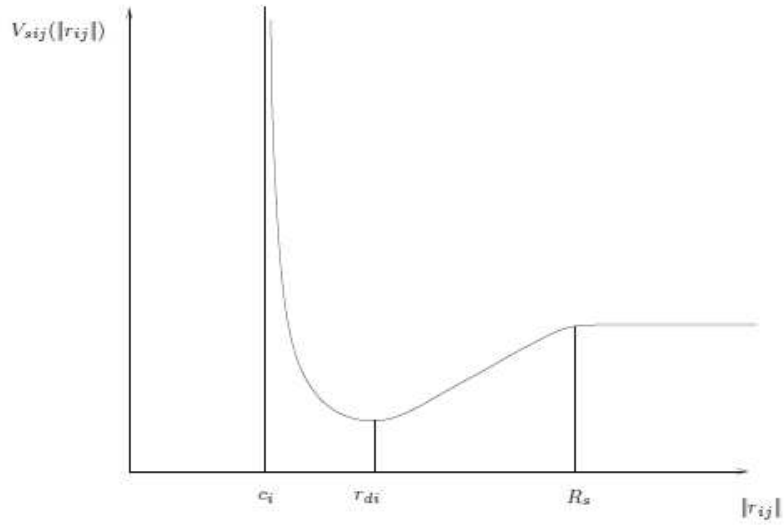


Figura 2.7: Função potencial artificial saturada e deslocada

Considera-se a dimensão do robô i um círculo de diâmetro c_i , como visto nas restrições acima. Apesar de não parecer viável para um robô real, sempre pode ser definido um círculo de raio $\frac{c_i}{2}$ em torno do robô que faça com que todas as dimensões do robô estejam dentro desse círculo, como mostra a Figura (Figura 2.8). Na mesma figura, pode-se ver um outro agente j que está fora da região de vizinhança, definida como um círculo de raio R_s , em torno do agente i . A posição de tal agente não estaria sendo captado pelo agente i e sua posição não influenciaria na decisão de controle do agente. Para garantir a convergência do conjunto de robôs para a formação então, é necessário garantir que grafo de comunicação entre agentes seja, ao menos, conectado (GOUVÊA, 2011).

O campo potencial f_{ij} entre dois agentes é então definido como o gradiente desta função V_{ij} , conforme as equações (2.20) e (2.21).

$$f_{ij} = -\nabla_{r_i} V_{ij}, \text{ onde } r_i = [x_i y_i]. \quad (2.20)$$

$$f_{ij} = - \begin{bmatrix} \frac{\partial V_{ij}}{\partial x} \\ \frac{\partial V_{ij}}{\partial y} \end{bmatrix} \quad (2.21)$$

E a função potencial total sobre um agente i é definida como a soma da função

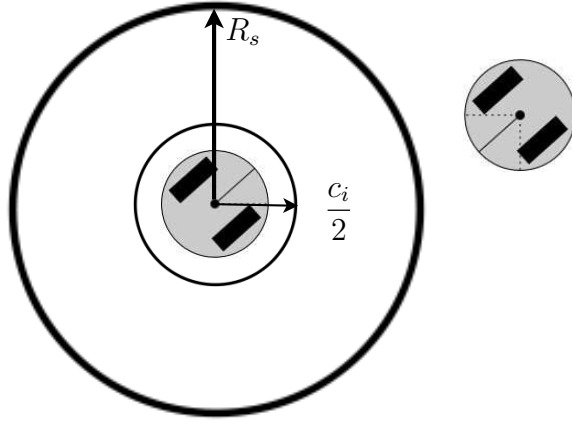


Figura 2.8: Representação das regiões de colisão e vizinhança de um agente i

potencial entre o agente i e todos os N agentes.

$$V_i = \sum_{j=1, j \neq i}^N V_{ij} \quad (2.22)$$

O que então implica na equação (2.23), para o campo total que age sobre o agente i

$$f_i = \sum_{j=1, j \neq i}^N f_{ij} \quad (2.23)$$

O teorema a seguir, proposto e demonstrado em (GOUVÊA, 2011), mostra que a lei de controle de formação descrita pelas equações (2.24) e (2.25) é suficiente para fazer com que os agentes venham a convergir para a formação desejada.

$$v_i = k_v f_i^T S_i, \quad \text{sendo } S_i = \begin{bmatrix} \cos(\theta_i) \\ \text{sen}(\theta_i) \end{bmatrix} \quad (2.24)$$

$$\omega_i = -k(\theta_i - \theta_{id}), \quad \text{sendo } \theta_{id} = \arctan \left(-\frac{\partial V_i}{\partial y_i}, -\frac{\partial V_i}{\partial x_i} \right) \quad (2.25)$$

Teorema 2.1 *Considere um grupo de robôs móveis não-holonômicos com modelo cinemático (2.11) trocando informações através de um grafo não direcionado. Se*

1. cada robô i for considerado circular com diâmetro c_i

2. a condição inicial dos estados da formação z_0 pertencer ao conjunto D onde

$$D = z \mid W_1(z) \leq W_0, \quad (2.26)$$

$W_0 > 0$,

$$2W_1 = \sum_{i=1}^N [V_i + \alpha(\theta_i - \theta_{id})^2], \quad (2.27)$$

a função potencial V_i é dada por (2.22), $\alpha > 0$ e

$$z^T = \left[(\theta_1 - \theta_{1d}) \quad \dots \quad (\theta_N - \theta_{Nd}) \quad r_1^T \quad \dots \quad r_N^T \right] \quad (2.28)$$

,

3. os sinais de controle u_i e ω_i forem dados respectivamente por (2.24) e (2.25),

4. o grafo de comunicação for sempre conectado,

então a colisão entre agentes é evitada e a formação alcança uma configuração que minimiza V

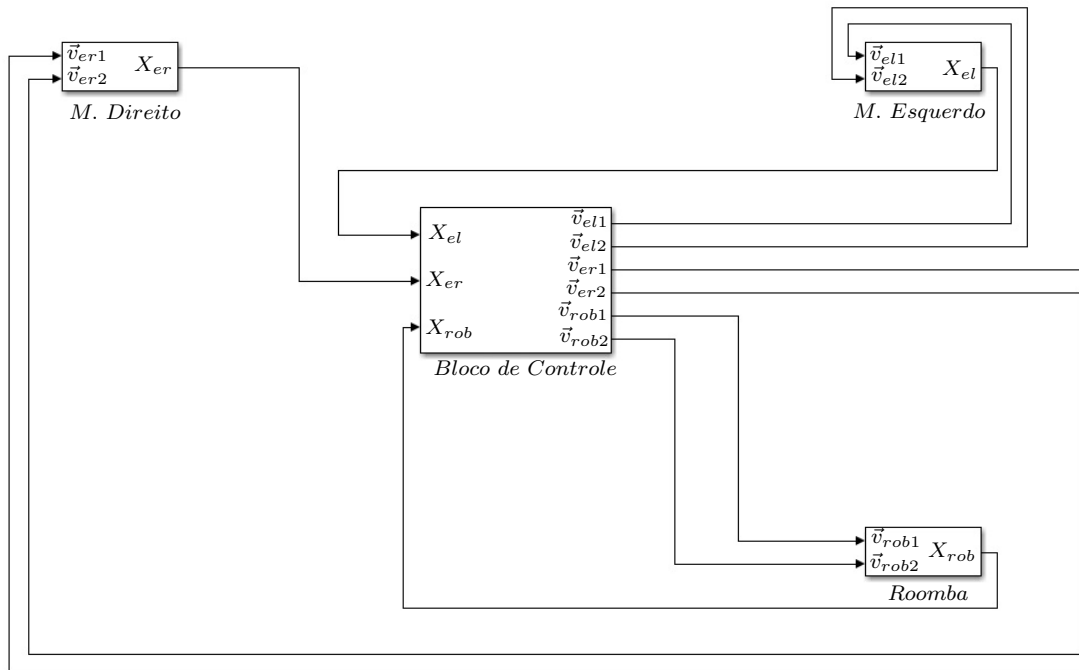


Figura 3.1: Modelo geral, com as três plantas e o bloco de controle.

Capítulo 3

Modelagem e Simulação

Na primeira parte deste capítulo, será apresentado a modelagem dos agentes deste projeto, em acordo com os fundamentos teóricos discutidos no capítulo anterior. Será discutido o modelo de cada uma das três plantas, o que inclui a modelagem da aquisição de dados por uma câmera ideal em cada uma das plantas, e depois do sistema de controle em si, apresentado como um todo. Tal modelo pode ser visto na Figura (Figura 3.1). Na parte posterior deste capítulo, serão realizadas simulações em condições diversas, com breves explicações do significado de tais condições e do resultado obtido com cada uma delas.

3.1 Modelagem

Nesta seção, será apresentado a modelagem específica de cada um dos quatro blocos mostrados na Figura (3.1), na seguinte ordem:

- Primeiro o modelo do roomba, por ser facilmente aproximado de um robô cinemático não-holonômico do tipo uniciclo.
- Os manipuladores esquerdo e direito do motoman, respectivamente, por representarem os demais agentes não-holonômicos do sistema com a condição imposta sobre eles.
- Por último, o bloco de controle, revelando qual a função potencial escolhida para o controle do sistema em questão.

3.1.1 Roomba

O *iRobot* Roomba é modelado como um robô não-holonômico cinemático, controlado pela velocidade de suas rodas esquerda e direita, em milímetros por segundo (Figura 3.2). Seu modelo então, pode ser praticamente descrito como no capítulo anterior, pela equação (2.11), porém o roomba é um robô diferencial, controlado pela velocidade de suas rodas. Considerando que as velocidades v_{rob1} e v_{rob2} são, respectivamente, as velocidades da roda direita e da roda esquerda do Roomba, em milímetros por segundo, é possível relacionar sua velocidade linear v_{rob} e sua velocidade angular ω_{rob} com a equação (3.1)

$$\begin{bmatrix} v_{rob} \\ \omega_{rob} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2r} & \frac{-1}{2r} \end{bmatrix} \begin{bmatrix} v_{rob1} \\ v_{rob2} \end{bmatrix} \quad (3.1)$$

onde r é a distância entre o centro de massa do roomba e suas rodas.

Então, em conjunto com a equação (2.11) e trocando-se a unidade para metro por segundo, obtém-se o modelo (3.2)

$$\begin{bmatrix} \dot{x}_{rob} \\ \dot{y}_{rob} \\ \dot{\theta}_{rob} \end{bmatrix} = \begin{bmatrix} 10^{-3} & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_{rob}) & 0 \\ \sin(\theta_{rob}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2r} & \frac{-1}{2r} \end{bmatrix} \begin{bmatrix} v_{rob1} \\ v_{rob2} \end{bmatrix} \quad (3.2)$$

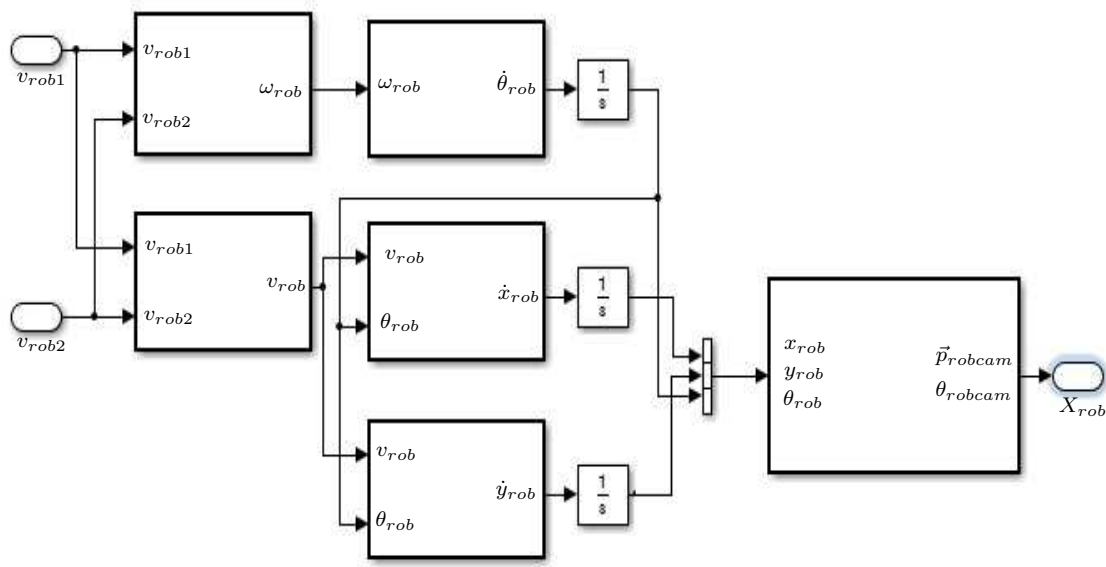


Figura 3.2: Modelo do Roomba

Com a obtenção da posição e orientação no plano real, modela-se com as equação (3.3) que representa a transformação do plano real para plano da imagem da câmera, como mostrado no capítulo anterior, nas equações (2.17), (2.18) e (2.19).

$$X_{rob} = \begin{bmatrix} x_{robcam} \\ y_{robcam} \\ \theta_{robcam} \end{bmatrix} = \alpha \frac{\lambda}{z_{rob}} \begin{bmatrix} \cos(\phi) & \text{sen}(\phi) & 0 \\ \text{sen}(-\phi) & \cos(\phi) & 0 \\ 0 & 0 & \frac{z_{rob}}{\alpha\lambda} \end{bmatrix} \left(\begin{bmatrix} x_{rob} \\ y_{rob} \\ \theta_{rob} \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \\ \phi \end{bmatrix} \right) \quad (3.3)$$

3.1.2 Manipuladores

Como já abordado antes, um manipulador planar de três elos por si só, é um robô holonômico. Isto é, com suas três juntas, é possível controlar todos os três graus de liberdade do efetuador simultaneamente. Para o caso de estudo, entretanto, será limitado o movimento dos mesmos para que eles se comportem como agentes não-holonômicos.

Serão virtualmente impostos robôs diferenciais ao efetuador dos manipuladores, controlando-os como se fosse dois outros roombas. Tal restrição, será tratada, durante a modelagem do sistema, como parte integrante da planta dos manipuladores, a serem descritos a seguir. Isso faz com que as variáveis manipuláveis para tais

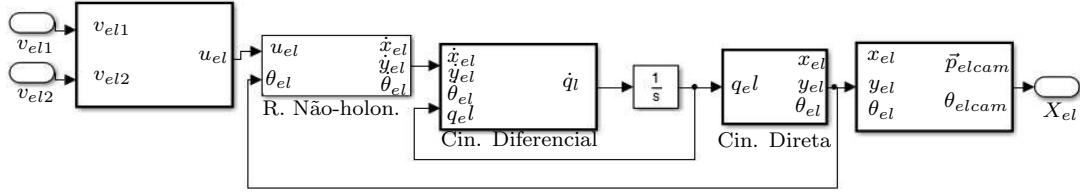


Figura 3.3: Modelo do Manipulador Esquerdo

braços robóticos sejam v_1 e ω_2 e não o ângulo das juntas q como seria esperado.

Manipulador Esquerdo

O manipulador esquerdo, conforme abordado no capítulo anterior, tem certas especificações de fábrica que devem ser levadas em consideração durante a modelagem. O ângulo q_{l1} tem incrementos positivos quando sua junta correspondente rotaciona no sentido horário, enquanto q_{l2} e q_{l3} tem incrementos positivos quando suas respectivas juntas rotacionam no sentido anti-horário. A Figura (3.3) mostra o modelo de tal manipulador, controlado por v_{el} e ω_{el} (equação 3.4).

$$u_{el} = \begin{bmatrix} v_{el} \\ \omega_{el} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2r} & \frac{-1}{2r} \end{bmatrix} \begin{bmatrix} v_{el1} \\ v_{el2} \end{bmatrix} \quad (3.4)$$

É possível então resolver a restrição não-holonômica conforme mostra a equação (3.5).

$$\begin{bmatrix} \dot{x}_{el} \\ \dot{y}_{el} \\ \dot{\theta}_{el} \end{bmatrix} = \begin{bmatrix} 10^{-3} & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_{el}) & 0 \\ \sin(\theta_{el}) & 0 \\ 0 & 1 \end{bmatrix} u_{el} \quad (3.5)$$

Com tal resolução, o modelo prossegue para a resolver a cinemática diferencial, como feito na equação (2.16).

$$\dot{q}_{el} = J_l^{-1}(q) \begin{bmatrix} \dot{x}_{el} \\ \dot{y}_{el} \\ \dot{\theta}_{el} \end{bmatrix} \quad (3.6)$$

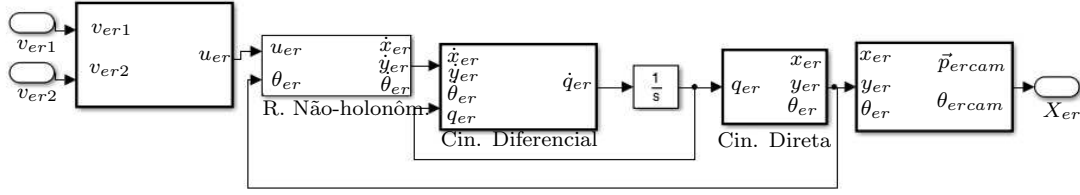


Figura 3.4: Modelo do Manipulador Direito

De posse da equação (3.6), atualiza-se ângulo das juntas e é calculada a nova posição do efetuador através da cinemática direta, com a equação (3.7).

$$\begin{bmatrix} x_{el} \\ y_{el} \\ \theta_{el} \end{bmatrix} = \begin{bmatrix} -a_0 - a_1 \cos(q_{l1}) - a_2 \cos(q_{l1} - q_{l2}) - a_3 \cos(q_{l1} - q_{l2} - q_{l3}) \\ a_1 \sin(q_{l1}) + a_2 \sin(q_{l1} - q_{l2}) + a_3 \sin(q_{l1} - q_{l2} - q_{l3}) \\ \pi - (q_{l1} - q_{l2} - q_{l3}) \end{bmatrix} \quad (3.7)$$

Então, finalmente, adiciona-se a transformação proporcionada pela câmera a, obtendo os estados finais do efetuador do manipulador esquerdo.

$$X_{el} = \begin{bmatrix} x_{elcam} \\ y_{elcam} \\ \theta_{elcam} \end{bmatrix} = \alpha \frac{\lambda}{z_{el}} \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ \sin(-\phi) & \cos(\phi) & 0 \\ 0 & 0 & \frac{z_{el}}{\alpha\lambda} \end{bmatrix} \left(\begin{bmatrix} x_{el} \\ y_{el} \\ \theta_{el} \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \\ \phi \end{bmatrix} \right) \quad (3.8)$$

Manipulador Direito

O manipulador direito tem modelagem análoga a do manipulador esquerdo, como pode ser visto na Figura (3.4). A diferença entre eles vai ser na modelagem da cinemática diferencial e na cinemática direta. No manipulador direito, o ângulo q_{r1} tem incrementos positivos quando sua junta correspondente rotaciona no sentido anti-horário, enquanto q_{r2} e q_{r3} tem incrementos positivos quando suas respectivas juntas rotacionam no sentido horário.

$$u_{er} = \begin{bmatrix} v_{er} \\ \omega_{er} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2r} & \frac{-1}{2r} \end{bmatrix} \begin{bmatrix} v_{er1} \\ v_{er2} \end{bmatrix} \quad (3.9)$$

Na mesma sequência feita para o manipulador esquerdo, calcula-se a restrição

não-holonômica, na equação (3.10).

$$\begin{bmatrix} \dot{x}_{er} \\ \dot{y}_{er} \\ \dot{\theta}_{er} \end{bmatrix} = \begin{bmatrix} 10^{-3} & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_{er}) & 0 \\ \text{sen}(\theta_{er}) & 0 \\ 0 & 1 \end{bmatrix} u_{er} \quad (3.10)$$

Novamente, o modelo prossegue para a resolver a cinemática diferencial, como feito na equação (2.16).

$$\dot{q}_{er} = J_r^{-1}(q) \begin{bmatrix} \dot{x}_{er} \\ \dot{y}_{er} \\ \dot{\theta}_{er} \end{bmatrix} \quad (3.11)$$

Atualizam-se os ângulos q_r e o modelo calcula a posição e orientação do efetuador através da cinemática direta.

$$\begin{bmatrix} x_{er} \\ y_{er} \\ \theta_{er} \end{bmatrix} = \begin{bmatrix} a_0 + a_1 \cos(q_{r1}) + a_2 \cos(q_{r1} - q_{r2}) + a_3 \cos(q_{r1} - q_{r2} - q_{r3}) \\ a_1 \text{sen}(q_{r1}) + a_2 \text{sen}(q_{r1} - q_{r2}) + a_3 \text{sen}(q_{r1} - q_{r2} - q_{r3}) \\ q_{r1} - q_{r2} - q_{r3} \end{bmatrix} \quad (3.12)$$

E finalmente, a atuação da câmera sobre os estados do sistema.

$$X_{er} = \begin{bmatrix} x_{ercam} \\ y_{ercam} \\ \theta_{ercam} \end{bmatrix} = \alpha \frac{\lambda}{z_{el}} \begin{bmatrix} \cos(\phi) & \text{sen}(\phi) & 0 \\ \text{sen}(-\phi) & \cos(\phi) & 0 \\ 0 & 0 & \frac{z_{el}}{\alpha \lambda} \end{bmatrix} \left(\begin{bmatrix} x_{er} \\ y_{er} \\ \theta_{er} \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \\ \phi \end{bmatrix} \right) \quad (3.13)$$

3.1.3 Bloco de Controle

O bloco de controle (Figura 3.5) foi projetado para calcular individualmente o controle u de cada agente, porém todos funcionam de forma bem similar. A função potencial saturada e deslocada usada nesse projeto (equação 3.14) foi escolhida por sua simplicidade, sendo uma pequena variação da função potencial saturada proposta por (TANNER *et al.*, 2003), vista em (GOUVÊA, 2011).

$$V_{ij}(\|r_{ij}\|) = \begin{cases} J(\|r_{ij}\|) & \text{para } (\|r_{ij}\| < R_s) \\ J(R_s) & \text{para } (\|r_{ij}\| \geq R_s) \end{cases} \quad (3.14)$$

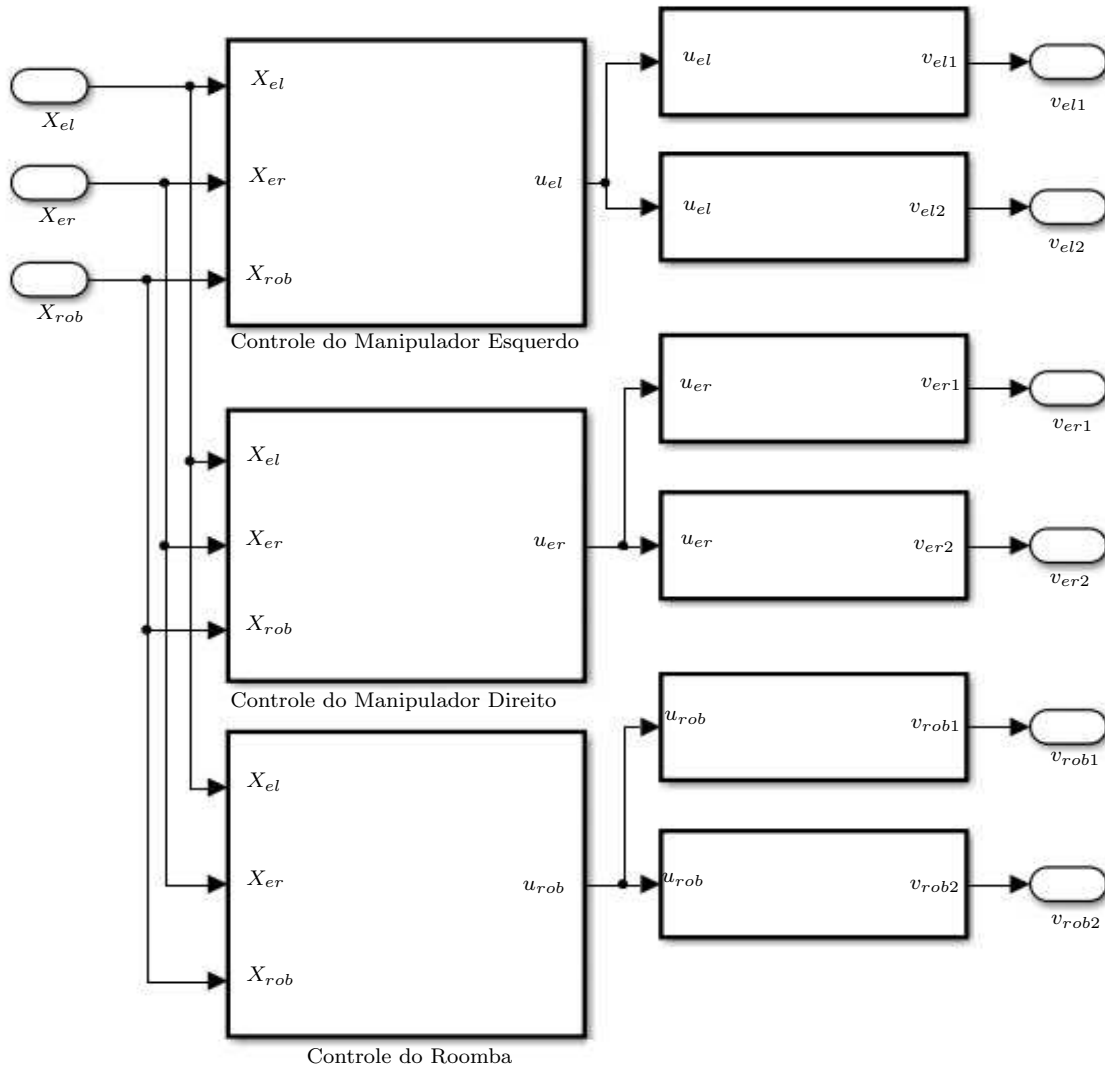


Figura 3.5: Modelo do Bloco de Controle

onde,

$$J(\|r_{ij}\|) = \log(\|r_{ij}\| - c_i) + \frac{a_2}{\|r_{ij}\| - c_i} + a_1(\|r_{ij}\| - c_i) \quad (3.15)$$

sendo,

$$a_1 = \frac{1}{r_{di} - 2c_i + R_s}$$

$$a_2 = \frac{(R_s - c_i)(r_{di} - c_i)}{r_{di} - 2c_i + R_s}$$

onde, por conveniência, $c_i = c$ e $r_{di} = r_d$.

Com essa função potencial, calcula-se o controle conforme descrito pelas equações (2.23), (2.24) e (2.25).

$$v_i = k_v \left(\sum_{j=1, j \neq i}^3 \begin{bmatrix} \frac{\partial V_i}{\partial x} & \frac{\partial V_i}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \cos(\theta_i) \\ \text{sen}(\theta_i) \end{bmatrix} \quad (3.16)$$

$$\omega_i = -k(\theta_i - \arctan \left(-\frac{\partial V_i}{\partial y_i}, -\frac{\partial V_i}{\partial x_i} \right)) \quad (3.17)$$

E então o controle é

$$u_i = \begin{bmatrix} v_i \\ w_i \end{bmatrix} \quad (3.18)$$

Por fim, troca-se o controle para um robô diferencial, controlado por velocidade das rodas em milímetros por segundo

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 10^3 \begin{bmatrix} 1 & r \\ 1 & -r \end{bmatrix} u_{rob} \quad (3.19)$$

3.2 Simulações

Neste segmento do projeto, será apresentado o resultado das simulações usando o modelo descrito, em três etapas:

- Apenas um dos agentes irá buscar a formação, com os demais agentes imóveis.
- Um par de agentes irá buscar a formação, indo atrás do outro agente imóvel.
- Todos os três agentes irão buscar a formação.

Como, por conveniência, a distância ideal sugerida entre os agentes são iguais, quando dois ou três agentes se buscam a formação, eles formam um triângulo equilátero no plano da imagem, se apenas um agente se mover, a melhor formação possível é um triângulo isósceles. Nestas simulações, o *Agente 1* é o manipulador esquerdo, o *Agente 2* é o manipulador direito e o *Agente 3* é o roomba.

Como o objetivo das simulações a seguir é testar o controle que vai ser aplicado no ambiente real, certos parâmetros, mostrados na Tabela (3.1), serão mantidos constantes por todas as simulações:

Tabela 3.1: Parâmetros Iniciais da Simulação

Parâmetro	Valor
a_0	$0.38m$
a_1	$0.39m$
a_2	$0.33m$
a_3	$0.21m$
$\alpha\lambda$	355
z_{roomba}	$1.03m$
z_{manip}	$1.03m$
x_{o-Cam}	$-0.8m$
y_{o-Cam}	$1.6m$
c	$100pixels$
R_s	$1000pixels$
r	$0.125m$

Outros fatores também foram levados em consideração, que não estão entre os valores até então tabelados. Não foram escolhidos ângulos iniciais para os manipuladores que fizessem o efetuador ficar fora do espaço de trabalho da câmera. Analogamente, as posições iniciais do roomba foram escolhidas segundo o mesmo critério. Trajetórias foram escolhidas de forma a evitar as limitações das juntas do motoman e também singularidades dos mesmos manipuladores. Embora tais decisões limitem o espaço de trabalho e as trajetórias traçadas, são decisões necessárias para que as simulações se aproximem o máximo possível do ambiente de trabalho real, servindo como informativo sobre o que pode acontecer no mesmo.

3.2.1 Um agente buscando a formação

Nesta seção, será estudado o caso em que apenas um agente sofre ação do campo potencial, os demais tendo velocidade linear v_i e velocidade angular ω_i nulos. Neste caso, a formação alcançada é um triângulo isósceles, não havendo alteração na distância entre os dois agentes que não se movem. Serão realizadas duas simulações para cada agente, com diferentes condições iniciais .

Tabela 3.2: Parâmetros para controle de um agente - Roomba Caso 1

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	206.94 <i>pixels</i>	429.67 <i>pixels</i>	241.43 pixels
$y(0)$	-336.78 <i>pixels</i>	-342.91 <i>pixels</i>	-137.96 pixels
$\theta(0)$	0	π	$\frac{-\pi}{2}$

Roomba - Caso 1

Os parâmetros k_v e k foram sintonizados de forma empírica. O parâmetro ϕ foi escolhido de acordo com as possibilidades de rotação da câmera instalada no local de trabalho. A distância desejada r_d foi escolhida de acordo com o planejamento de trajetória adequado. Os parâmetros iniciais dos agentes como capturados pela câmera, por sua vez, estão especificados na Tabela (3.2).

- $k_v = 20$,
- $k = 0.5$,
- $\phi = 0$,
- $r_d = 200$,
- $q_{el} = \begin{bmatrix} \frac{\pi}{3} \\ -\pi \\ \frac{\pi}{3} \\ -\pi \\ \frac{\pi}{3} \end{bmatrix}$
- $q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\pi \\ \frac{\pi}{4} \\ -\pi \\ \frac{\pi}{2} \end{bmatrix}$
- $\begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} -0.1m \\ 1.2m \\ -\pi \\ \frac{\pi}{2} \end{bmatrix}$

A Figura (3.6) mostra a trajetória do roomba na câmera até alcançar a formação, enquanto a Figura (3.7). Observe que como um robô não-holonômico, ele não segue a trajetória mais curta até alcançar tal posição desejada. A Figura (3.8) mostra que

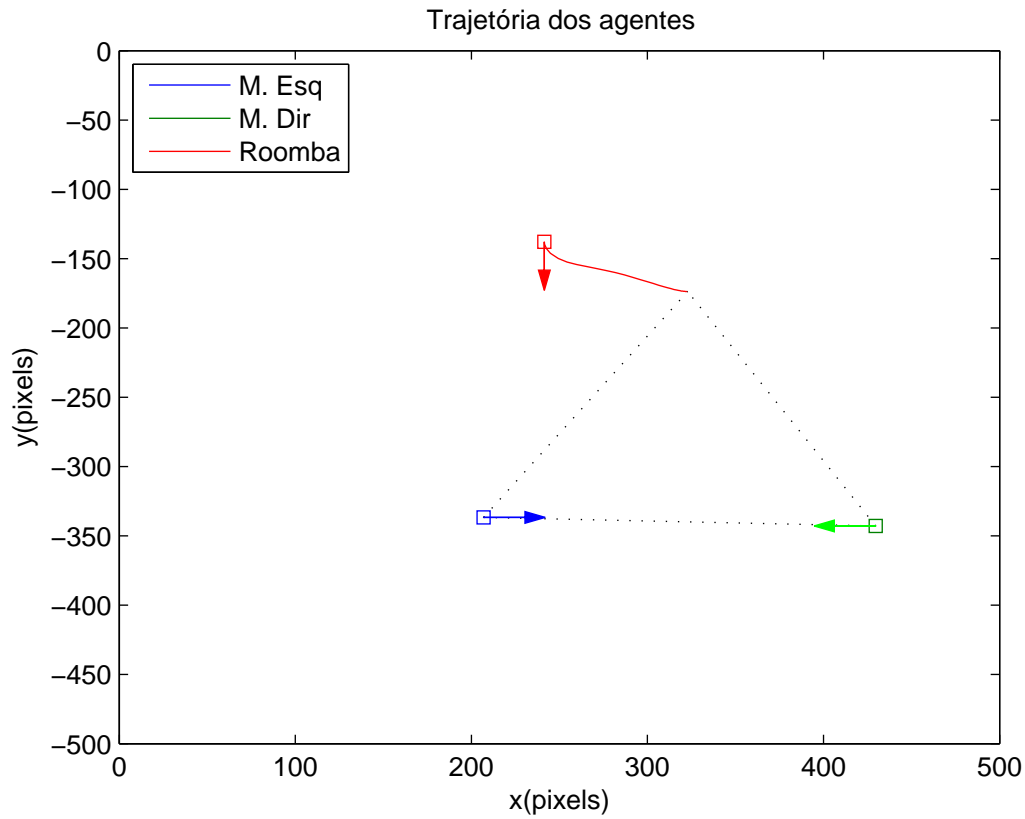


Figura 3.6: Trajetória dos agentes - Roomba Caso 1

o roomba estava numa posição inicial muito próxima da distância desejada entre o mesmo e o manipulador esquerdo, se aproximando do ainda mais do mesmo enquanto o roomba corrige sua orientação, variação máxima entre a distância desejada e a distância entre os agentes 1 e 3 ocorre em $t = 3.6s$ com uma amplitude de $13.3pixels$. Outro importante dado que pode ser coletado da mesma Figura (3.8) é o tempo que o sistema leva até chegar a distância r_d desejada que minimiza a função potencial para o roomba, no caso, em torno de 22 segundos.

Roomba - Caso 2

Os parâmetros k_v e k foram sintonizados de forma empírica, com o intuito de acelerar a correção do erro de orientação, quando comparado ao caso anterior. O parâmetro ϕ também foi escolhido como um valor limite, neste caso não-nulo, que a instalação da câmera permite. Os parâmetros iniciais dos agentes como capturados pela câmera, por sua vez, estão especificados na Tabela (3.3).

- $k_v = 20$,

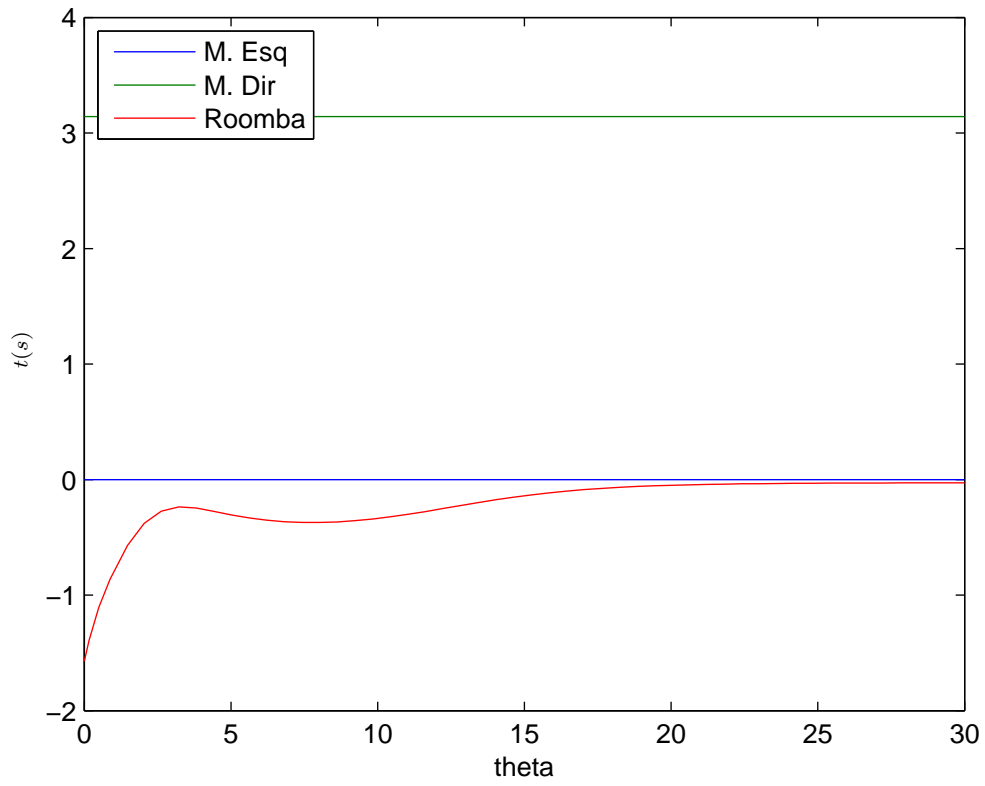


Figura 3.7: Orientação dos agentes - Roomba Caso 1

- $k = 20,$

- $\phi = \frac{\pi}{6},$

- $r_d = 250,$

- $q_{el} = \begin{bmatrix} \frac{\pi}{3} \\ -\pi \\ \frac{\pi}{3} \\ -\pi \\ \frac{\pi}{3} \end{bmatrix}$

- $q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\pi \\ \frac{\pi}{4} \\ -\pi \\ \frac{\pi}{2} \end{bmatrix}$

- $\begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} 0.5m \\ 1m \\ -\pi \\ \frac{\pi}{2} \end{bmatrix}$

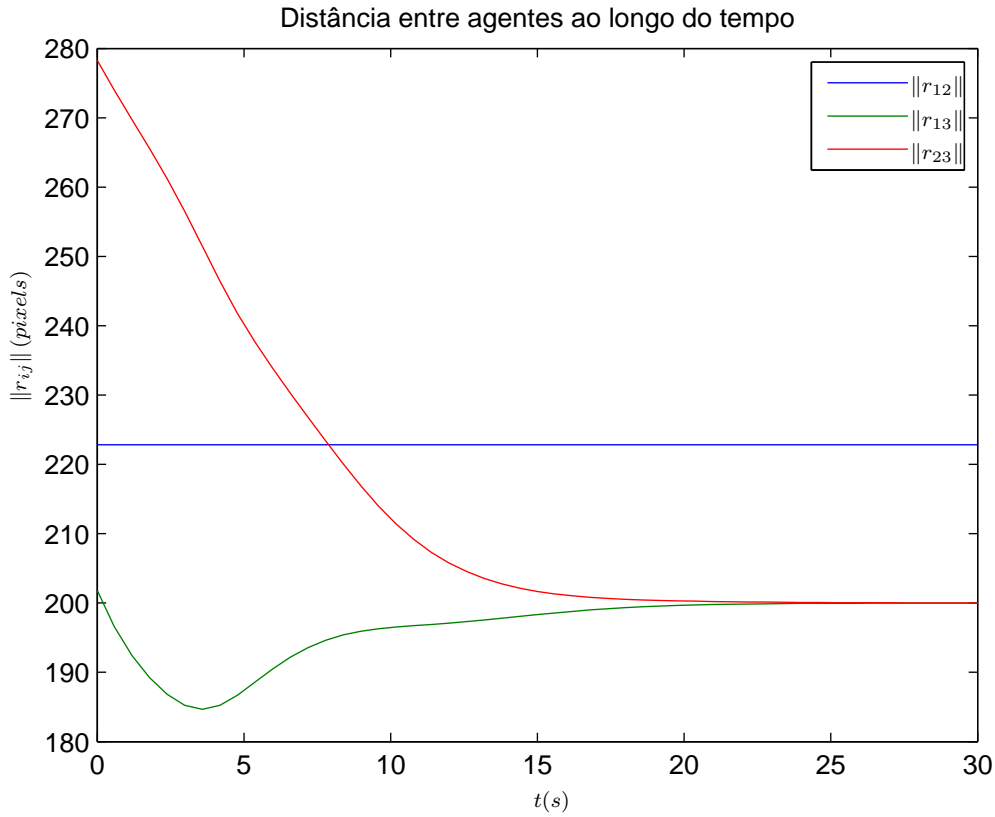


Figura 3.8: Distância entre agentes - Roomba Caso 1

Tabela 3.3: Parâmetros para controle de um agente - Roomba Caso 2

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$10.82pixels$	$200.65pixels$	$284.83 pixels$
$y(0)$	$-395.13pixels$	$-511.81pixels$	$-403.40 pixels$
$\theta(0)$	$\frac{-\pi}{6}$	$\frac{5\pi}{6}$	$\frac{-2\pi}{3}$

A Figura (3.9) mostra a trajetória do roomba na câmera até alcançar a formação, enquanto a Figura (3.10) mostra a orientação dos agentes. Observe que a posição inicial do roomba faz com que o mesmo esteja muito próximo do manipulador direito, $r_{23} = 137pixels$, considerando que os agentes tem uma região de colisão de 100 pixels de diâmetro. O interação entre o roomba e o manipulador direito tem magnitude muito maior do que o roomba e o manipulador esquerdo, suficientemente forte para que a prioridade do controle seja afastar o roomba e o manipulador direito, mesmo a custo de fazer com que o roomba e o manipulador esquerdo se afastem ainda mais. Entretanto, o grande aumento do ganho de orientação, faz com que não

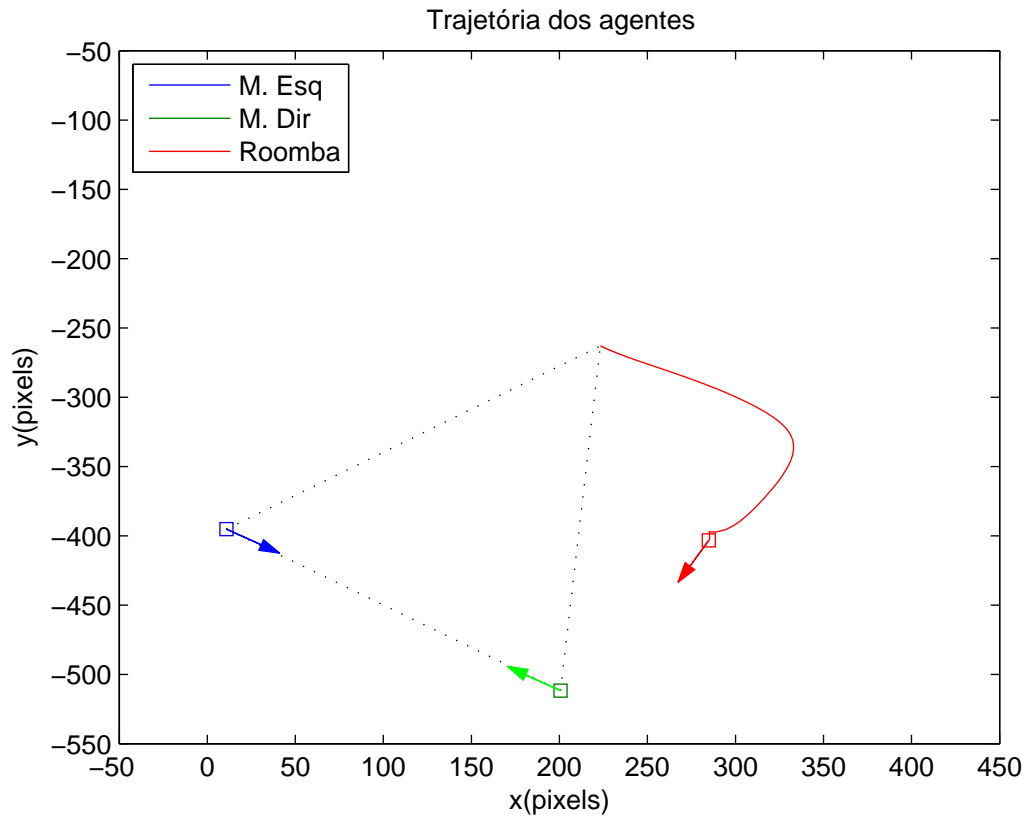


Figura 3.9: Trajetória dos agentes - Roomba Caso 2

exista sobressinal para a distância entre os agentes 2 e 3. enque deveria ter uma correção de orientação mais severa devido ao aumento do ganho k , parece mais suave. Vale ressaltar que o sistema de 2 segundos nessa configuração de rápida variação de distância, suavizando a curva após esse momento, até $t = 68$ segundos quando o sistema atinge a formação.

Manipulador Esquerdo - Caso 1

Os parâmetros k_v e k foram novamente sintonizados como desejado, agora para o manipulador esquerdo, priorizando a velocidade linear. O parâmetro ϕ foi neste caso escolhido como nulo, que apesar de impraticável, é um valor bem próximo dos ângulos pequenos que serão usados com a câmera no espaço de trabalho. A distância desejada r_d foi escolhida de acordo com o planejamento de trajetória adequado. Os parâmetros iniciais dos agentes como capturados pela câmera, por sua vez, estão especificados na Tabela (3.4).

O planejamento de trajetória, para todos os manipuladores, leva em consideração

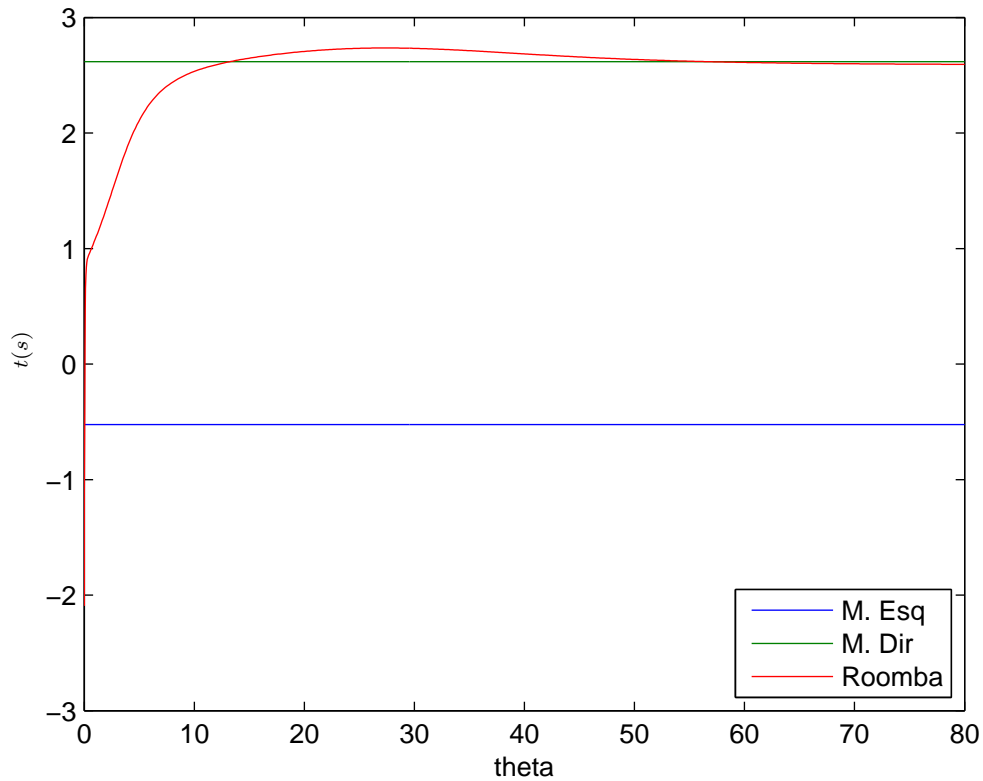


Figura 3.10: Orientação dos agentes - Roomba Caso 2

um fato extra. Devido as limitações das juntas do motoman, principalmente no espaço de trabalho limitado pela câmera, é implausível considerar que, em qualquer configuração de posição e orientação do efetuador, ele consiga realizar grandes rotações no efetuador com pouco movimento linear. Portanto, nas trajetórias traçadas pelos manipuladores, mesmo em simulação, serão evitados grandes sobresinais e raios de curvatura pequenos em sua trajetória, situação na qual, como só está sendo controlado um único agente, o efetuador teria que ter alta velocidade angular para corrigir sua orientação.

- $k_v = 20$,
- $k = 0.5$,
- $\phi = 0$,
- $r_d = 200$,

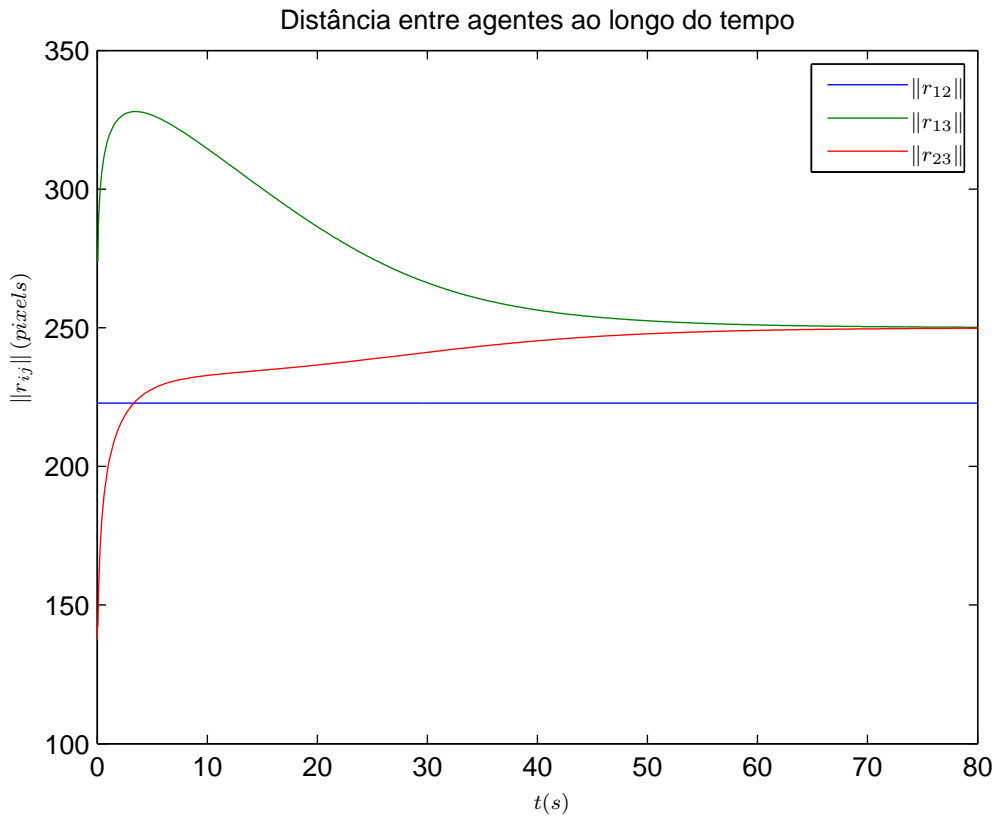


Figura 3.11: Distância entre agentes - Roomba Caso 2

$$\bullet q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{4} \\ -\frac{\pi}{2} \end{bmatrix}$$

$$\bullet q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{4} \\ -\frac{\pi}{2} \end{bmatrix}$$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} -0.1m \\ 1.3m \\ 0 \end{bmatrix}$$

A Figura (3.12) mostra a trajetória do efetuador do manipulador esquerdo, enquanto a Figura (3.13) mostra a orientação. É fácil ver que não há inversão de direção em sua trajetória, e com a Figura (3.14) também fica claro que não há sobressinal, em nenhuma das distâncias entre agentes. O sistema leva em torno de

Tabela 3.4: Parâmetros para controle de um agente - M. Esq. Caso 1

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$206.94pixels$	$429.67pixels$	$241.43 pixels$
$y(0)$	$-336.78pixels$	$-342.91pixels$	$-103.47 pixels$
$\theta(0)$	0	π	0

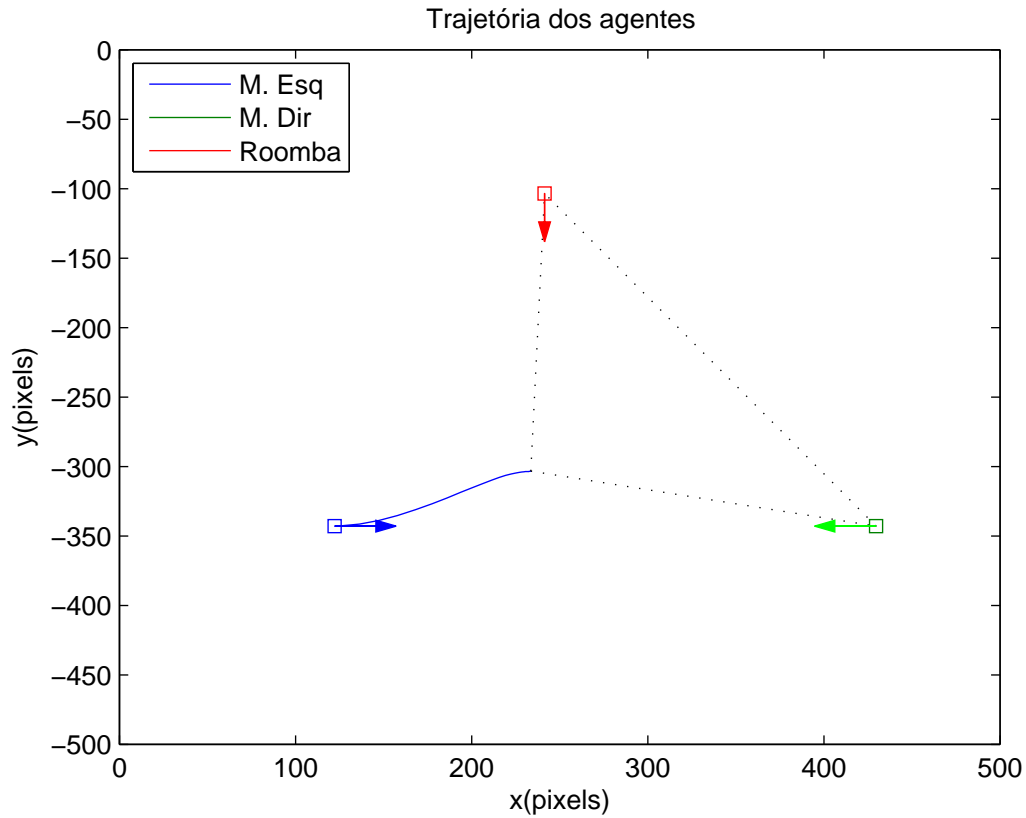


Figura 3.12: Trajetória dos agentes - M. Esq. Caso 1

14 segundos para atingir a distância r_d desejada, minimizando a função potencial. Resta analisar, no caso seguinte, se a rotação da câmera irá ter a mesma atuação sobre o sistema que no caso anterior, apenas alterando suas configurações no espaço porém não no tempo de resposta do mesmo.

Manipulador Esquerdo - Caso 2

Os parâmetro k , k_v não foram alterados, como mencionado anteriormente. ϕ foi novamente escolhido para testar os efeitos da rotação da câmera no sistema. A distância desejada r_d foi mantida em relação as simulações anteriores. Os parâmetros

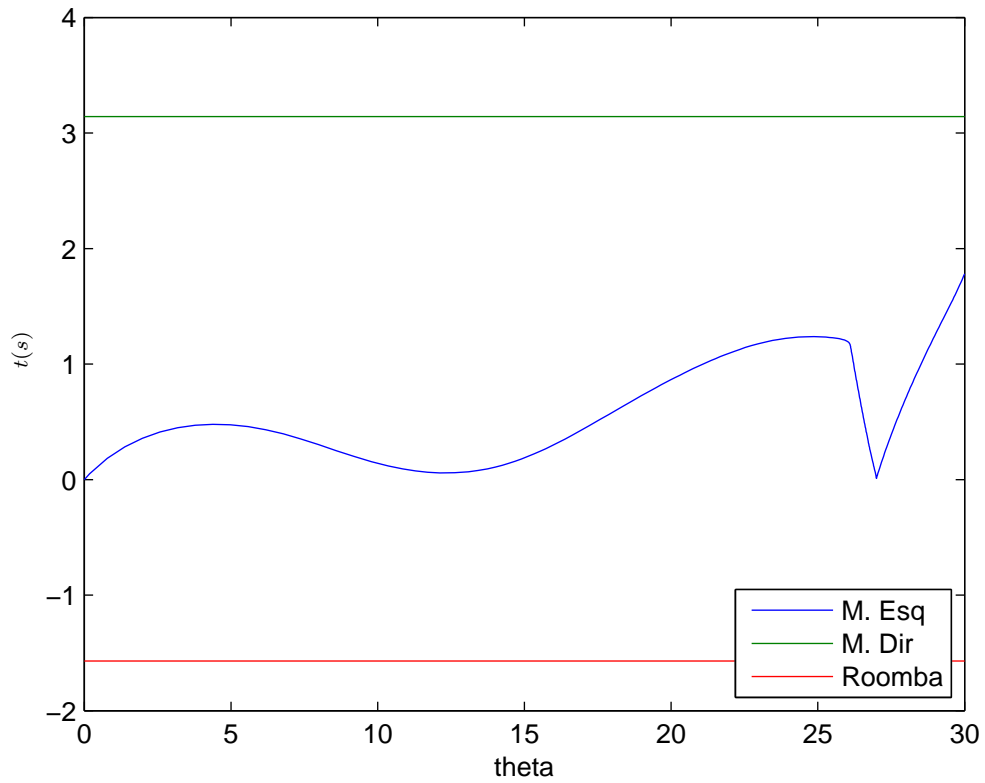


Figura 3.13: Orientação dos agentes - M. Esq. Caso 1

iniciais dos agentes, como capturados pela câmera, estão especificados na Tabela (3.5).

- $k_v = 20,$

- $k = 0.5,$

- $\phi = \frac{\pi}{6},$

- $r_d = 200,$

- $q_{el} = \begin{bmatrix} \frac{\pi}{3} \\ \frac{3}{- \pi} \\ \frac{3}{- \pi} \\ \frac{3}{3} \end{bmatrix}$

- $q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ \frac{4}{- \pi} \\ \frac{4}{- \pi} \\ \frac{2}{2} \end{bmatrix}$

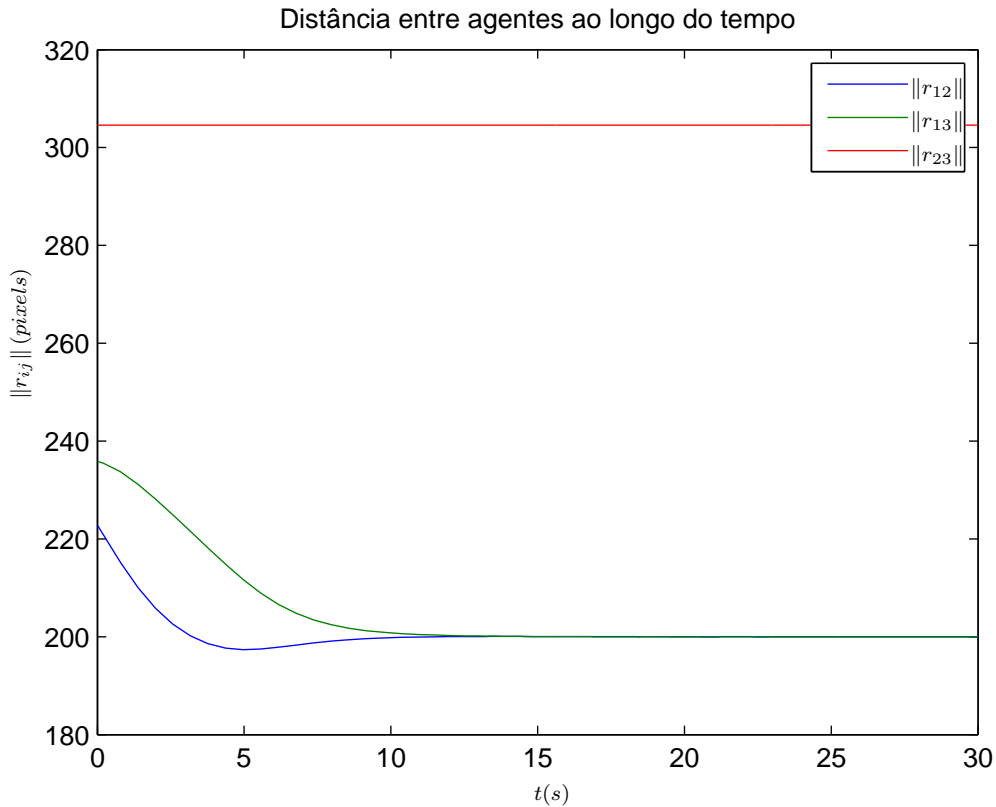


Figura 3.14: Distância entre agentes - M. Esq. Caso 1

Tabela 3.5: Parâmetros para controle de um agente - M. Esq. Caso 2

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$10.82pixels$	$200.65pixels$	187.22 pixels
$y(0)$	$-395.13pixels$	$-511.81pixels$	-227.57 pixels
$\theta(0)$	$\frac{-\pi}{6}$	$\frac{5\pi}{6}$	$\frac{-\pi}{6}$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} -0.1m \\ 1.3m \\ 0 \end{bmatrix}$$

A Figura (3.15) mostra a trajetória do efetuador do manipulador esquerdo, enquanto a Figura (3.16) mostra a orientação. A Figura (3.17) mostra claramente que o sistema, com relação a distância entre agentes, tem exatamente a mesma resposta que o caso anterior. Da mesma figura, é possível ver que o sistema continua levando em torno de 14 segundos para atingir a distância r_d desejada, minimizando a função potencial. Tudo indica que a rotação da câmera não traz grandes diferenças na

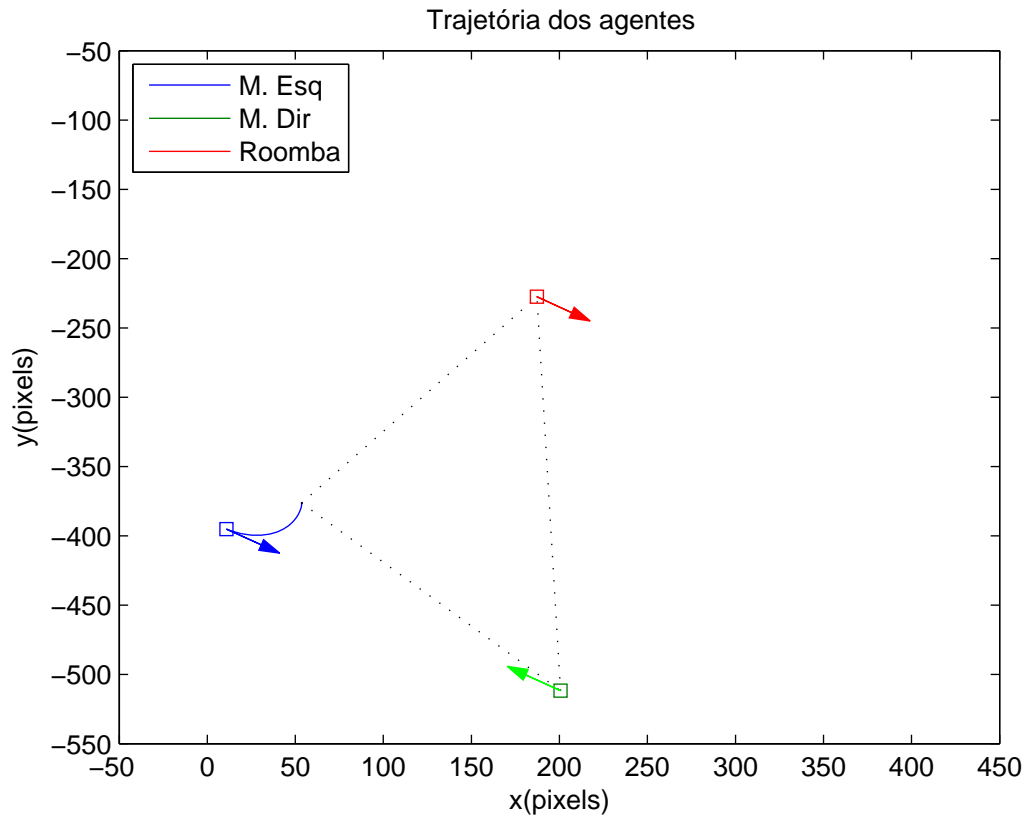


Figura 3.15: Trajetória dos agentes - M. Esq. Caso 2

resposta das distâncias entre agentes, $\|r_{ij}\|$, no tempo.

Manipulador Direito - Caso 1

Nesta simulação e na seguinte, será simulada a atuação do terceiro agente, o manipulador direito. Diferente das simulações anteriores, nesta será testada a resposta do sistema a ganhos maiores, como listado nos parâmetros a seguir. Neste primeiro caso, um aumento do ganho de orientação k . A posição e orientação inicial dos agentes pode ser vista na Tabela (3.6).

- $k_v = 20$,
- $k = 20$,
- $\phi = 0$,
- $r_d = 200$,

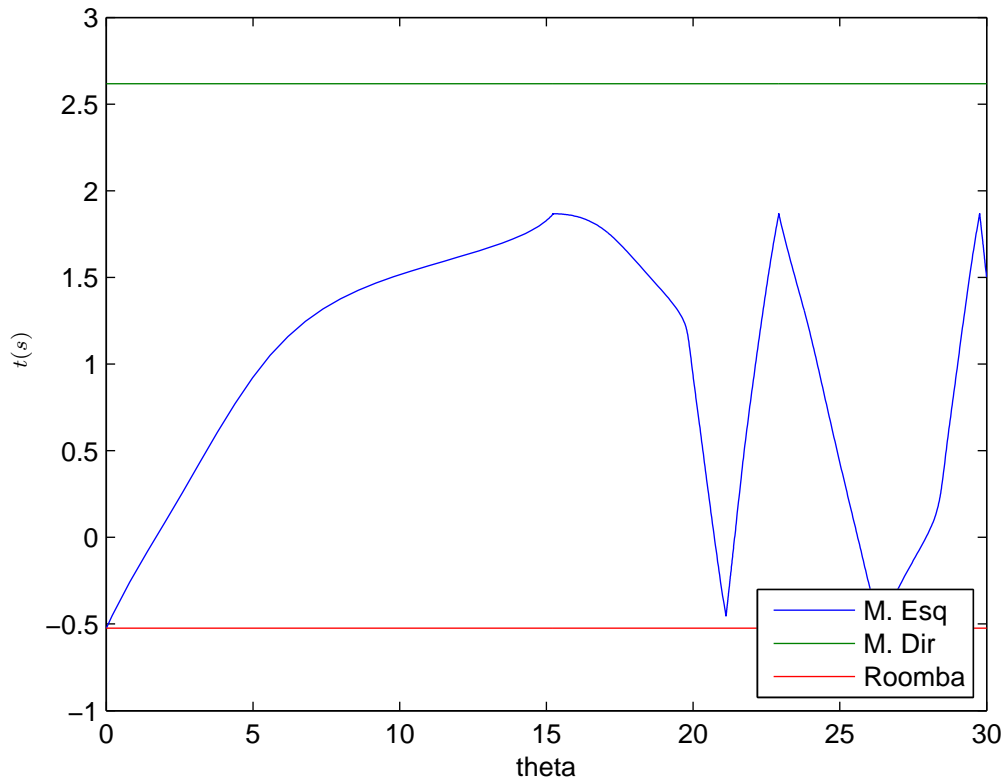


Figura 3.16: Orientação dos agentes - M. Esq. Caso 2

$$\bullet q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\pi \\ \frac{4}{2} \end{bmatrix}$$

$$\bullet q_{er} = \begin{bmatrix} \frac{\pi}{3} \\ -\pi \\ \frac{3}{3} \end{bmatrix}$$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} -0.4m \\ 1m \\ 0 \end{bmatrix}$$

A Figura (3.20) mostra que o tempo que o sistema leva para chegar a tal distância e minimizar a função potencial é de 35 segundos. A figura também mostra um pequeno sobressinal na distância entre os agentes 1 e 2, mas o gráfico de trajetória dos agentes (Figura 3.18) ainda tem curvatura suave, mesmo com o alto ganho de

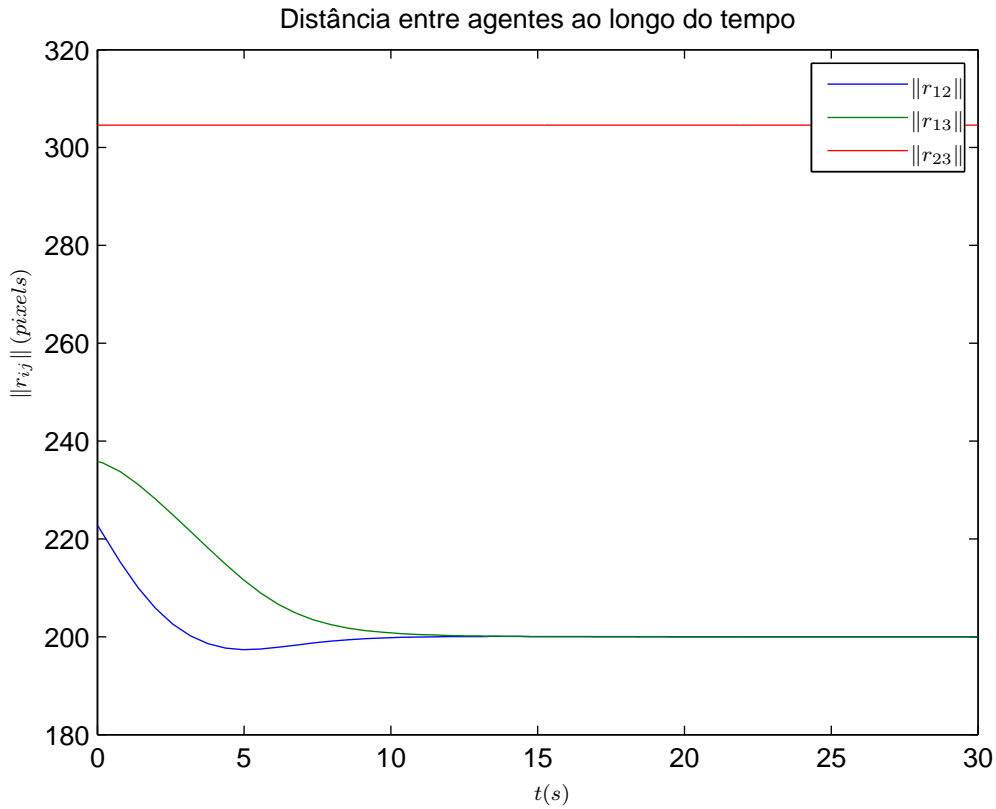


Figura 3.17: Distância entre agentes - M. Esq. Caso 2

Tabela 3.6: Parâmetros para controle de um agente - M. Dir. Caso 1

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$122.17pixels$	$344.90pixels$	137.96 pixels
$y(0)$	$-342.91pixels$	$-336.78pixels$	-206.94 pixels
$\theta(0)$	0	π	0

orientação, o que indica que esse pequeno sobressinal não chega a ser uma impossibilidade operacional para o motoman real. Na Figura (3.19) vê-se a orientação dos agentes. No próximo caso, veremos a resposta do sistema com um aumento no ganho da velocidade linear do controle.

Manipulador Direito - Caso 2

Nesta segunda simulação envolvendo o controle do manipulador direito, observa-se o que acontece com o sistema quando se aumenta o ganho de velocidade linear k_v . Os ganho de velocidade angular k foi reduzido para 0.5 e o ganho de velocidade liner

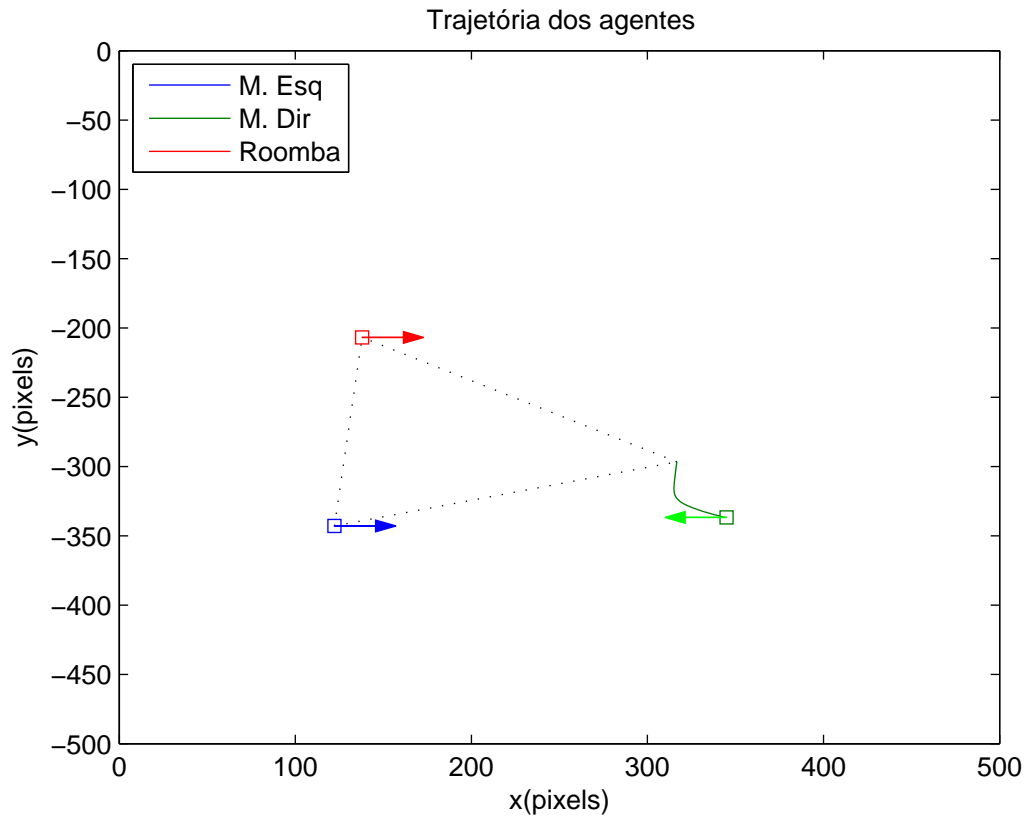


Figura 3.18: Trajetória dos agentes - M. Dir. Caso 1

k_v foi dobrado, para 40. O ângulo de rotação da câmera também foi alterado. Os novos valores iniciais de posição e orientação dos agentes encontram-se na Tabela (3.7).

- $k_v = 40$,
- $k = 0.5$,
- $\phi = \frac{\pi}{6}$,
- $r_d = 200$,

- $q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\pi \\ \frac{4}{2} \end{bmatrix}$

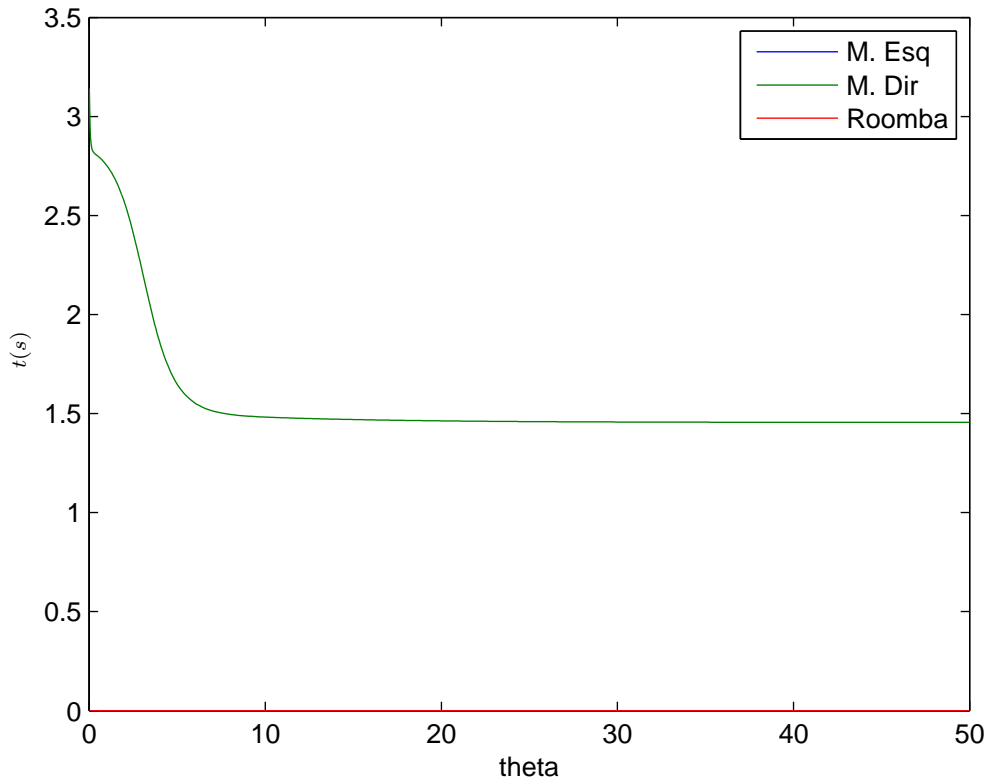


Figura 3.19: Orientação dos agentes - M. Dir. Caso 1

Tabela 3.7: Parâmetros para controle de um agente - M. Dir. Caso 2

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$-65.65pixels$	$130.30pixels$	$16.01 pixels$
$y(0)$	$-358.06pixels$	$-464, 11pixels$	$-248.20 pixels$
$\theta(0)$	$\frac{-\pi}{6}$	$\frac{5\pi}{6}$	$\frac{-2\pi}{e}$

$$\bullet q_{er} = \begin{bmatrix} \frac{\pi}{3} \\ \frac{-\pi}{3} \\ \frac{-\pi}{3} \\ \frac{\pi}{3} \end{bmatrix}$$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} -0.4m \\ 1m \\ \frac{\pi}{2} \end{bmatrix}$$

Comparando com a trajetória do caso 1 (Figura 3.18), é possível ver que a trajetória tem uma curvatura mais fechada, porém ainda realizável. O tempo que o sistema leva para chegar a formação desejada e reduzir todas as funções potenciais

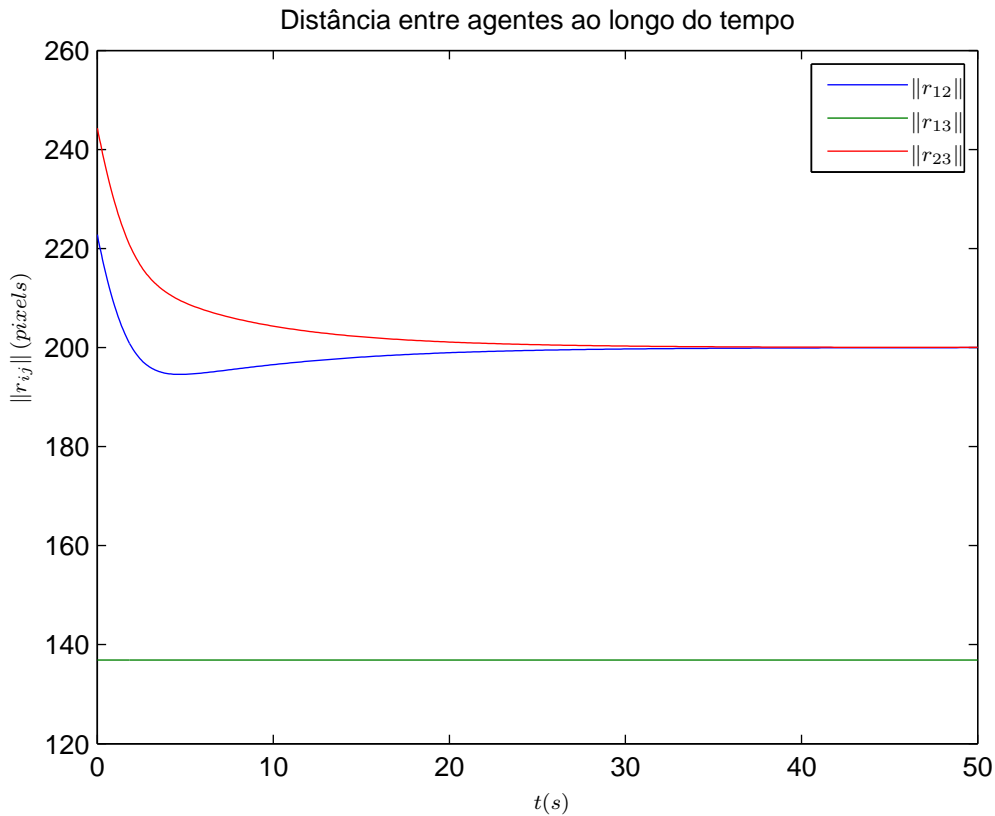


Figura 3.20: Distância entre agentes - M. Dir. Caso 1

ao seu mínimo diminuiu de forma considerável, como esperado, enquanto com a Figura (3.22) pode-se ver a diferença na orientação. A Figura (3.23) mostra que tal minimização ocorre em torno de $t = 19s$, pouco mais 50% dos 35 segundos da simulação anterior. Da mesma Figura (3.23), é possível ver que a amplitude do sobressinal não sofreu mudanças significativas, porém sua duração foi reduzida como a do sistema como um todo.

3.2.2 Dois agentes buscando a formação

Nesta seção, será estudado o caso em que dois dos agentes sofrem ação do campo potencial, o último agente estando imóvel. Neste caso, a formação alcançada já é um triângulo equilátero, a distância entre um agente que se move e um que não se move é regulada de forma unilateral, entre os agentes que se movem, entretanto, essa distância é regulada por ambos os agentes, influenciados indiretamente pelo movimento um do outro em um instante anterior. Serão realizadas duas simulações para cada agente, com diferentes condições iniciais .

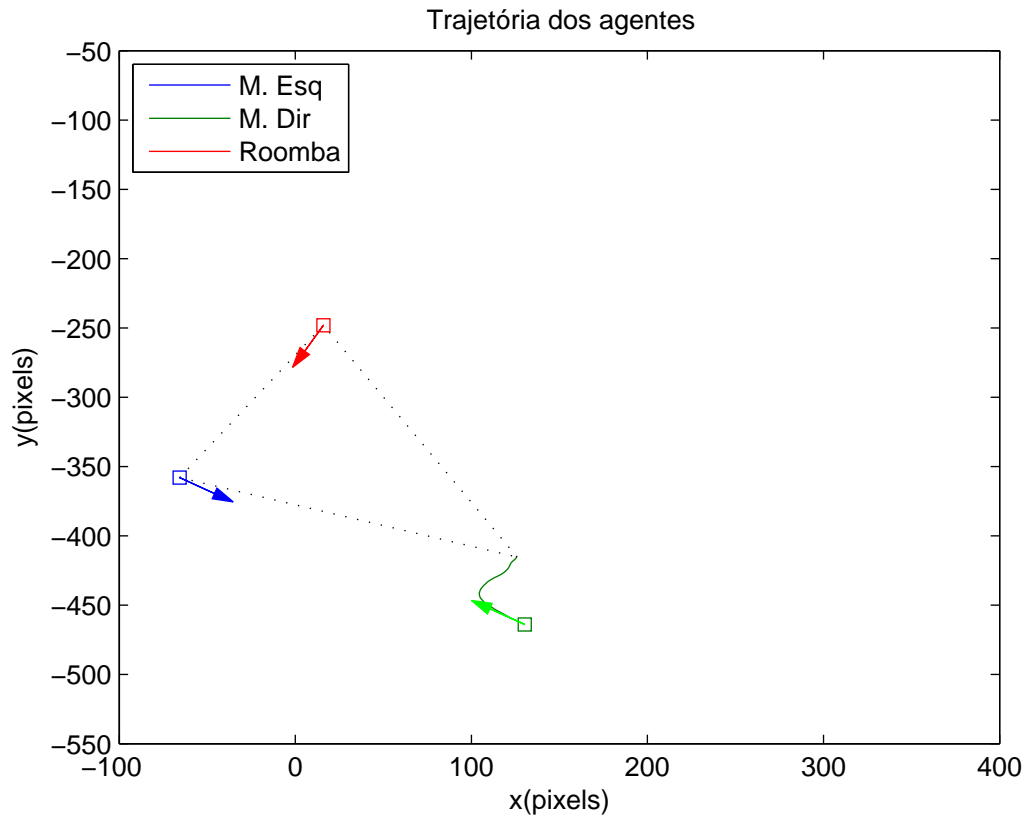


Figura 3.21: Trajetória dos agentes - M. Dir. Caso 2

Manipuladores Esquerdo e Direito - Caso 1

Foi escolhido um valor arbitrariamente baixo para o parâmetro k de forma a suavizar a trajetória. O parâmetro k_v foi escolhido visando um baixo tempo de resposta do sistema, mas sem gerar grandes sobressinalis no processo. O parâmetro ϕ da câmera foi zerado para a primeira análise, e será alterado no caso 2. A distância desejada r_d foi escolhida de acordo com o planejamento de trajetória adequado. Os parâmetros iniciais dos agentes como capturados pela câmera estão especificados na Tabela (3.8).

- $k_v = 20$,
- $k = 0.5$,
- $\phi = 0$,
- $r_d = 200$,

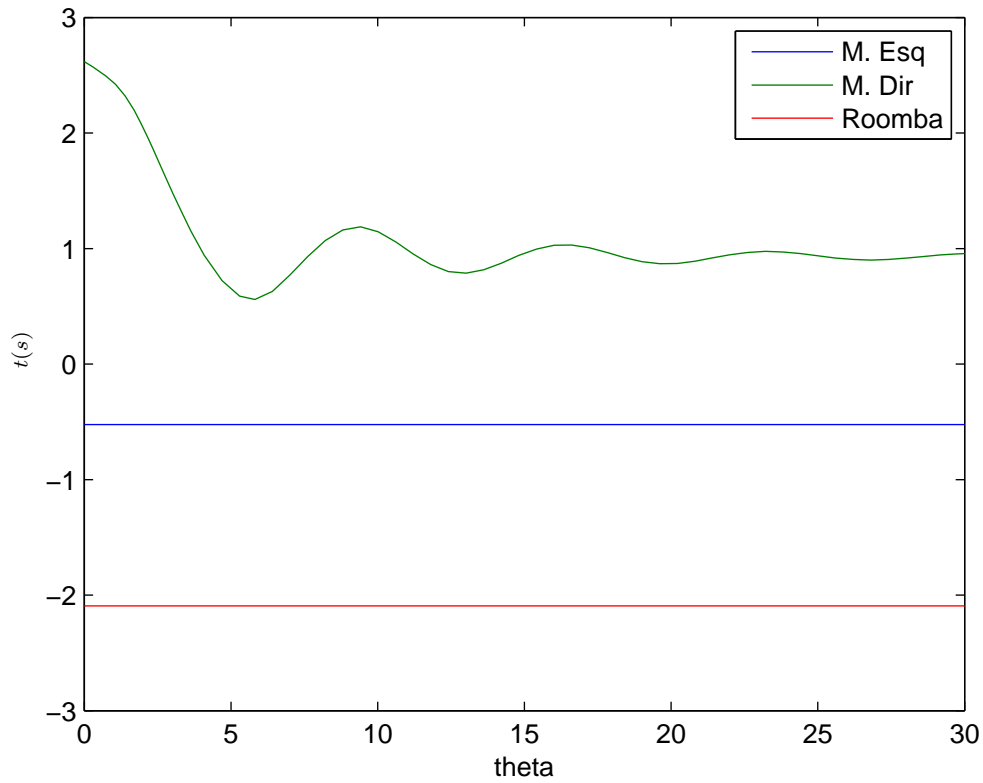


Figura 3.22: Orientação dos agentes - M. Dir. Caso 2

$$\bullet q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}$$

$$\bullet q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}$$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} 0.1m \\ 1.3m \\ -\frac{\pi}{2} \end{bmatrix}$$

A Figura (3.24) mostra a trajetória dos manipuladores, no plano da imagem, até alcançar a formação, enquanto a Figura (3.25) mostra a orientação dos agentes. Observe que os movimentos foram realizados de forma não-holonômica, de acordo com as restrições impostas em ambos os manipuladores. A Figura (3.26) mostra um

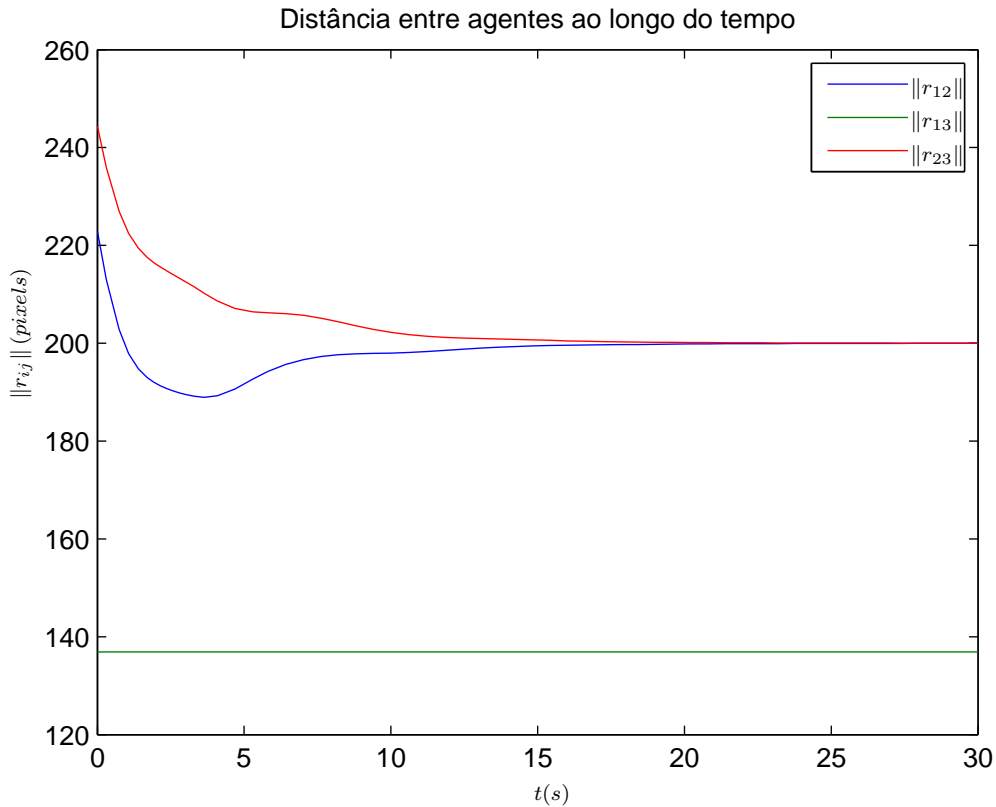


Figura 3.23: Distância entre agentes - M. Dir. Caso 2

Tabela 3.8: Parâmetros para controle de 2 agentes - M.Esq./M.Dir. Caso 1

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$150.02pixels$	$429.67pixels$	310.41 pixels
$y(0)$	$-339.27pixels$	$-342.91pixels$	-103.47 pixels
$\theta(0)$	0	π	$\frac{-\pi}{2}$

sobressinal na variação da distância entre os efetuadores do manipulador esquerdo e do manipulador direito, visto na Figura (3.24) como curvas conforme os efetuadores dos manipuladores se aproximam. O sistema leva 24 segundos para ir de suas condições iniciais até a formação minimiza a função potencial saturada.

Manipuladores Esquerdo e Direito - Caso 2

O parâmetros k_v foi dobrado em relação ao escolhido no caso 1, e o ganho de orientação k foi multiplicado por 4. Novamente, adicionou-se um ângulo ϕ de rotação para a câmera, para verificação dos seus efeitos sobre o sistema, quando dois agentes

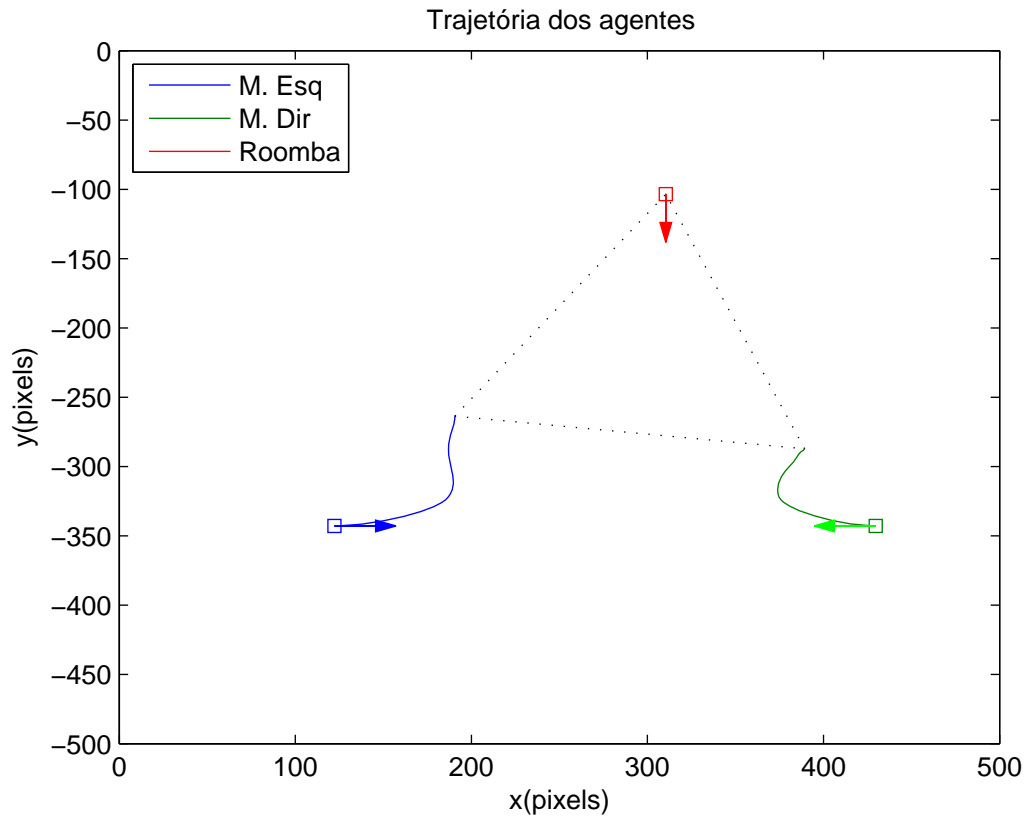


Figura 3.24: Trajetória dos agentes - M.Esq./M.Dir. Caso 1

são controlados . A distância desejada r_d foi mantida, ainda gerando uma trajetória adequada. Os parâmetros iniciais dos agentes estão especificados na Tabela (3.9).

- $k_v = 40,$

- $k = 2,$

- $\phi = \frac{\pi}{6},$

- $r_d = 200,$

- $q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ \frac{4}{- \pi} \\ \frac{4}{- \pi} \\ \frac{2}{2} \end{bmatrix}$

- $q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ \frac{4}{- \pi} \\ \frac{4}{- \pi} \\ \frac{2}{2} \end{bmatrix}$

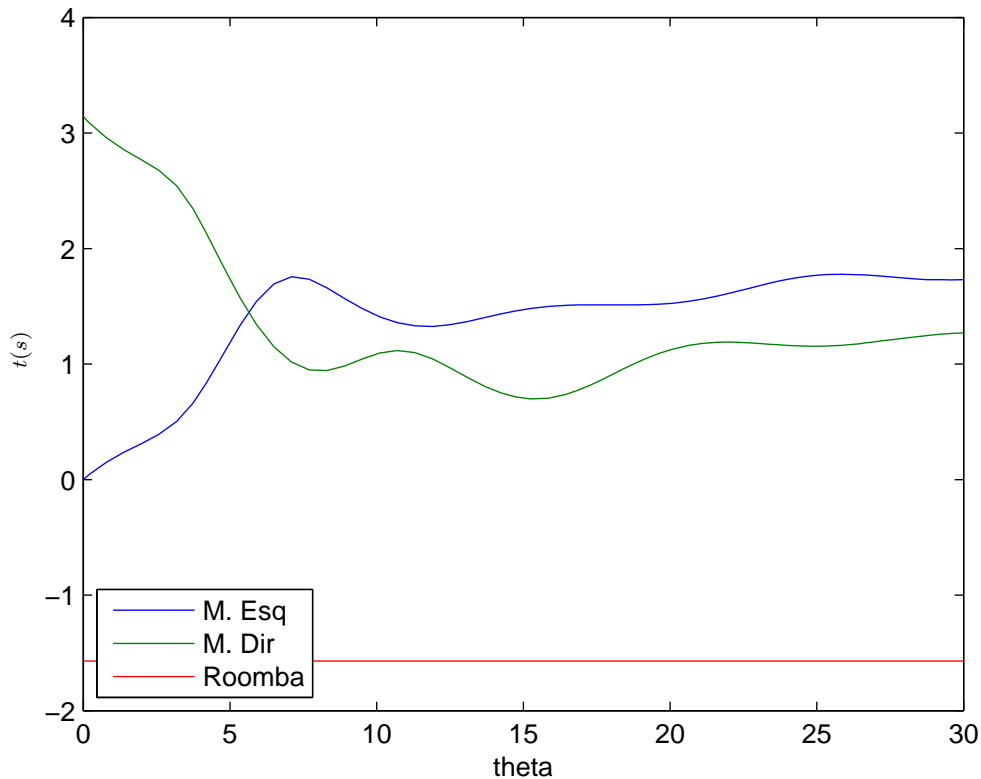


Figura 3.25: Orientação dos agentes - M. Esq./Dir. Caso 1

Tabela 3.9: Parâmetros para controle de 2 agentes - M.Esq./M.Dir. Caso 2

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$-44.57pixels$	$200.65pixels$	$217.09 pixels$
$y(0)$	$-365.56pixels$	$-511.81pixels$	$-244.81 pixels$
$\theta(0)$	$\frac{-\pi}{6}$	$\frac{5\pi}{6}$	$\frac{-2\pi}{3}$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} 0.1m \\ 1.3m \\ \frac{-\pi}{2} \end{bmatrix}$$

Observa-se pela Figura (3.27) que a trajetória continuou curva, durante o tempo de sobressinal visto na Figura (3.29). Pela mesma Figura (3.29), vemos que a amplitude do sobressinal se mantém, como esperado pelos resultados anteriores, o mesmo comportamento pode ser observado na orientação dos agentes, como visto na Figura (3.28). Assim como visto no caso para um único agente, o aumento do ganho k_v reduz consideravelmente o tempo que o sistema leva para minimizar sua função

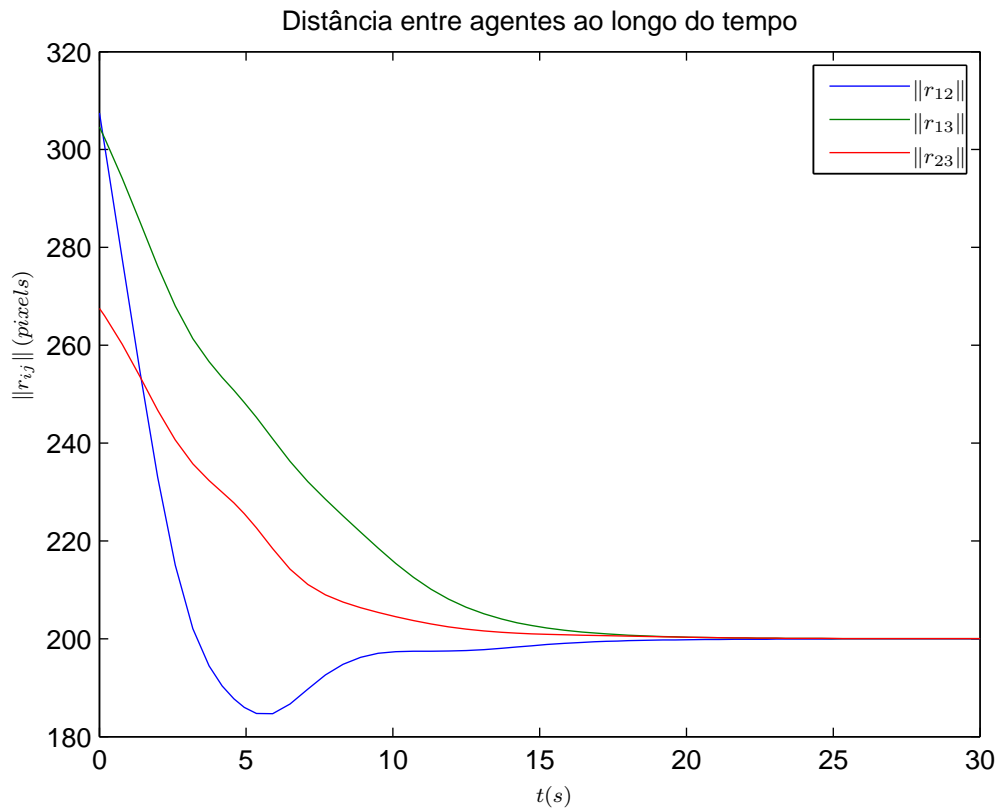


Figura 3.26: Distância entre agentes - M.Esq./M.Dir. Caso 1

potencial, neste caso, 11 segundos.

Manipulador Esquerdo e Roomba - Caso 1

Para o caso 1, foram escolhidos os mesmos parâmetros k_v , k , ϕ e r_d do primeiro caso com os manipuladores esquerdo e direito. Os parâmetros iniciais dos agentes estão especificados na Tabela (3.10).

- $k_v = 20$,
- $k = 0.5$,
- $\phi = 0$,
- $r_d = 200$,

- $q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{4} \\ \frac{\pi}{2} \end{bmatrix}$

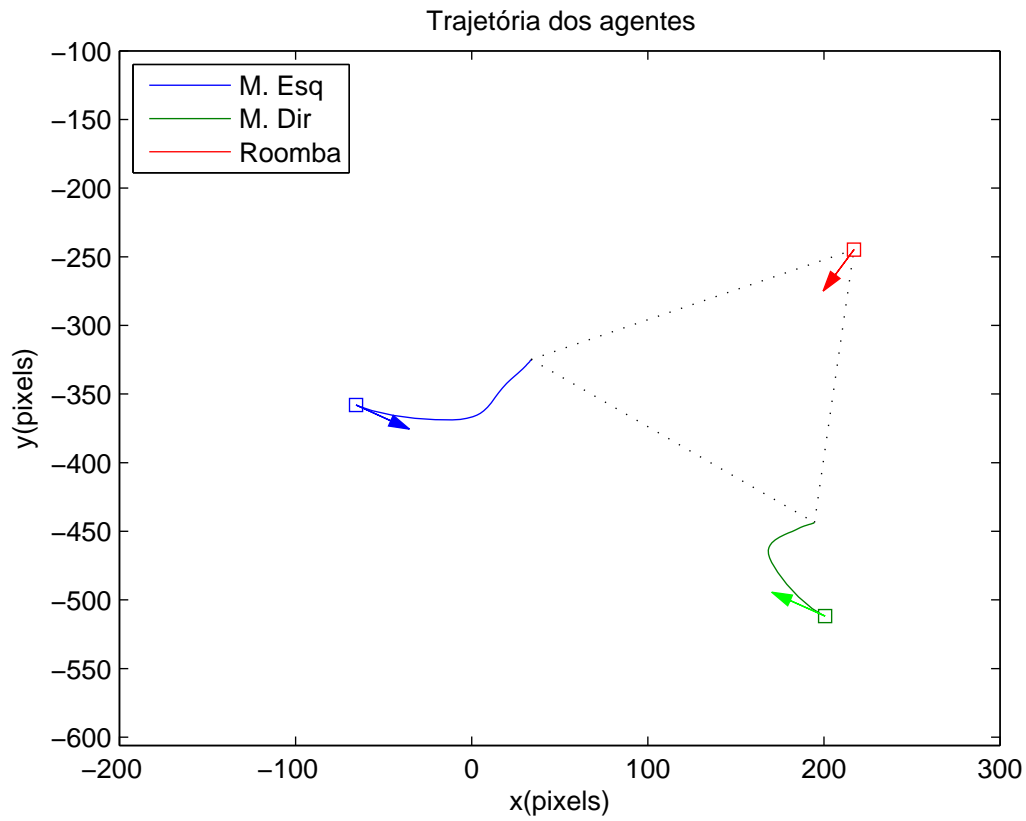


Figura 3.27: Trajetória dos agentes - M.Esq./M.Dir. Caso 2

Tabela 3.10: Parâmetros para controle de 2 agentes - M.Esq./Rob. Caso 1

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$122.17pixels$	$429.67pixels$	$275.92 pixels$
$y(0)$	$-342.91pixels$	$-342.91pixels$	$-103.47 pixels$
$\theta(0)$	0	π	0

$$\bullet q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\pi \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{2} \end{bmatrix}$$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.3m \\ 0 \end{bmatrix}$$

A Figura (3.30) mostra a trajetória dos manipulador esquerdo e do roomba, no plano da imagem, enquanto a Figura (3.31) mostra sua orientação. Observe que os

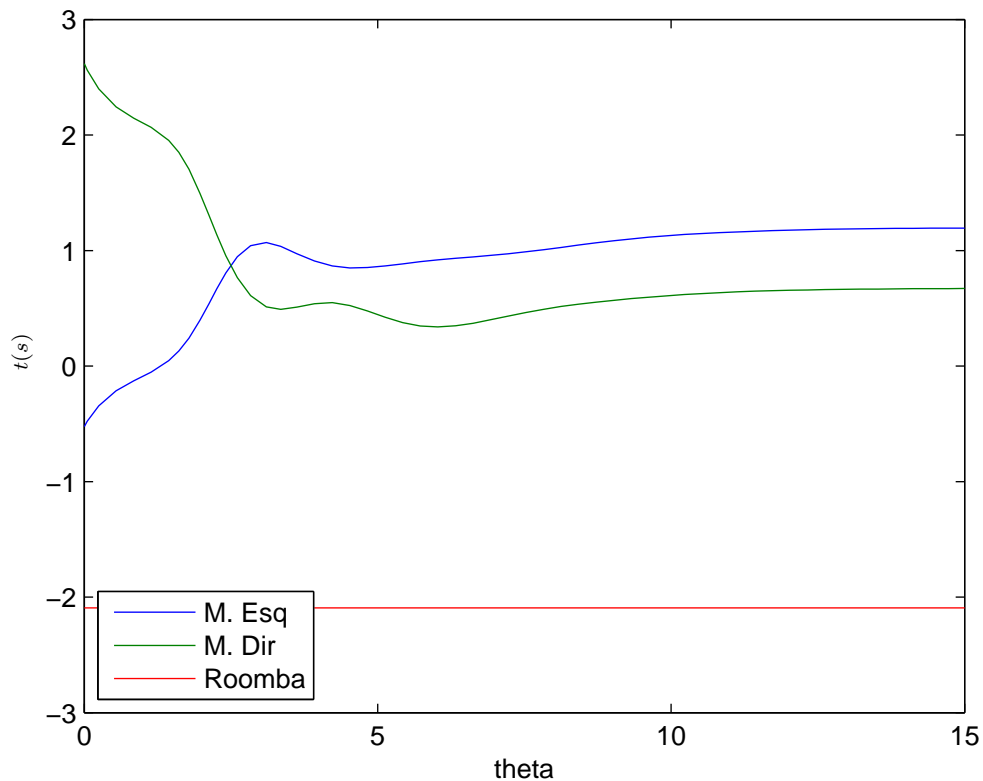


Figura 3.28: Orientação dos agentes - M. Esq./Dir. Caso 2

dois agentes em questão se aproximam além da distância desejada, mas corrigem esse erro de maneira suave, como mostra a Figura (3.32). A Figura (3.26) também mostra como sobressinal só ocorre na distância entre os dois agentes que se controlados. O sistema como um todo leva 21 segundos para ir de suas condições iniciais até a formação minimiza a função potencial saturada.

Manipulador Esquerdo e Roomba - Caso 2

Nesta simulação, será alterado somente o valor do ângulo de rotação da câmera, ϕ , com o intuito de comparar o novo resultado com o da simulação anterior, observando se há alteração na efetividade do controle, tendo como parâmetro o tempo para minimizar a função potencial e o sobressinal do sistema. A posição e orientação inicial do sistema, no plano da imagem, podem ser vistas na Tabela (3.11).

- $k_v = 20$,
- $k = 0.5$,

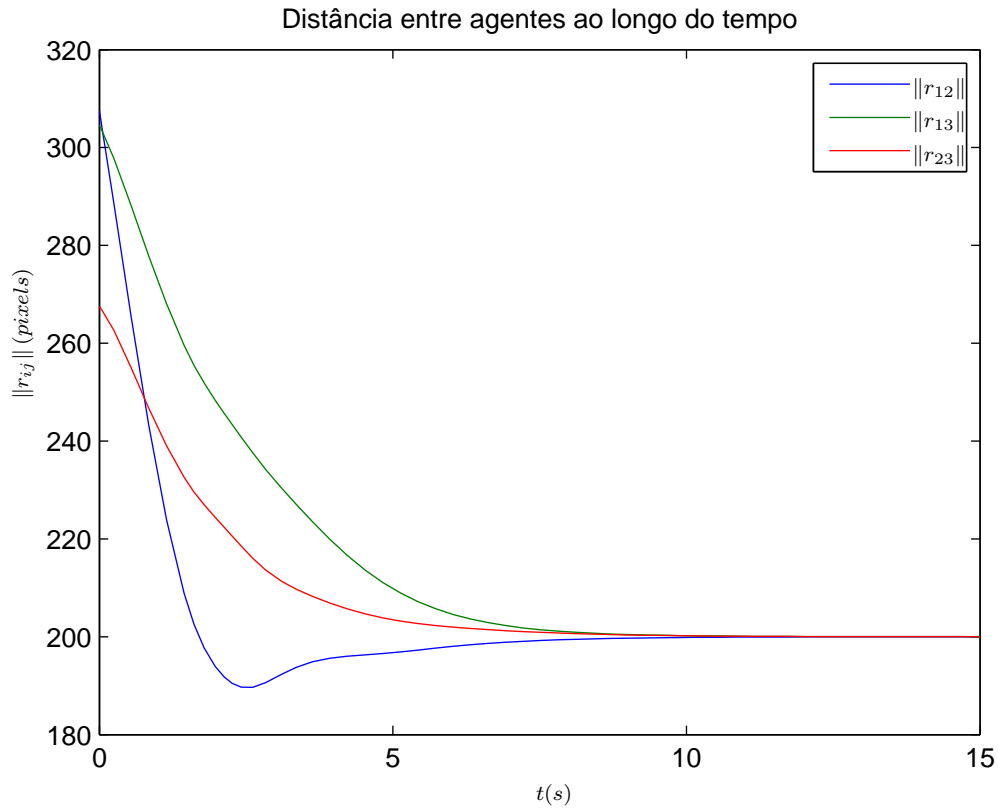


Figura 3.29: Distância entre agentes - M.Esq./M.Dir. Caso 2

- $\phi = \frac{\pi}{6}$,

- $r_d = 200$,

- $q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\pi \\ \frac{4}{2} \end{bmatrix}$

- $q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\pi \\ \frac{4}{2} \end{bmatrix}$

- $\begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} 0.1m \\ 1.3m \\ 0 \end{bmatrix}$

A Figura (3.33) mostra o resultado mais notório, que é a diferença na trajetória após a rotação. Entretanto, a Figura (3.35), quando comparada à Figura (3.32),

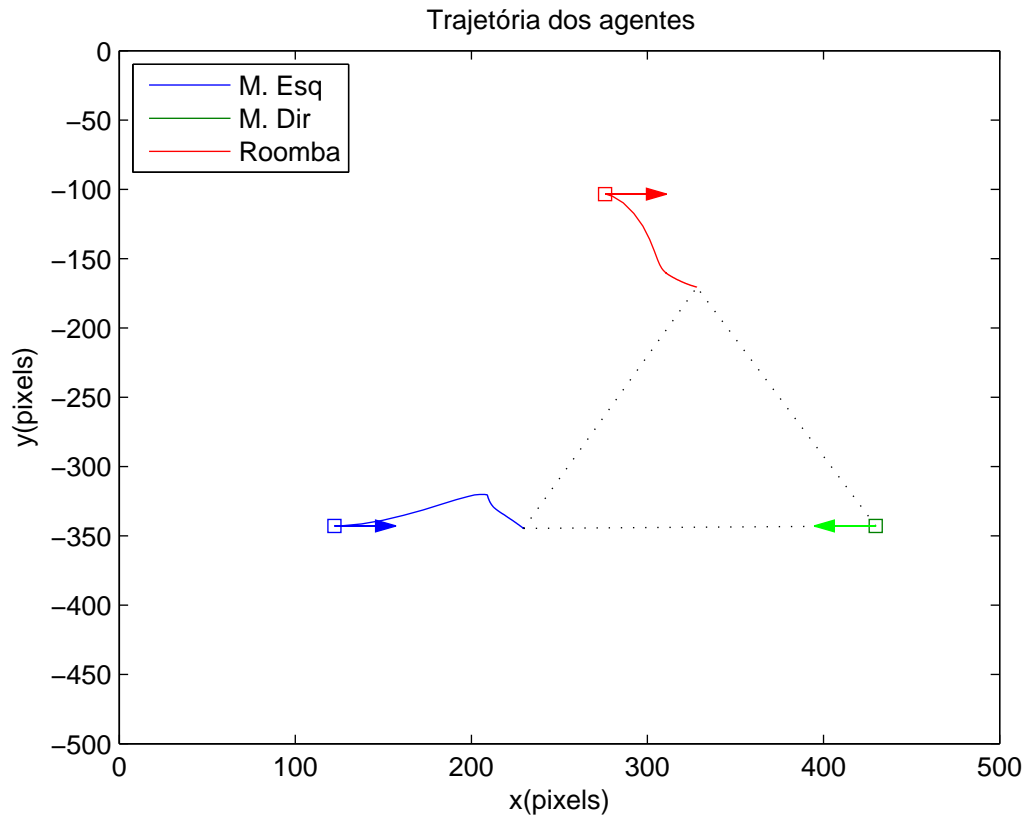


Figura 3.30: Trajetória dos agentes - M.Esq./Rob. Caso 1

Tabela 3.11: Parâmetros para controle de 2 agentes - M.Esq./Rob. Caso 2

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$-65.65pixels$	$200.65pixels$	$187.22 pixels$
$y(0)$	$-358.06pixels$	$-511.81pixels$	$-227.57 pixels$
$\theta(0)$	$\frac{-\pi}{6}$	$\frac{5\pi}{6}$	$\frac{-\pi}{6}$

revela que não existe diferença no tempo de minimização do campo potencial, nem mesmo alterações no sobressinal. A Figura (3.34) mostra que não existe mudança na variação das orientações ao longo do tempo, quando comparada à Figura (3.31), portanto a rotação da câmera não altera a ação da função potencial, como esperado com o resultado das simulações anteriores.

Manipulador Direito e Roomba - Caso 1

Para esse conjunto de agentes, o objetivo será testar um incremento alto no ganho k com uma redução no ganho k_v . Para tal, primeiro faz-se essa simulação base, que

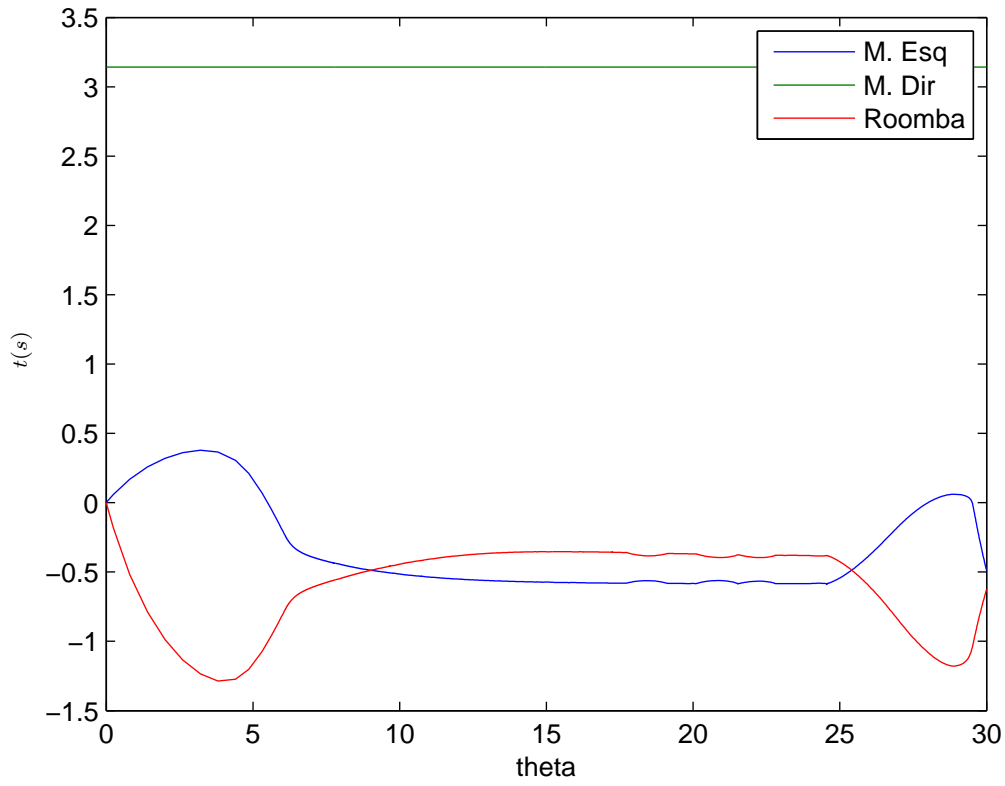


Figura 3.31: Orientação dos agentes - M. Esq./Roomba Caso 1

servirá de parâmetro de comparação. Os demais parâmetros estão listados abaixo e as condições iniciais dos agentes podem ser encontradas na Tabela (3.13).

- $k_v = 20,$

- $k = 0.5,$

- $\phi = 0,$

- $r_d = 200,$

- $q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{2} \end{bmatrix}$

- $q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{2} \end{bmatrix}$

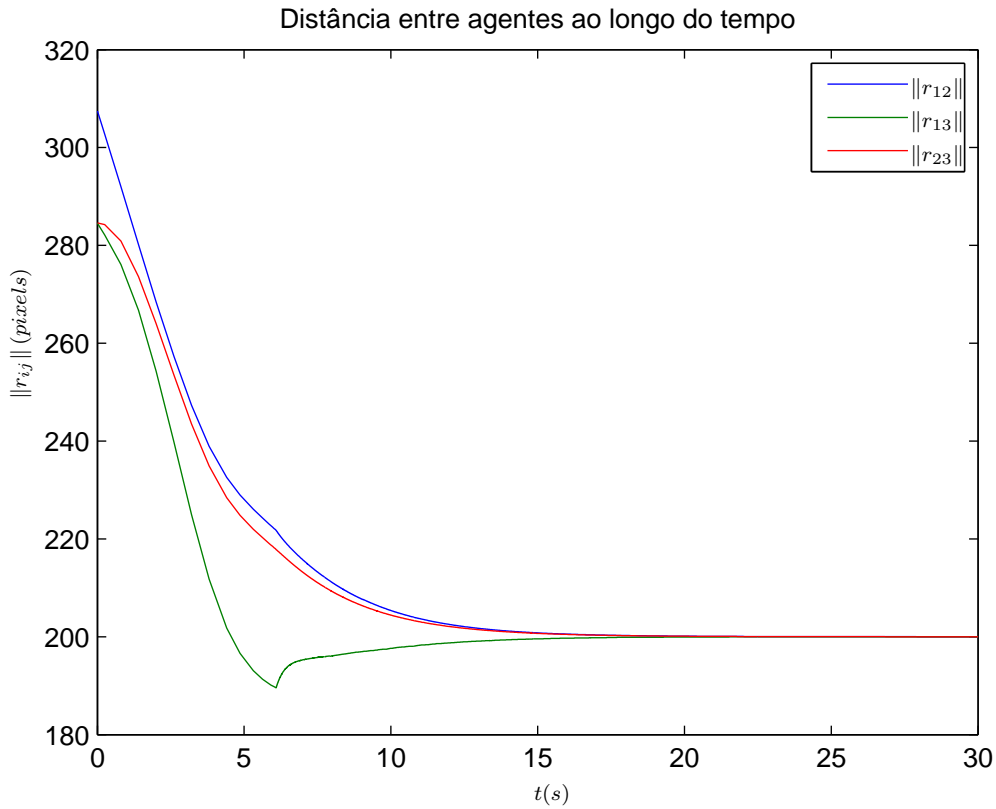


Figura 3.32: Distância entre agentes - M.Esq./Rob. Caso 1

Tabela 3.12: Parâmetros para controle de 2 agentes - M.Dir./Rob. Caso 1

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$122.17pixels$	$429.67pixels$	$241.43 pixels$
$y(0)$	$-342.91pixels$	$-342.91pixels$	$-137.96 pixels$
$\theta(0)$	0	π	$\frac{-\pi}{2}$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} -0.1m \\ 1.2m \\ -\frac{\pi}{2} \end{bmatrix}$$

Na Figura (3.36), vê-se que o o roomba realizada uma curva brusca, seguindo a tendência dos agentes controlados se aproximarem mais rápido um do outro que do terceiro agente, o que gera um sobressinal na distância entre eles. A curva que o manipulador direito realiza é mais suave, pela distância inicial entre o mesmo e o manipulador esquerdo ser maior que a do roomba e o mesmo manipulador, o que pode ser confirmado com a orientação dos agentes, visto na Figura (3.37). Da Figura

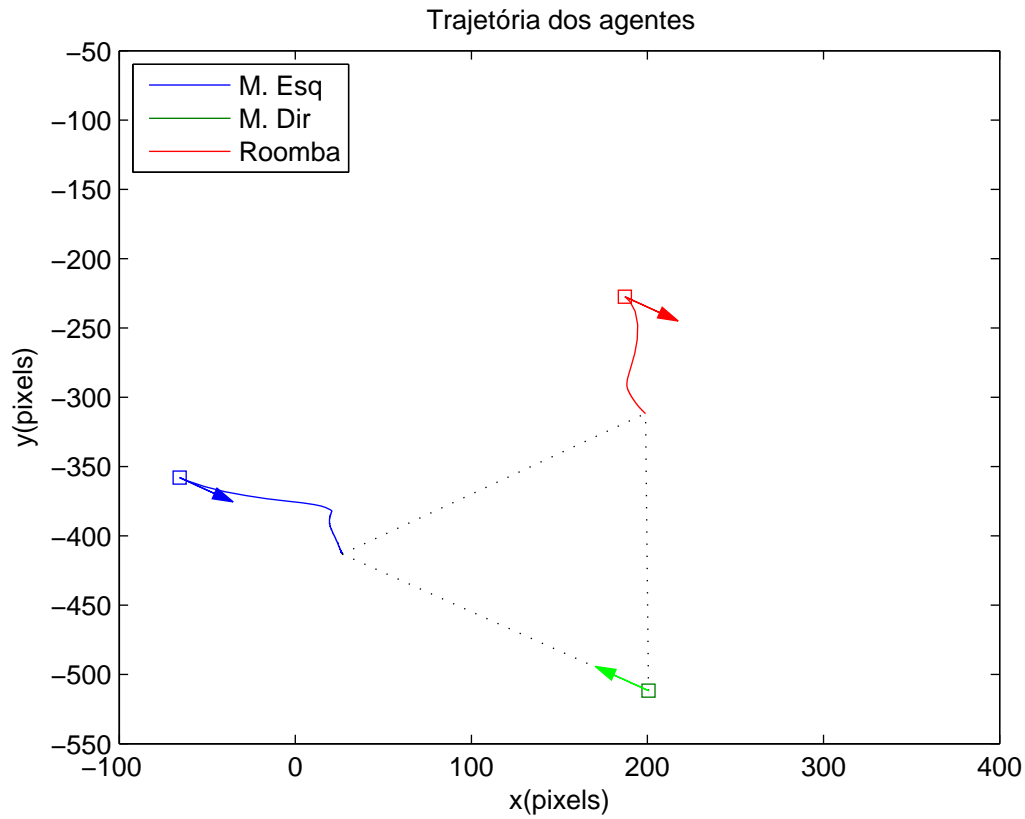


Figura 3.33: Trajetória dos agentes, com seu $\theta(0)$ - M.Esq./Rob. Caso 2

(3.38), também nota-se que o tempo que o sistema leva para minimizar a sua função potencial é de 19 segundos.

Manipulador Direito e Roomba - Caso 2

Nesta simulação, o ganho k_v foi mantido, e o ganho k multiplicado por 40, em relação a simulação anterior. Não se incluiu rotação da câmera, neste caso, para uma melhor análise das diferenças entre trajetórias e das resposta do sistema. Todos os demais parâmetros foram mantidos idênticos ao caso anterior, as condições iniciais do sistema disponíveis na Tabela (3.13).

- $k_v = 20$,
- $k = 20$,
- $\phi = 0$,
- $r_d = 200$,

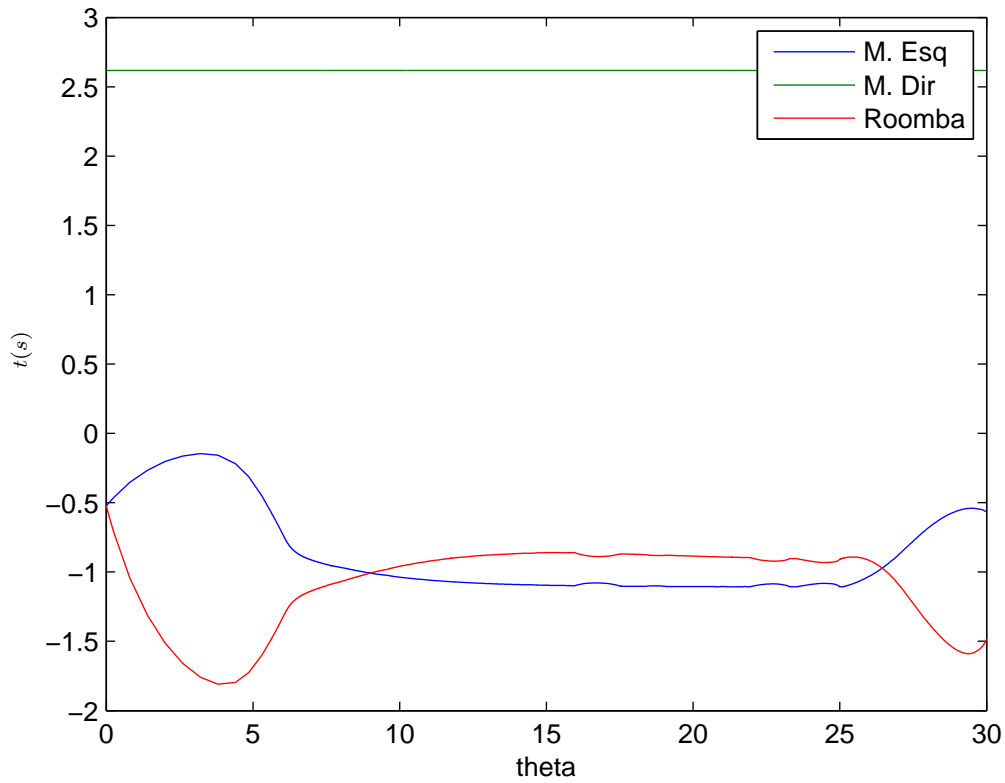


Figura 3.34: Orientação dos agentes - M. Esq./Roomba Caso 2

$$\bullet q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{4} \end{bmatrix}$$

$$\bullet q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{4} \end{bmatrix}$$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} -0.1m \\ 1.2m \\ -\frac{\pi}{2} \end{bmatrix}$$

Nota-se, pela Figura (3.39), que a trajetória é mais aberta em relação ao caso anterior (Figura 3.36), já que a orientação dos agentes varia mais rápida na intenção de chegar na orientação desejada. Pelo mesmo motivo, a correção de orientação quando os agentes 2 e 3 se aproximam ocorre de forma acelerada, o que diminui o

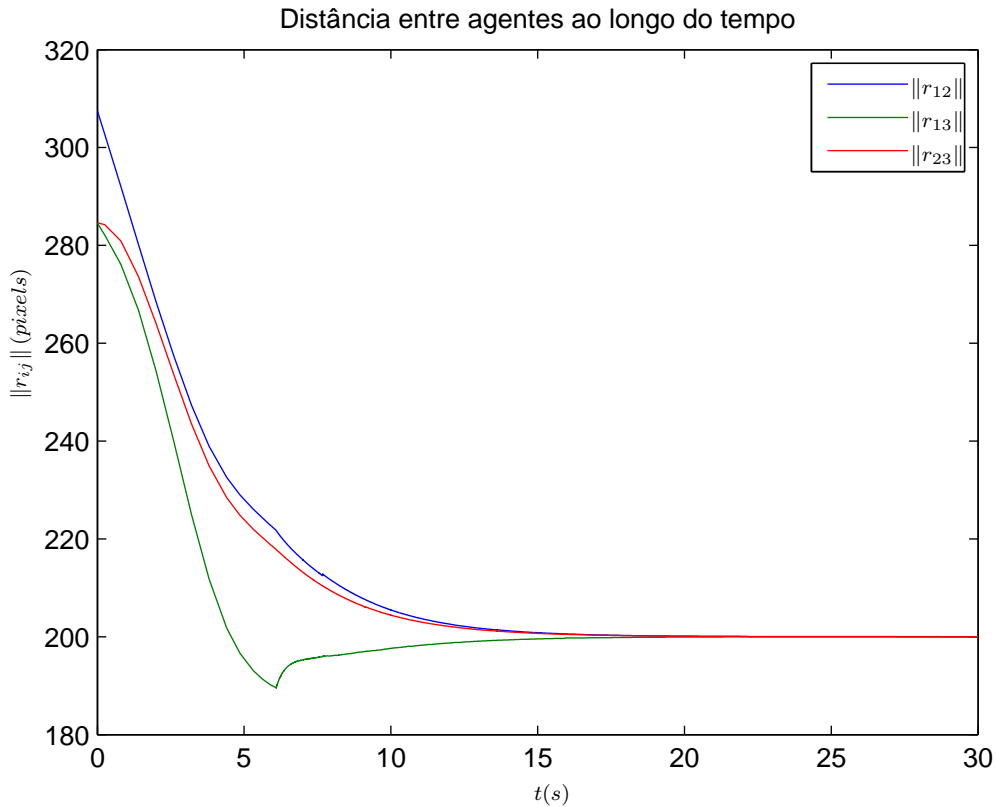


Figura 3.35: Distância entre agentes - M.Esq./Rob. Caso 2

Tabela 3.13: Parâmetros para controle de 2 agentes - M.Dir./Rob. Caso 2

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$122.17pixels$	$429.67pixels$	241.43 pixels
$y(0)$	$-342.91pixels$	$-342.91pixels$	-137.96 pixels
$\theta(0)$	0	π	$\frac{-\pi}{2}$

sobressinal, visto na Figura (3.41) e que pode ser confirmado com a Figura (3.40). A Figura (3.41) também mostra que o tempo levado para percorrer essa mesma trajetória, entretanto, não foi diferente do caso anterior, ainda precisando de 19 segundos para minimizar a função potencial. É válido notar que apesar das trajetórias diferentes, a posição final dos agentes ainda foi a mesma.

3.2.3 Três agentes buscando a formação

Nesta seção, será finalmente simulado o caso em que todos os agentes sofrem ação do campo potencial. A formação alcançada continua sendo um triângulo equilátero,

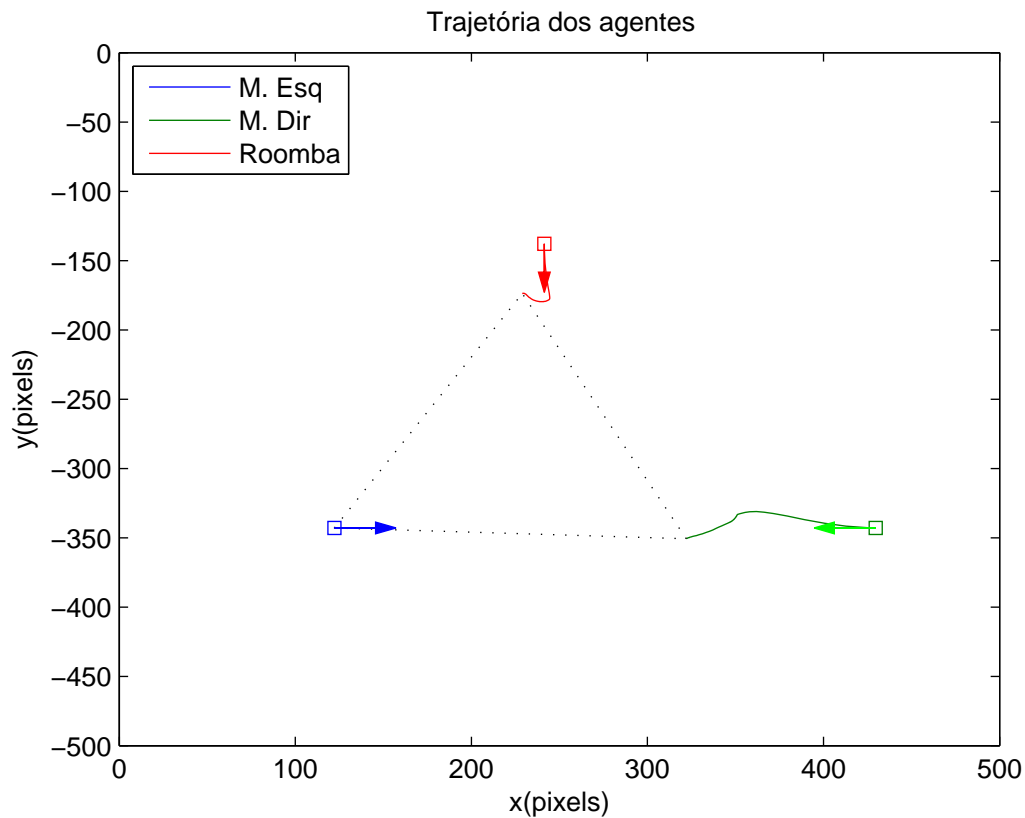


Figura 3.36: Trajetória dos agentes - M.Dir./Rob. Caso 1

e cada par de agentes influencia na regulação da distância entre os mesmos, com contribuições de mesmo módulo e sentidos opostos, entre esse par.

Caso 1

Os parâmetros k_v e k foram sintonizados por tentativa, com base nas simulações anteriores. Não incluiu-se a rotação ϕ da câmera neste primeiro caso. A distância desejada r_d também foi escolhida com base nas simulações anteriores. Os parâmetros iniciais dos agentes como capturados pela câmera, por sua vez, estão especificados na Tabela (3.14).

- $k_v = 20$,
- $k = 0.5$,
- $\phi = 0$,
- $r_d = 200$,

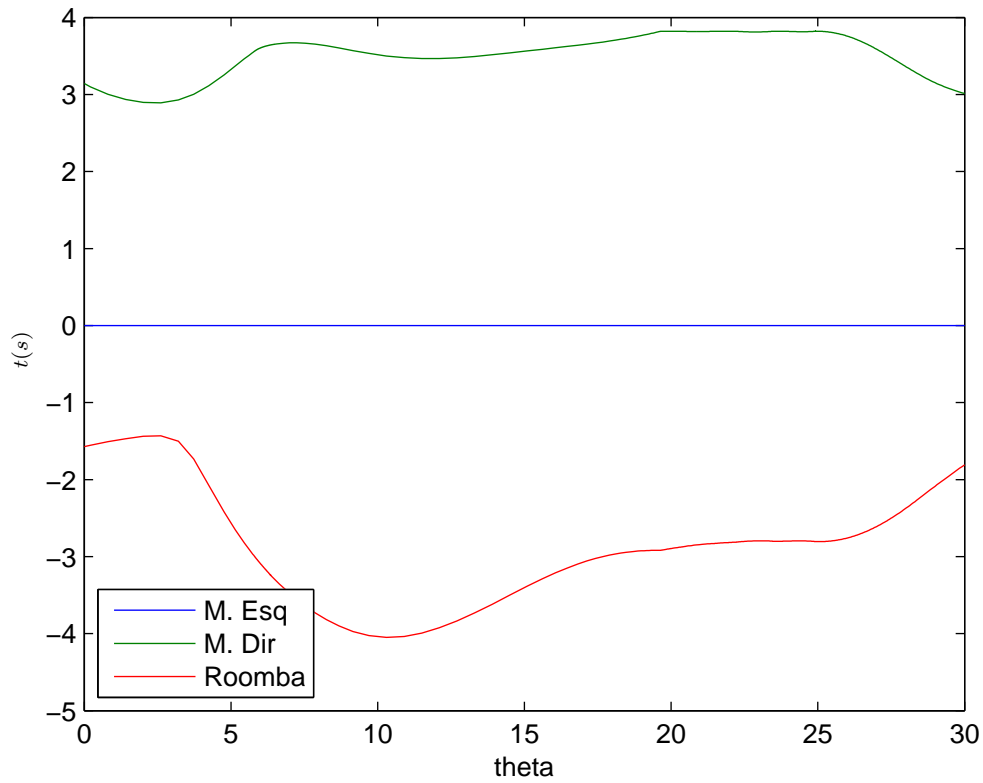


Figura 3.37: Orientação dos agentes - M. Dir./Roomba Caso 1

$$\bullet q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{4} \end{bmatrix}$$

$$\bullet q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{2} \\ \frac{\pi}{4} \end{bmatrix}$$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} 0.1m \\ 1.3m \\ \frac{\pi}{2} \end{bmatrix}$$

Um primeiro resultado, esperado, que pode ser visto na Figura (3.42) é o fato de que cada agente percorre, individualmente, uma trajetória menor que em casos anteriores, para condições iniciais semelhantes. Há sobressinal visível na distância entre o agente 2 e o agente 3 e um sobressinal suave entre os agentes 1 e 2, como

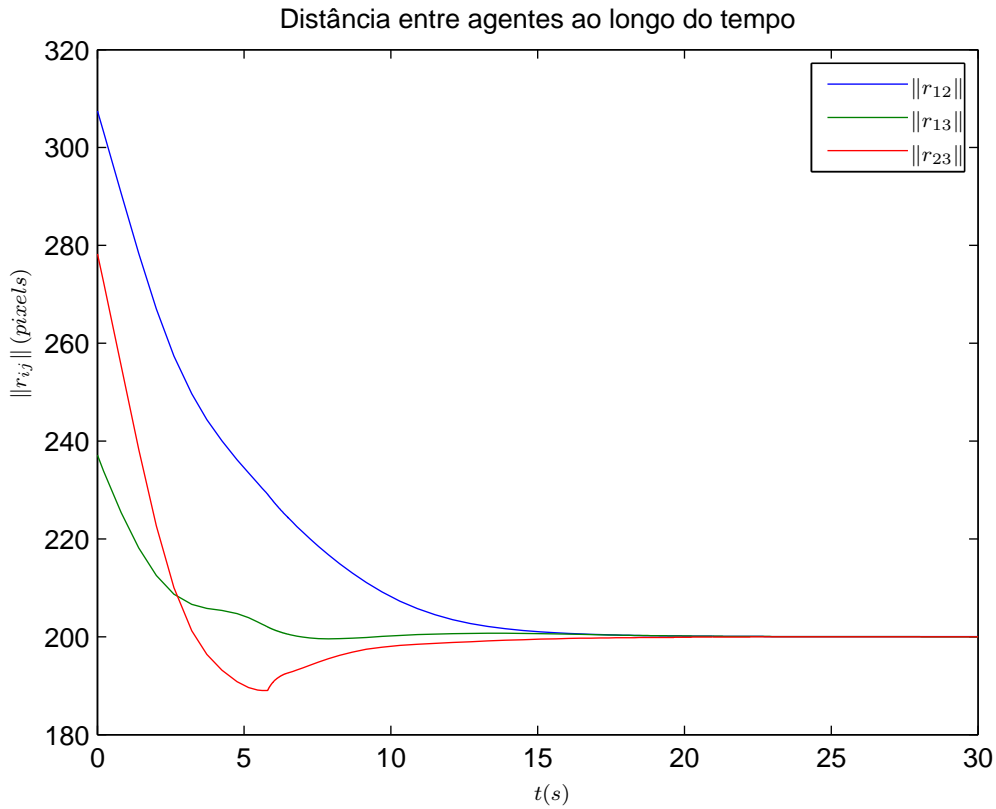


Figura 3.38: Distância entre agentes - M.Dir./Rob. Caso 1

Tabela 3.14: Parâmetros iniciais para controle de 3 agentes - Caso 1

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$122.17pixels$	$429.67pixels$	310.41 pixels
$y(0)$	$-342.91pixels$	$-342.91pixels$	-103.47 pixels
$\theta(0)$	0	π	$\frac{-\pi}{2}$

mostra a Figura (3.44), a Figura (3.42) confirma a suavidade da trajetória. O tempo que o sistema leva para conjunto de estados que minimiza a função potencial, pode ser visto na mesma figura, vale $t = 15s$. A orientação dos agentes durante a trajetória pode ser vista na figura (3.43)

Caso 2

Nesta segunda simulação, serão mantidos os valores de k_v , k e r_d . Diferente do caso anterior, neste haverá uma rotação ϕ da câmera, e a posição inicial do roomba no plano real foi levemente ajustada, uma alteração de $-0.1m$ no eixo y . As condições

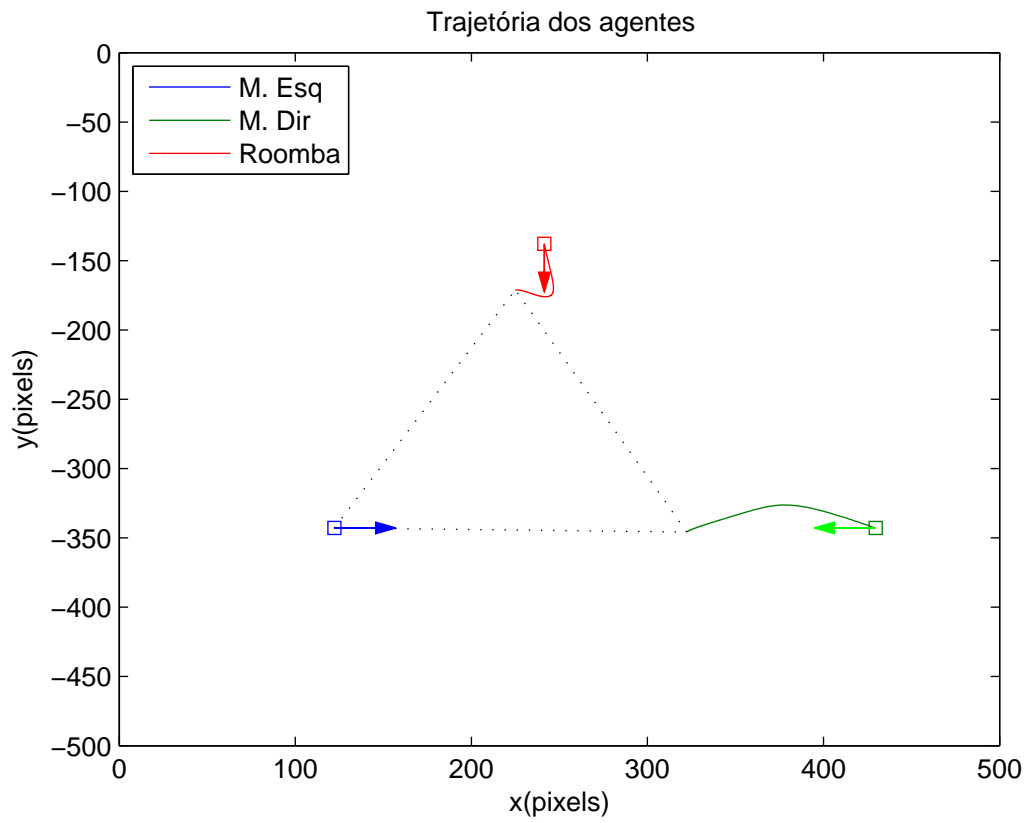


Figura 3.39: Trajetória dos agentes - M.Dir./Rob. Caso 2

iniciais dos agentes, levando em consideração essas duas mudanças, encontram-se na Tabela (3.15).

- $k_v = 20$,

- $k = 0.5$,

- $\phi = \frac{\pi}{6}$,

- $r_d = 200$,

- $q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \end{bmatrix}$

- $q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \\ \frac{4}{-2\pi} \end{bmatrix}$

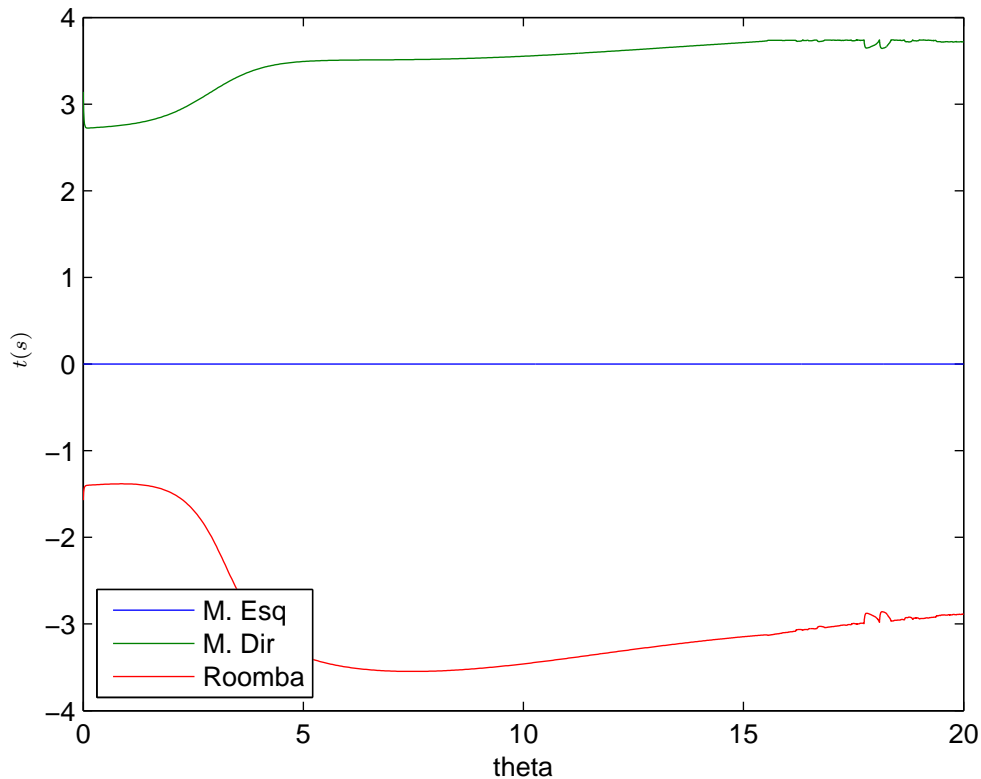


Figura 3.40: Orientação dos agentes - M. Dir./Roomba Caso 2

Tabela 3.15: Parâmetros iniciais para controle de 3 agentes - Caso 2

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$-65.65pixels$	$200.65pixels$	$187.96pixels$
$y(0)$	$-358.06pixels$	$-511.81pixels$	$-292.57pixels$
$\theta(0)$	$\frac{-\pi}{6}$	$\frac{5\pi}{6}$	$\frac{-\pi}{6}$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} 0.1m \\ 1.2m \\ \frac{\pi}{2} \end{bmatrix}$$

A Figura (3.45) mostra a trajetória dos agentes enquanto a Figura (3.46) mostra a orientação dos agentes durante esta trajetória. Observa-se que, nessas condições, há um sobressinal entre o roomba e o manipulador direito, mas a Figura (3.47) mostra que a distância entre os agentes não passa novamente da distância desejada, como ocorreu no caso anterior. A mesma figura também mostra há um pequena redução no tempo que o sistema leva para regular suas distâncias, descendo dos 15

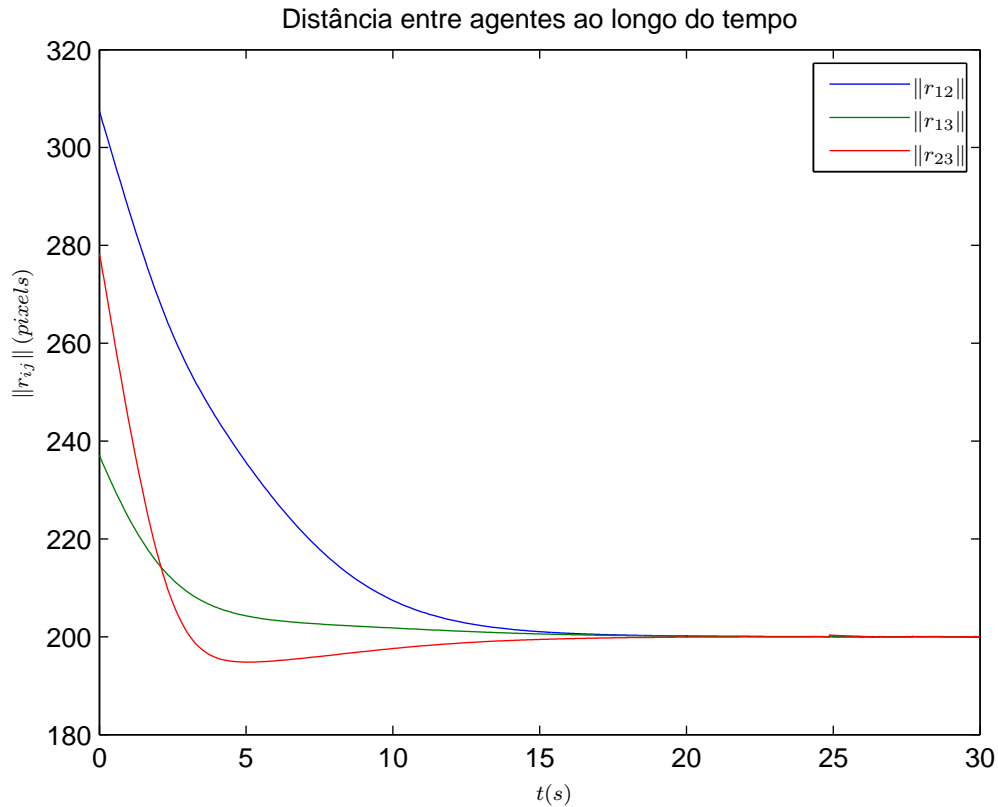


Figura 3.41: Distância entre agentes - M.Dir./Rob. Caso 2

segundos da simulação anterior para 11 segundos nesta simulação.

Caso 3

Nesta terceira simulação, escolhe-se condições iniciais incomuns ao espaço do trabalho, um pouco mais adversas para testar a eficiência do controlador, a mais relevante delas sendo a distância desejada de 105 pixels, enquanto mantém-se a zona de colisão em 100 pixels. As condições iniciais dos agentes, levando em consideração essas mudanças, encontram-se na Tabela (3.16). Observa-se a priori que para o sistema continuar sem colisão, o sobressinal deve ser quase nulo, se existir.

- $k_v = 20$,
- $k = 0.5$,
- $\phi = 0$,
- $r_d = 105$,

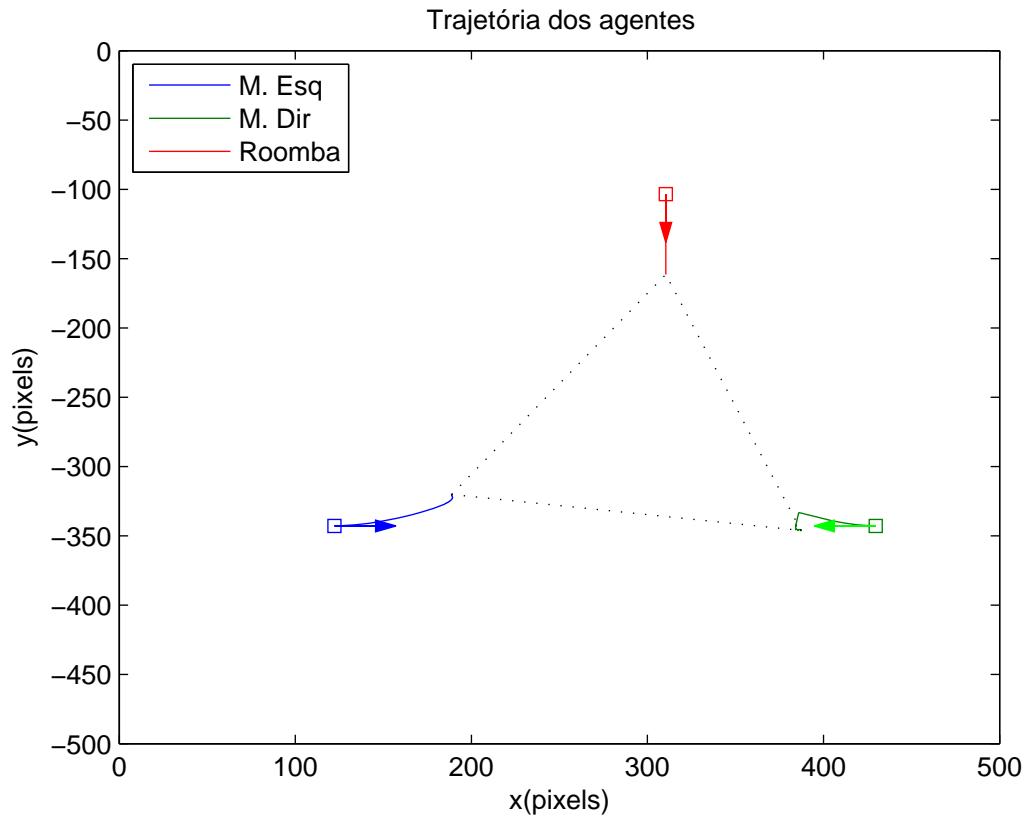


Figura 3.42: Trajetória dos agentes - M.Esq./M.Dir./Rob. Caso 1

$$\bullet q_{el} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{4} \\ \frac{4}{2} \end{bmatrix}$$

$$\bullet q_{er} = \begin{bmatrix} \frac{\pi}{4} \\ -\frac{\pi}{4} \\ \frac{4}{2} \end{bmatrix}$$

$$\bullet \begin{bmatrix} p_{rob} \\ \theta_{rob} \end{bmatrix} = \begin{bmatrix} 0.1m \\ 0.6m \\ 0 \end{bmatrix}$$

A Figura (3.48) mostra a trajetória dos agentes enquanto a Figura (3.49) mostra a orientação dos agentes durante esta trajetória. Observa-se com a Figura (3.50) que o sobressinal do sistema foi realmente nula, e o sistema evitou colisões com perfeição apesar da distância desejada entre agentes ser tão próxima da região de colisão.

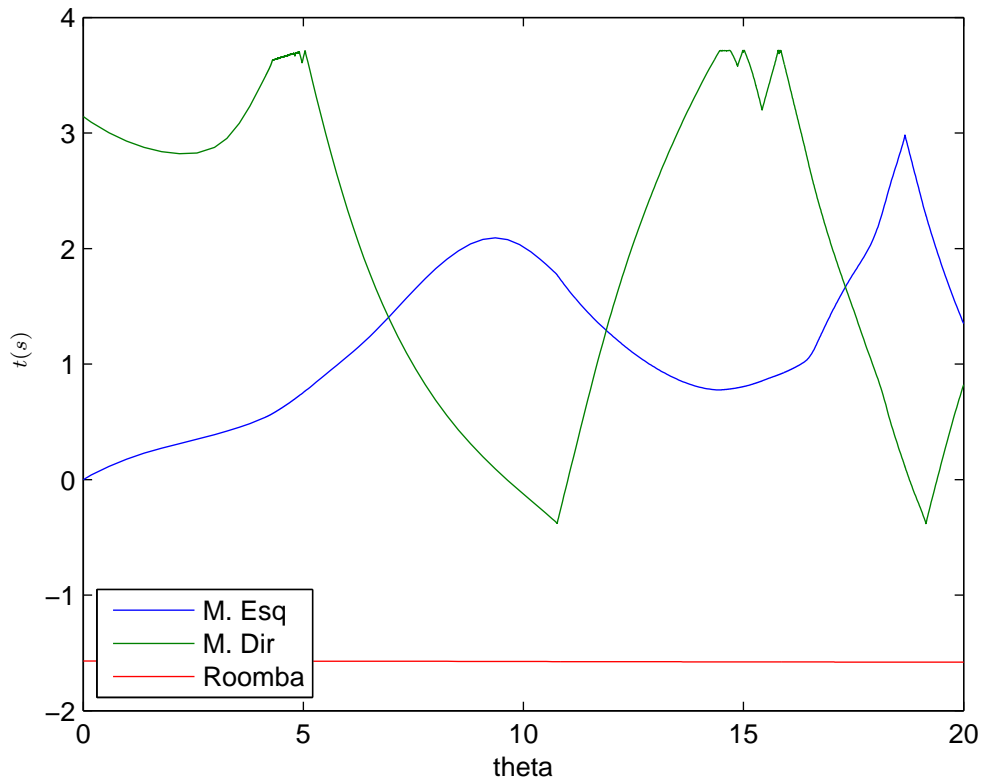


Figura 3.43: Orientação dos agentes - M. Esq./Dir./Roomba Caso 1

Tabela 3.16: Parâmetros iniciais para controle de 3 agentes - Caso 3

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$122.17pixels$	$429.67pixels$	$310.41pixels$
$y(0)$	$-342.91pixels$	$-342.91pixels$	$-344.90pixels$
$\theta(0)$	$\frac{-\pi}{6}$	π	0

A mesma figura também mostra uma diferença de comportamento em relação as simulações anteriores, onde após dois agentes chegarem a uma distância desejada os mesmos continuam a mantê-la durante todo o resto da trajetória.

3.3 Conclusão sobre as simulações

Enfim, analisando o conjunto das simulações realizadas, nota-se os diversos efeitos exercidos pelos parâmetros que podem ser alterados no espaço de trabalho. Nesta seção, serão apresentados a extensão da variação desses parâmetros.

O parâmetro ϕ de rotação da câmera não influencia na minimização da função

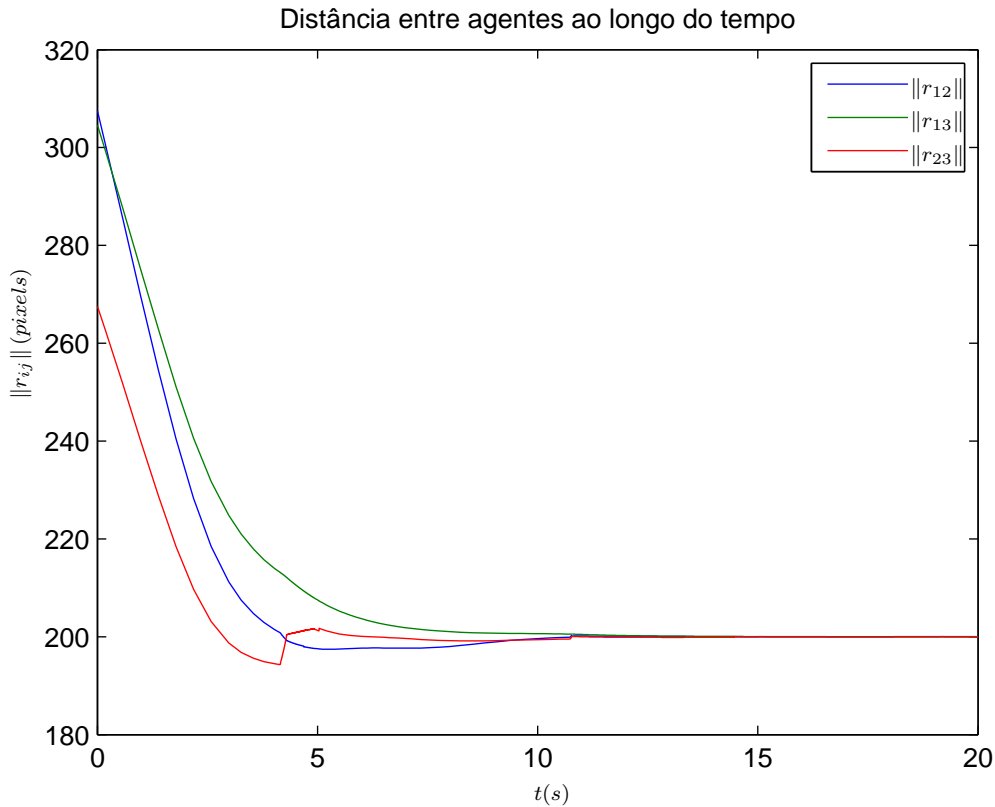


Figura 3.44: Distância entre agentes - M.Esq./M.Dir./Rob. Caso 1

potencial, uma vez que não altera o módulo da distância entre os agentes, como mostraram as simulações envolvendo o Roomba e o Manipulador Esquerdo (figuras 3.32 e 3.35). A ausência de alterações significativas nas simulações envolvendo os três agentes (figuras 3.44 e 3.47), também implica em tal resultado. Do ponto de vista da trajetória dos agentes no plano da imagem, contudo, certamente sofre mudanças, tanto em posição quanto em orientação, como pode ser visto pelo resultado das mesmas simulações (figuras 3.30, 3.33, 3.42 e 3.45).

O ganho de orientação k , é sintonizado com valores pequenos para manter uma trajetória suave, mas devido às restrições do espaço de trabalho e dos limites das juntas dos manipuladores, as simulações mostram que ele tem ação direta na trajetória do sistema, podendo garantir a viabilidade da trajetória com valores adequados. Entretanto, como mostram as simulações com o roomba e o manipulador direito sendo controlados, se usado um ganho de orientação k suficientemente grande, ainda existem diferenças visíveis na trajetória dos agentes (figuras 3.36 e 3.39), mesmo para trajetórias suaves. Apesar disso, foi visto que apesar da diferença na trajetória, o

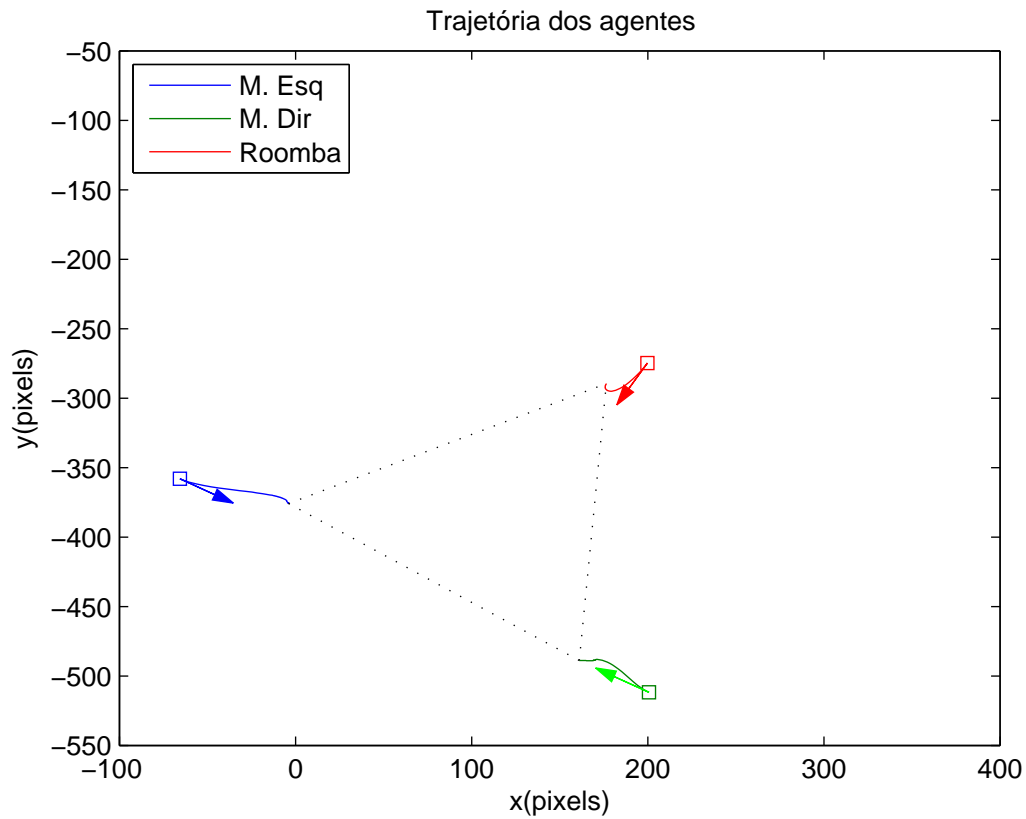


Figura 3.45: Trajetória dos agentes - M.Esq./M.Dir./Rob. Caso 2

tempo que o sistema leva para alcançar sua configuração final não se alteram entre um caso e o outro.

Já o ganho de velocidade linear k_v tem, por sua vez, influência direta no tempo que o sistema leva para alcançar a formação que minimiza a função potencial V . Vê-se nas simulações que um aumento no ganho tem uma redução no tempo em que os agentes levam para percorrer uma trajetória, porém não altera-se muito a trajetória em si. As figuras (3.26) e (3.47), da simulação com os três agentes, ilustram bem a diferença no tempo para alcançar a formação.

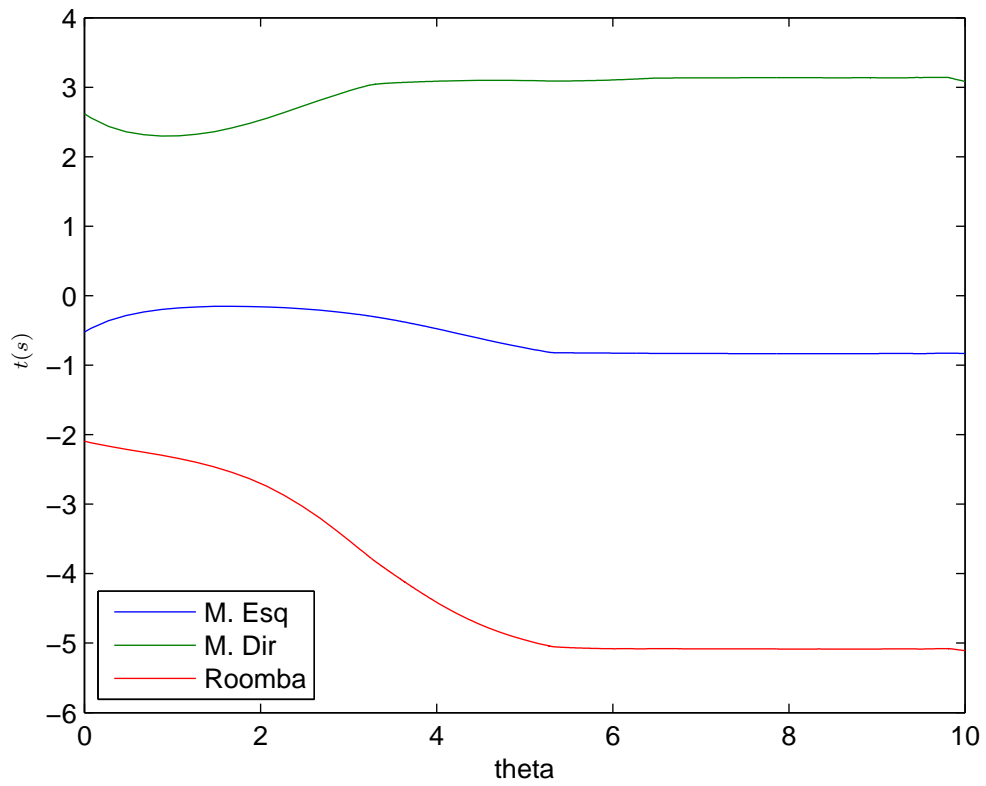


Figura 3.46: Orientação dos agentes - M. Esq./Dir./Roomba Caso 2

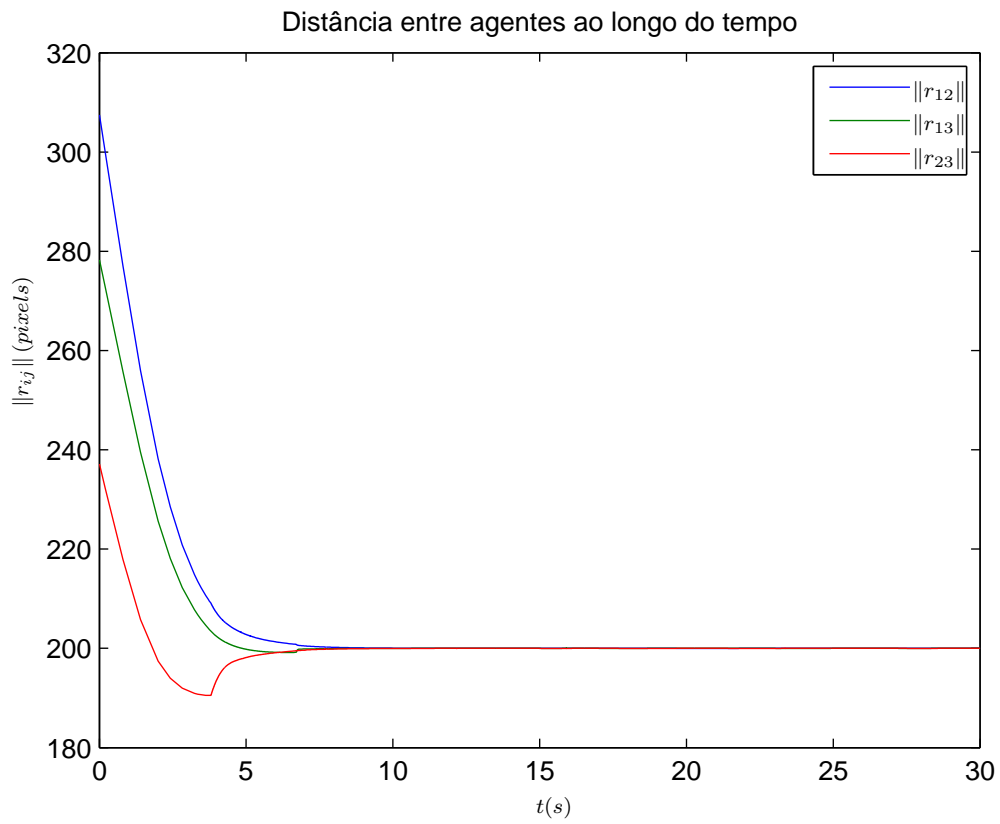


Figura 3.47: Distância entre agentes - M.Esq./M.Dir./Rob. Caso 2

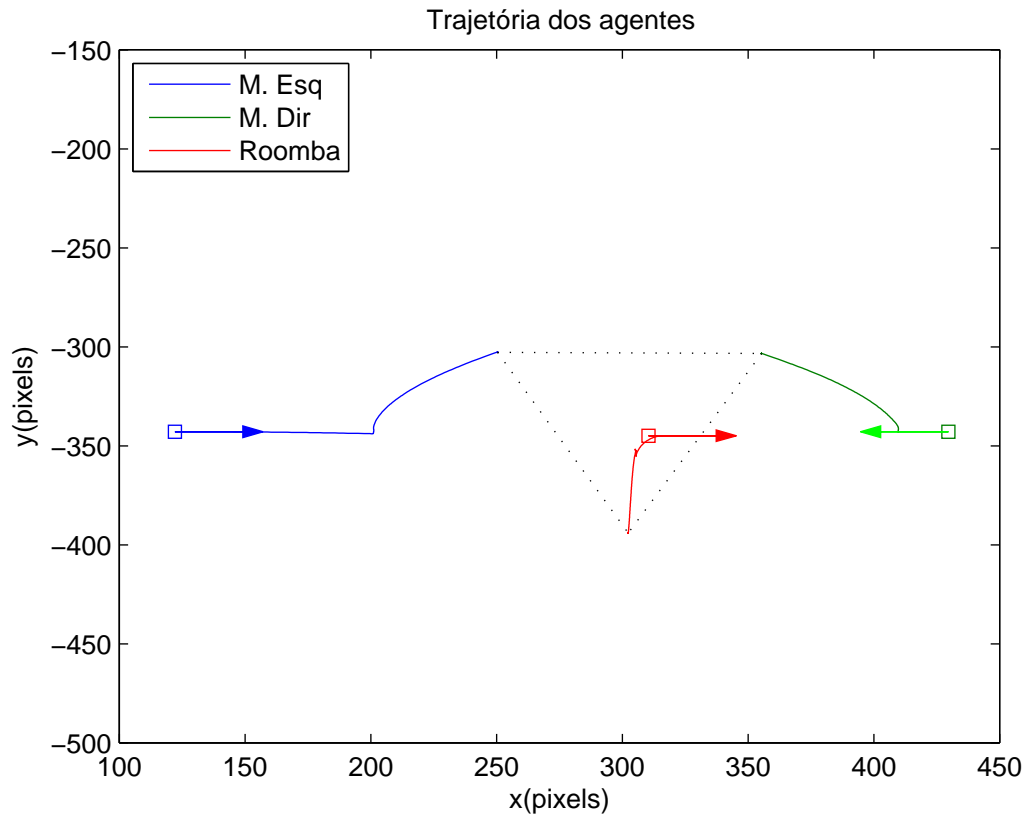


Figura 3.48: Trajetória dos agentes - M.Esq./M.Dir./Rob. Caso 3

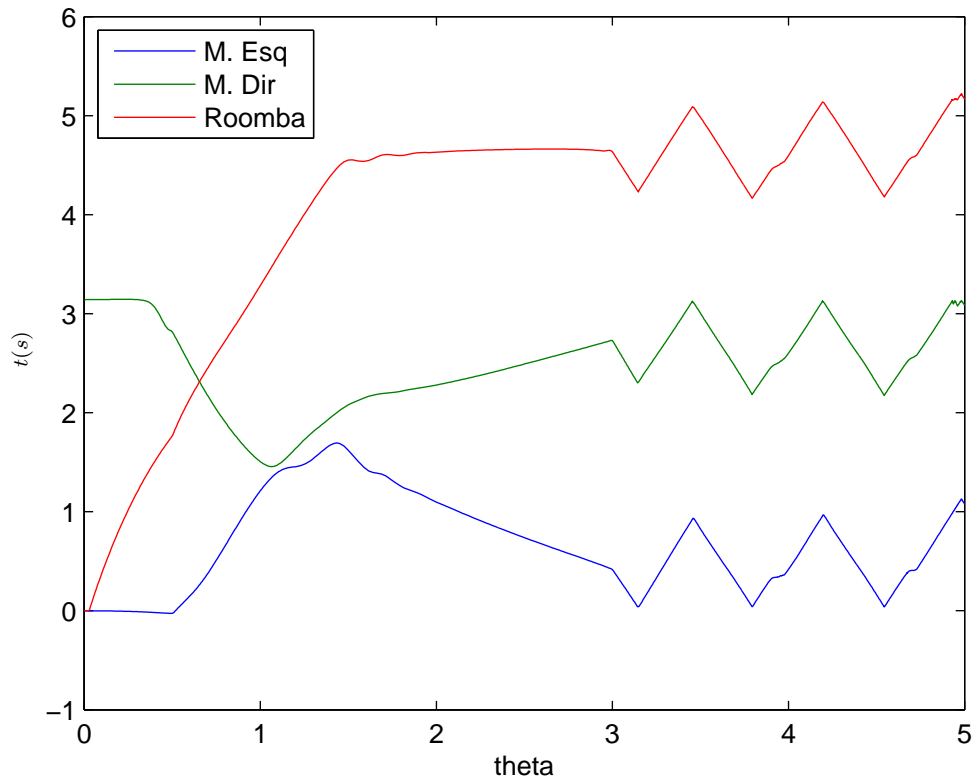


Figura 3.49: Orientação dos agentes - M. Esq./Dir./Roomba Caso 3

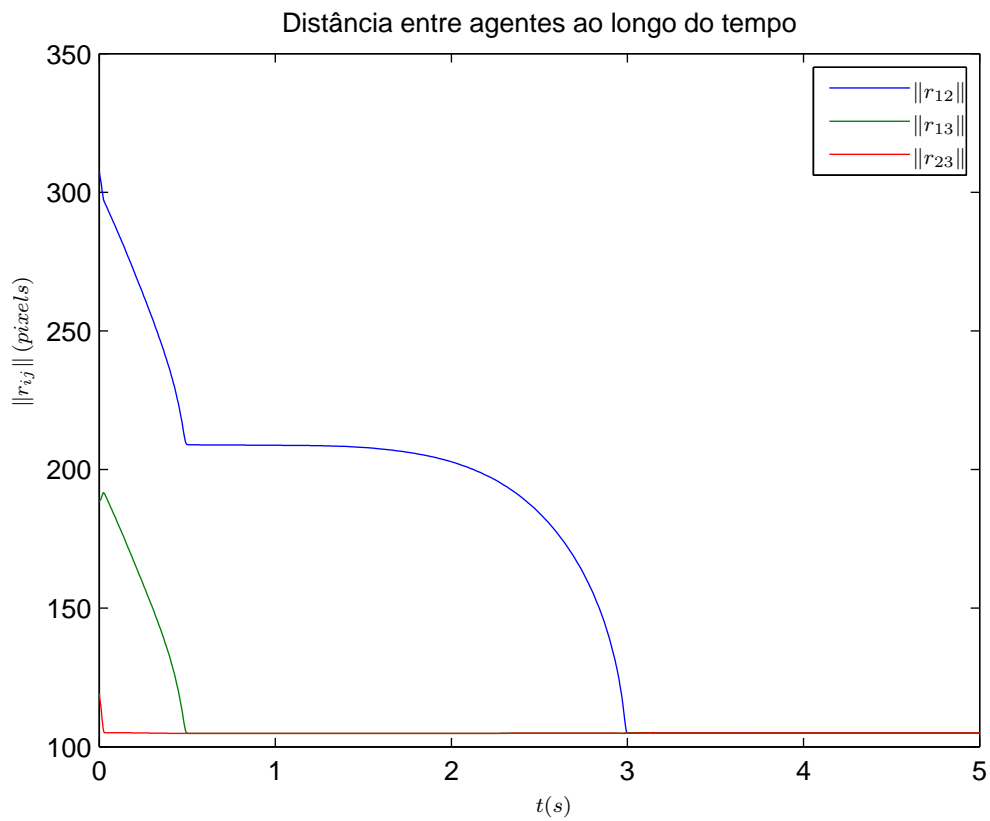


Figura 3.50: Distância entre agentes - M.Esq./M.Dir./Rob. Caso 3

Capítulo 4

Resultados Experimentais

Este capítulo é dividido em duas seções. Na primeira, apresentar-se-á os equipamentos usados neste projeto, com uma breve explanação de como são usados, e qual o seu propósito no experimento. Na seção seguinte, serão mostrados os experimentos propriamente ditos, com gráficos de trajetória, imagens de suas configurações iniciais e finais, bem como os parâmetros que são definidos para o cálculo do controle. Os experimentos não serão conduzidos de forma contínua, mas em passos, esperando que cada etapa da rotina seja completada antes de se começar a etapa seguinte.

4.1 Equipamentos

Nesta seção, estão listados os equipamentos necessários para realizar o experimento, além de breves explicações sobre seu modo de operação e adaptações feita nas equações do modelo para atender especificações dos mesmos. A Figura (4.1) mostra a ligação entre os dispositivos listados a seguir:

- Uma webcam Logitech QuickCam Communicate Deluxe
- Motoman Dia10, um controlador de alta velocidade e um computador para comunicação
- iRobot Roomba 530 series
- Raspberry Pi Model B
- Um computador com Matlab R2013b

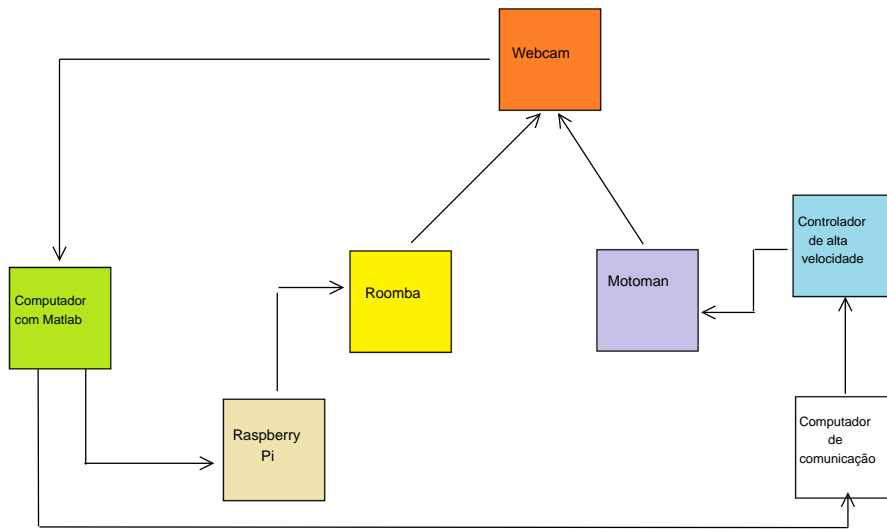


Figura 4.1: Diagrama ilustrando disposição e comunicação dos equipamentos



Figura 4.2: Logitech QuickCam Communicate Deluxe

4.1.1 Logitech QuickCam Communicate Deluxe

A webcam (Figura 4.2) está fixada, verticalmente sobre o espaço de trabalho, e é acessada pelo Matlab através de comunicação USB. Sua função é fornecer uma imagem do espaço de trabalho a cada passo da experiência. A baixa resolução da câmera (640 x 480 pixels), gera certos problemas para a detecção de padrões que é usada neste projeto, até mesmo gerando necessidade de elevar o padrão de identificação do roomba, originalmente na altura do chão, até aproximadamente a altura dos braços do motoman, visto na Figura (4.3).



Figura 4.3: Roomba com haste para elevar o padrão-alvo

Análise do efeito da haste na detecção do roomba

A diferença entre a detecção do padrão-alvo e a real posição do roomba da imagem é um fator que não foi computado nos experimentos relacionados no projeto. Entretanto, o problema será analisado, ao menos. A situação está descrita na Figura (4.4)

Seja λ a distância focal, x_0 a distância entre o roomba e o eixo de rotação da câmera, z_0 a distância entre o padrão-alvo e plano da imagem. Deseja-se obter o erro x_d , entre o padrão-alvo projetado na imagem e o roomba projetado na imagem.

Por semelhança de triângulos, é possível obter a equação (4.1).

$$x_d + x_p = \frac{\lambda}{\lambda + z_0} x_o \quad (4.1)$$

Para achar o erro x_d em função dos valores conhecidos, resta então achar a distância x_p que é a projeção da distância x_0 no plano da imagem. Tal distância também pode ser facilmente calculada por uma semelhança de triângulos, já que se trata de uma projeção, como mostra a equação (4.2).

$$x_i = \frac{\lambda}{\lambda + z_0 + z_1} x_o \quad (4.2)$$

Com as equações (4.1) e (4.2), enfim, obtém-se o erro desejado, x_d , com a equação

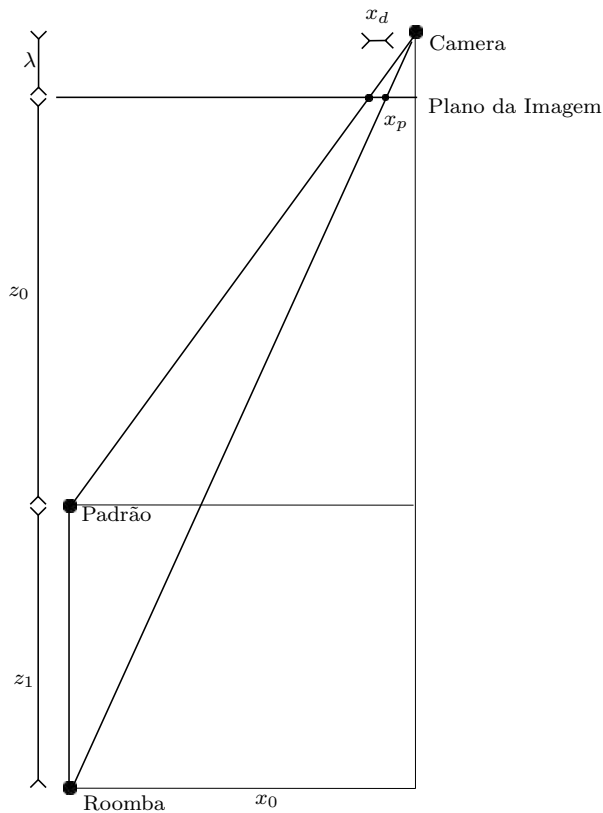


Figura 4.4: Modelo da projeção do padrão-alvo e do roomba na imagem

(4.3).

$$x_d = \left(\frac{\lambda}{\lambda + z_0} - \frac{\lambda}{\lambda + z_0 + z_1} \right) x_0 \quad (4.3)$$

Fica claro então, que o erro entre a projeção do padrão alvo e do roomba aumento quanto mais distante o roomba estiver do eixo de rotação da câmera.

4.1.2 iRobot Roomba 530 series

O Roomba é um aspirador de pó robótico, circular, produzido pela empresa iRobot (Figura 4.5). Ele possui duas rodas controladas de forma independente, o que permite trajetórias sem restrição de curvatura para o robô. Para este projeto, ele é um típico robô cinemático diferencial, controlado pela velocidade de suas rodas, com um limite de ± 500 mm/s em cada uma.

O roomba recebe comandos através de uma porta serial, localizada em sua face superior. Do manual (iRo, 2007), vê-se os comandos listados abaixo, necessários para o projeto, números entre colchetes representando a sequência serial enviada ao



Figura 4.5: iRobot Roomba 530 Series

roomba.

- Comando [128] inicializa a *open interface* do robô, em modo passivo, e sempre deve ser o primeiro comando.
- Comando [131] troca o robô para modo seguro, permitindo que ele seja controlado.
- Comando [145][*Right velocity high byte*][*Right velocity low byte*][*Left velocity high byte*][*Left velocity low byte*] é a sequência que controla o roomba pela velocidade em cada roda. Os quatro bytes de dados são interpretados como dois valores inteiros em 16-bits, usando complemento de dois.
- Comando [137] comando o roomba através de sua velocidade e curvatura da trajetória, ele será usado com a sequência [0][0][0][0] para desligar os motores do roomba.

Embora o controle do roomba seja por velocidade, como o sistema está sendo calculado de forma discreta, a atualização da posição do roomba entre cada passo ocorre em passos como previamente mencionado nesse capítulo, pela equação (4.4).

$$\begin{bmatrix} x(n) \\ y(n) \\ \theta(n) \end{bmatrix} = \begin{bmatrix} x(n-1) \\ y(n-1) \\ \theta(n-1) \end{bmatrix} + h \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (4.4)$$

sendo h o passo do sistema.

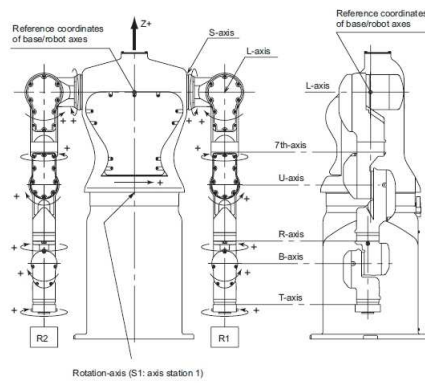


Figura 4.6: Juntas do motoman, com nome e especificações.



Figura 4.7: Motoman Dia10 NX100

4.1.3 Motoman Dia10

O Motoman é um robô com dois manipuladores robóticos, cada um com 7 juntas de revolução independentes, mais uma junta de revolução em seu torso, num total de 15 graus de liberdade. Uma esquemática do funcionamento de suas juntas e uma visualização do robô em si se encontram respectivamente nas figuras (4.6) e (4.7).

O manual completo de todo o processo operacional do motoman, do acionamento ao controle, pode ser visto em (RIBEIRO, 2013), mas verá-se a seguir uma explicação sucinta da rotina de funcionamento do mesmo. Para controlar o motoman através de um computador, é necessário conectar o mesmo a um controlador de alta velocidade, capaz de conciliar a malha de tempo real do controlador industrial e a malha de controle implementada no computador.

Dos equipamentos disponíveis para este projeto, tal controlador faz a transmissão entre o motoman e o computador através de uma rede industrial Star Fabric, que utiliza um cabeamento de ethernet para recepção de informações e outro para envio, o que faz necessário o uso de um computador especial naquele destinado a controlar o Motoman. O controlador espera parâmetros de posição, atual e desejada, para cada uma das juntas, em pulsos do encoder. Esses pulsos são convertidos para radianos em uma biblioteca de comunicação com o controlador de alta velocidade. É essa biblioteca que vai ser acessada do computador que vai calcular o controle deste projeto, através do Robot Racounteur.

O Robot Racounteur é uma biblioteca de comunicação, com suporte a Matlab, com a qual é definido um nó, no caso o computador conectado ao controlador do Motoman, o qual possui um objeto. O objeto definido nesse trabalho, o motoman, tem duas propriedades. A primeira, `ActualRobotJointAngles`, retorna a posição atual das juntas do motoman, enquanto a segunda, `DesiredRobotJointAngles`, envia para o Motoman a nova posição desejada para as juntas, confirmando-o como um robô controlado por posição. A primeira função retorna um vetor 15×1 , onde estão listados respectivamente, o ângulo do torso, os ângulos do braço direito, do ombro ao punho, e os ângulos do braço esquerdo. A segunda aceita como argumento um vetor no mesmo formato. Para um computador qualquer na rede acessar esse nó, basta conectar-se ao endereço de IP e porta do computador que está rodando o serviço.

Como visto, no capítulo 2, neste trabalho, usa-se apenas 6 das 15 juntas de revolução do Motoman, 3 de cada braço, em configuração planar. No vetor de 15 juntas que usados para controlar o Motoman, para os manipuladores modelados, essas juntas correspondem às juntas 3, 5 e 7 para o manipulador direito, e 10, 12 e 14 para o manipulador esquerdo. Essas juntas estão na ordem q_1 , q_2 , q_3 anteriormente apresentadas no modelo.

Por último, podemos ver os limites das juntas utilizadas neste projeto, na Figura (4.8). Ressalta-se que o workspace mostrado na figura não inclui a restrição do espaço de trabalho gerado pela captação de imagem na câmera, nem possíveis colisões entre os dois braços.

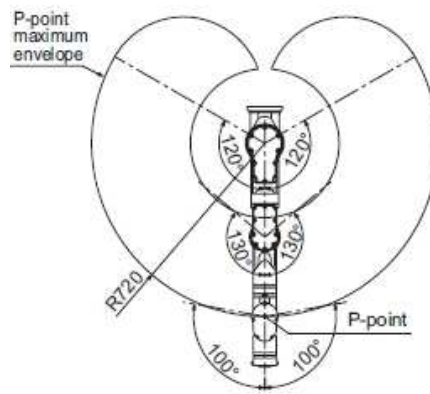


Figura 4.8: Limites das juntas do braço do motoman, em configuração planar.

4.1.4 Raspberry Pi Model B

Neste projeto, a raspberry tem uma de suas entradas USB conectada a um adaptador de rede sem fio, a outra conectada ao roomba através de um adaptador USB-Serial.

Sua função é sustentar um servidor, em python, que espera uma requisição do cliente, no caso o Matlab, para realizar uma simples rotina. A rotina é executada sempre que ele recebe, por protocolo udp através da rede sem fio, informações com as velocidades das rodas do roomba deveria ter naquele instante. Quando isso acontece ele converte tais velocidades de valores inteiros para bytes, e conecta-se ao roomba através da serial. Ele envia o comando para ligar os motores do roomba, fazendo que as rodas andem na velocidade desejada pelo passo da experiência, e então desliga os motores e fecha a conexão serial. Assim que esta pequena rotina é concluída, ele novamente aguarda pelo próximo pedido do Matlab.

4.1.5 Matlab 2013b

O Matlab é responsável por três distintas etapas do experimento, listadas a seguir

- Processamento de Imagem
- Cálculo do Controle
- Envio do Sinal de Controle

Processamento de Imagem

Na primeira das etapas, o Matlab acessa através da *Image Acquisition Toolbox*, usando o comando *videoinput* para acessar a câmera, e o comando *getsnapshot* para adquirir um frame do workspace, sempre que necessário. Depois de ter adquirido a imagem, o Matlab procura por padrões-alvos, previamente definidos, através do *Computer Vision toolbox*. A base de tal algoritmo é a extração e comparação de Speeded-Up Robust Features, SURF, um algoritmo proposto em (BAY *et al.*, 2006) que detecta regiões de interesse de uma imagem, descritores distintos, invariantes frente a deformações geométricas e fotométricas, e depois compara esses descritores com os de uma segunda imagem, buscando semelhanças entre elas.

Seguindo uma sequência, a detecção é implantada da seguinte forma:

- Detectar SURF *features* da imagem e do padrão-alvo através da função `detectSURFFeatures`.
- Comparar tais *features*, pareando pontos iguais, com a função `matchFeatures`.
- Reduzir as *features* detectadas no passo 1 apenas às *features* dos pontos pareados no passo 2, apagando qualquer *feature* não-pareada.
- Com os dois novos conjuntos de pontos, estimar a transformada geométrica através da função `estimateGeometricTransform`, que utiliza o algoritmo MSAC (TORR e ZISSERMAN, 2000) para remover discrepâncias que possam interferir no resultado da transformada.
- Cria-se um retângulo que representa os limites do padrão-alvo, uma caixa de detecção do tamanho do padrão-alvo, mais especificamente, uma matriz com a coordenada dos 4 vértices deste quadrado.
- Aplica-se a transformada geométrica à caixa de detecção, gerando os 4 vértices do retângulo na imagem onde se deseja detectar o objeto
- Com estes 4 novos vértices, obtém-se as coordenadas do centro do padrão-alvo na imagem do espaço de trabalho.

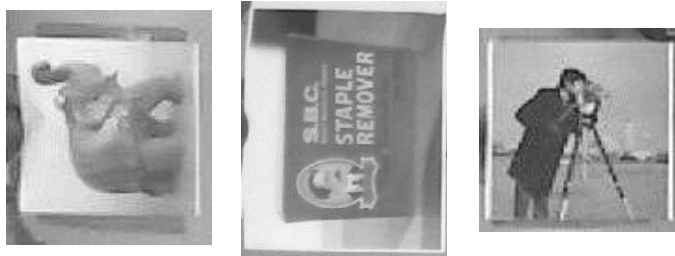


Figura 4.9: Padrões-Alvo, da esquerda para a direita, M. Esq., M. Dir. e Roomba

- Para descobrir a orientação, inverte-se a transformada geométrica, o primeiro elemento da primeira coluna da nova matriz é o cosseno da rotação multiplicado por uma fator de escala, enquanto o segundo elemento da primeira coluna é o seno da rotação igualmente multiplicado pelo mesmo fator.
- Aplica-se finalmente, a função atan2 , obtendo a orientação do objeto a ser detectado.

Os padrões-alvos mencionados podem ser vistos na Figura (4.9). Da esquerda para a direita, veem-se, respectivamente, os padrões que identificam o manipulador esquerdo, o manipulador direito e o roomba.

Cálculo do Controle

Essa é uma etapa simples, onde o Matlab pega a posição e orientação dos agentes, conforme o processo descrito na etapa anterior, e calcula o sinal de controle para cada um deles, como modelado no capítulo anterior.

Envio do Sinal de Controle

Duas funções são usadas para enviar o sinal de controle, uma para o motoman e outra para o roomba.

A primeira função recebe as velocidades das rodas virtuais de cada um dos manipuladores e o objeto motoman criado pelo RobotRaconteur como argumentos. Ela então realiza todos os cálculos descritos desde a equação (3.4) até a equação (3.6) para o manipulador esquerdo, e, analogamente, da equação (3.9) até a equação (3.11) para o manipulador direito. Como visto na descrição do motoman nesse capítulo, o motoman é controlado através da posição de suas juntas, então a função multiplica

a velocidade das juntas por um passo fixo da experiência, soma a posição atual das juntas e envia as posições atualizados para o computador que serve de *host* para o motoman, como descrito pela equação (4.5).

$$q(n) = q(n - 1) + h\dot{q} \quad (4.5)$$

sendo h o passo do sistema.

O roomba por outro lado é um robô cinemático diferencial, controlado pela velocidade de suas rodas, como dito anteriormente, então o que esta função faz é enviar, como um cliente, a velocidade das rodas esquerda e direita, através de um protocolo udp, via rede sem fio, para a Raspberry Pi.

4.2 Resultados

Neste seção, serão finalmente apresentados os resultados experimentais do projeto, com os equipamentos descritos na seção anterior. Assim como nas simulações, foram realizados experimentos em três etapas

- Apenas um dos agentes irá buscar a formação, com os demais agentes imóveis.
- Um par de agentes irá buscar a formação, indo atrás do outro agente imóvel.
- Todos os três agentes irão buscar a formação.

A formação a ser alcançada ainda é um triângulo, isósceles na primeira etapa, equilátero nas demais. Os parâmetros necessários para calcular o controle que se mantém o mesmo durante todos os experimentos estão listados na tabela (4.1), onde a_i é a dimensão do elo i do motoman, c é o diâmetro da zona de colisão, R_s é a região de vizinhança, r é a distância entre o centro de massa do roomba e uma de suas rodas. Note que, em relação ao capítulo anterior, certos parâmetros como a rotação ϕ da câmera e a posição de sua origem serão tratados como completamente desconhecidos, e outros como ângulo q das juntas dos manipuladores serão usados para garantir somente que os efetuadores dos mesmos se locomovam segundo restrições não-holonômicas.

As considerações feitas durante a simulação são ainda mais importantes no caso real, onde o espaço de trabalho é deveras limitado. As posições iniciais do roomba

Tabela 4.1: Parâmetros comuns a todos os experimentos

Parâmetro	Valor
a_0	$0.38m$
a_1	$0.39m$
a_2	$0.33m$
a_3	$0.21m$
c	$100pixels$
R_s	$1000pixels$
r	$125mm$

foram escolhidas de modo que ele ficasse dentro do espaço de trabalho visto pela câmera. Ângulos iniciais para os manipuladores foram escolhidos de modo a fazer uso do espaço de trabalho da melhor forma possível. E acima disso, as trajetórias foram escolhidas de forma a evitar limites das juntas do motoman e as singularidades do seu jacobiano. Para todas as distâncias desejadas r_d escolhidas, assume-se uma tolerância de $\pm 10 pixels$.

4.2.1 Um agente buscando a formação

Primeiro, será mostrada a ação do campo potencial sobre um agente, os demais sempre com $v_i = 0$ e $\omega_i = 0$, com os parâmetros pré-estabelecidos na tabela 4.1, e as demais condições especificadas no segmento correspondente.

Roomba

O ganho de velocidade angular k e o ganho de velocidade linear k_v foram sintonizados com base nos resultados das simulações mostradas no capítulo anterior, a distância r_d escolhida de acordo com a trajetória realizada adequada. O parâmetro h representa o passo usado para a experiência, em segundos. A tabela (4.2) dá valores à configuração inicial do experimento, enquanto a Figura (4.10) ilustra tal configuração.

- $k_v = 20$,
- $k = 0.5$,

Tabela 4.2: Posição e orientação inicial dos agentes - Roomba

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$79.50pixels$	$512.53pixels$	$292.5379pixels$
$y(0)$	$-344.04pixels$	$-337.99pixels$	$-189.44pixels$
$\theta(0)$	0.0682	-3.1175	-1.5283



Figura 4.10: Posição e orientação inicial dos agentes - Roomba

- $r_d = 250 \pm 10 pixels$
- $h = 0.6s$

Dos dados coletados, vê-se que nesse movimento simples, o sistema leva 6.6 segundos para alcançar a formação desejada como mostra a Figura (4.12). A distância entre o roomba e o manipulador esquerdo é inicialmente menor que a desejada, mas não se encontra próximo da zona de colisão, então não há o rápido movimento que foi visto na simulação Roomba - Caso 2 (Figura 3.11).

Na Figura (4.11), tem-se a trajetória do roomba de sua posição inicial até alcançar a formação. É de interesse notar que a trajetória descrita se aproxima dos resultados simulados, na simulação do Roomba - Caso 1, com um sistema estável

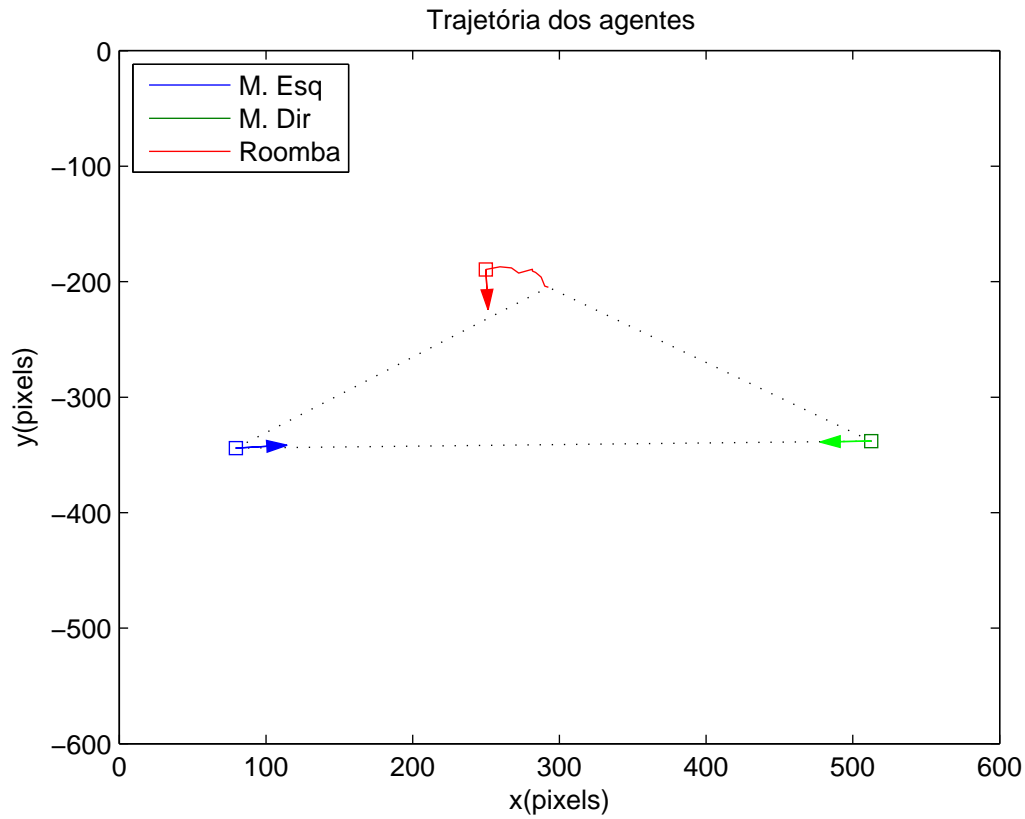


Figura 4.11: Trajetória dos agentes - Roomba

Tabela 4.3: Posição e orientação final dos agentes - Roomba

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$79.50pixels$	$512.53pixels$	$292.54pixels$
$y(0)$	$-344.04pixels$	$-337.99pixels$	$-204.87pixels$
$\theta(0)$	0.0682	-3.1175	-0.8475

que converge para a formação. Tal situação é ilustrada pela Figura (4.13), com valores da configuração na tabela (4.3) e variação de orientação dos agentes na Figura (4.14)

Manipulador Esquerdo

Os ganhos k e k_v foram mantidos em relação ao experimento anterior. O módulo da distância r_d desejada foi reduzido, gerando uma trajetória que percorre uma distância maior, mas mantendo a mesma dentro dos estados alcançáveis pelo manipulador. A configuração inicial está ilustrada na Figura (4.15), com a posição e

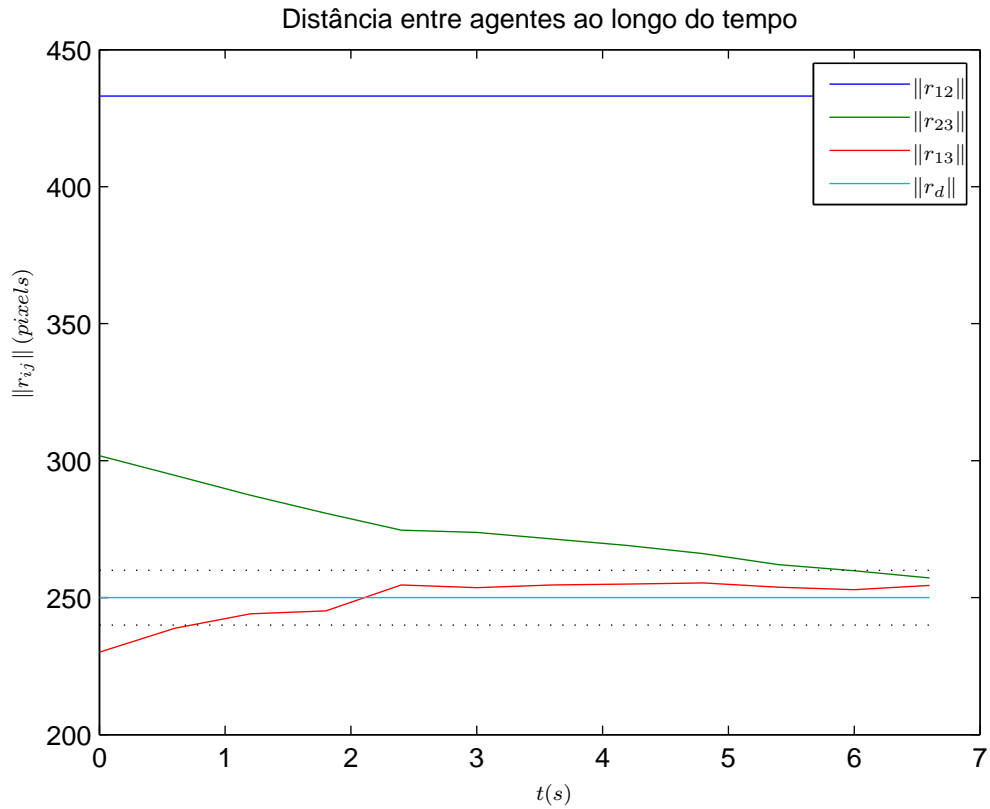


Figura 4.12: Distância entre agentes - Roomba

Tabela 4.4: Posição e orientação inicial dos agentes - M. Esq.

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$95.50pixels$	$517.33pixels$	349.43 pixels
$y(0)$	$-349.17pixels$	$-335.97pixels$	-143.04 pixels
$\theta(0)$	0.1287	3.1101	-1.4655

orientação de cada agente na tabela (4.4).

- $k_v = 20$,
- $k = 0.5$,
- $r_d = 200 \pm 10 pixels$,
- $h = 0.6s$

Dos dados coletados, o sistema leva 12 segundos para alcançar a formação desejada com este trajeto mais longo, como mostra a Figura (4.17). Em torno de 6



Figura 4.13: Posição e orientação final dos agentes - Roomba

segundos, a distância entre o roomba e o manipulador esquerdo se torna menor que a desejada, e persiste em diminuir até 7.2 segundos, quando a orientação é corrigida, mesmo com o baixo ganho de velocidade angular. Observa-se como a correção da orientação é mais suave quando as distâncias estão aumento do que quando a distância se aproxima da região de colisão..

Na Figura (4.16), tem-se a trajetória do roomba de sua posição inicial até alcançar a formação. É de interesse notar que, na trajetória, a súbita correção de orientação conforme a distância dos agentes 1 e 3 diminuía. Entretanto, o sistema ainda permanece estável enquanto converge para a formação. Tal situação é ilustrada pela Figura (4.18), com valores da configuração na tabela (4.5) e orientação na Figura (4.19).

Manipulador Direito

Os ganhos k e k_v foram mantidos em relação ao experimento anterior. O módulo da distância r_d desejada foi também mantido, a trajetória ainda dentro dos estados alcançáveis pelo manipulador. A posição inicial do roomba foi alterada, como

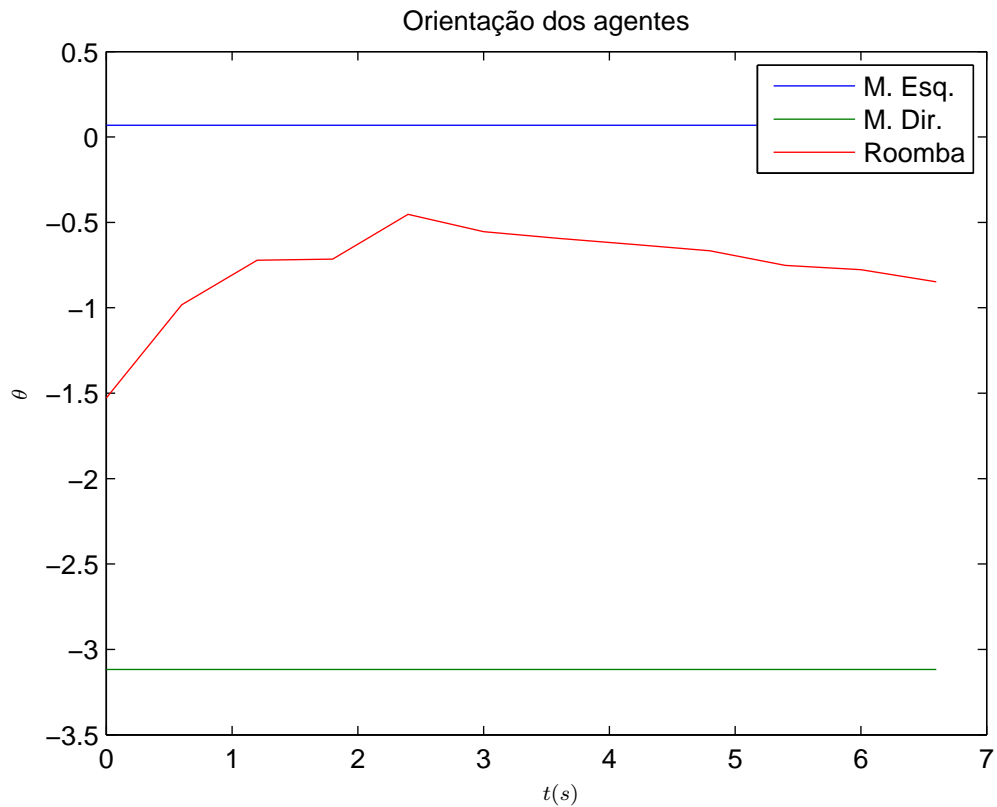


Figura 4.14: Orientação dos agentes - Roomba

Tabela 4.5: Posição e orientação final dos agentes - M. Esq.

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$308.77pixels$	$517.33pixels$	$349.43pixels$
$y(0)$	$-344.46pixels$	$-335.97pixels$	$-143.04pixels$
$\theta(0)$	0.3598	3.1101	-1.4655

pode ser visto na configuração inicial, ilustrada na Figura (4.15), com a posição e orientação de cada agente na tabela (4.4).

- $k_v = 20$,
- $k = 0.5$,
- $r_d = 200 \pm 10 pixels$,
- $h = 0.6s$

Comparado ao experimento anterior, observa-se que este tem uma trajetória simples, sem sobressinal, como mostra a Figura (4.21), desde sua posição inicial até



Figura 4.15: Posição e orientação inicial dos agentes - M. Esq.

Tabela 4.6: Posição e orientação inicial dos agentes - M. Dir.

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$87.23pixels$	$511.51pixels$	$192.36pixels$
$y(0)$	$-346.96pixels$	$-336.56pixels$	$-132.76pixels$
$\theta(0)$	0.0558	-3.1042	0.0335

a configuração final do sistema. A Figura (4.24) O sistema é estável, e a configuração final do mesmo está ilustrada na Figura (4.23). A posição e orientação final dos agentes está na tabela (4.7).

A Figura (4.22) confirma a suavidade da trajetória, o manipulador direito levando 8.4 segundos para alcançar a formação desejada. O sistema tem comportamento dentro do esperado, pelo resultados das simulações Manipulador Direito - Caso 1(figuras 3.18 e 3.20) e Caso 2(figuras 3.21 e 3.23).

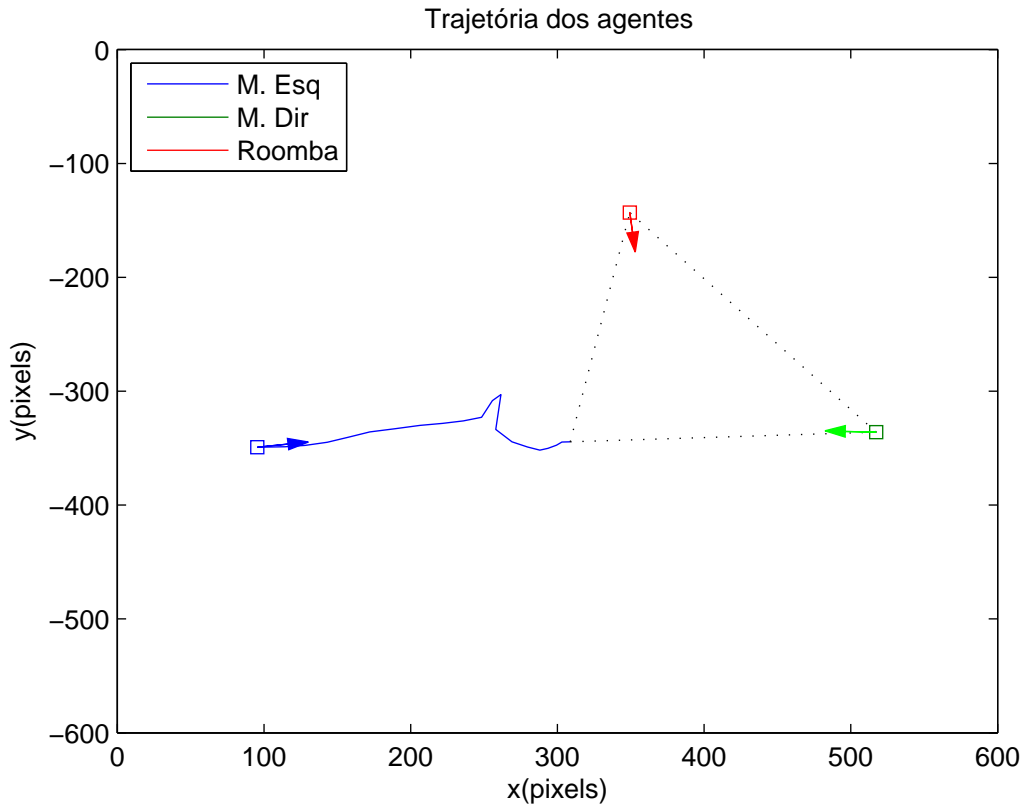


Figura 4.16: Trajetória dos agentes - M. Esq.

Tabela 4.7: Posição e orientação final dos agentes - M. Dir.

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$87.23pixels$	$293.67pixels$	$192.36pixels$
$y(0)$	$-346.96pixels$	$-314.83pixels$	$-132.76pixels$
$\theta(0)$	0.0558	2.8312	0.0335

4.2.2 Dois agentes buscando a formação

Neste segmento, os agentes foram controlados dois a dois, num total de três experimentos. Assim como antes, o agente não controlado permanece com $v_i = 0$ e $\omega_i = 0$, durante toda a duração do experimentos. Os parâmetros comuns a todos foram pré-estabelecidos na tabela 4.1, e as demais condições são especificadas no experimento correspondente. Com dois agentes controlados, a formação desejada é um triângulo equilátero, e a distância entre tais agentes responde mais rápido a alterações, em muitos casos alcançando a distância desejada mais rápido que as demais, como visto na simulação.

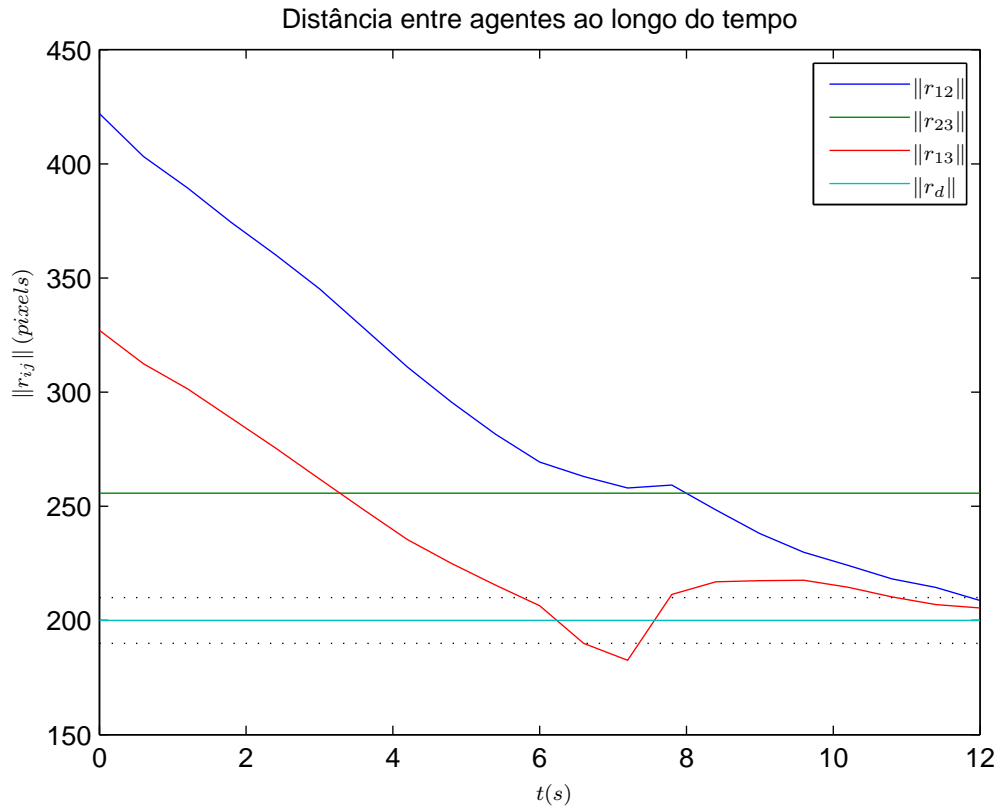


Figura 4.17: Distância entre agentes - M. Esq.

Manipuladores Esquerdo e Direito

Os ganhos k e k_v foram mantidos, por apresentarem tempos de resposta baixos com trajetórias suficientemente suaves, como mostraram as simulações. O módulo da distância r_d desejada foi escolhido de forma a usar o máximo do espaço de trabalho limitado, mantendo trajetórias alcançáveis pelo manipulador. A configuração inicial está ilustrada na Figura (4.25), com a posição e orientação de cada agente na tabela (4.8).

- $k_v = 20$,
- $k = 0.5$,
- $r_d = 200 \pm 10 \text{ pixels}$,
- $h = 0.6s$

Da Figura (4.27), vê-se que a distância entre os agentes 1 e 2, apesar de ser a maior distância inicial, é a primeira a chegar na distância desejada r_d , como



Figura 4.18: Posição e orientação final dos agentes - M. Esq.

Tabela 4.8: Posição e orientação inicial dos agentes - M. Esq./Dir.

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$79.90pixels$	$511.44pixels$	$312.66pixels$
$y(0)$	$-343.91pixels$	$-339.76pixels$	$-132.07pixels$
$\theta(0)$	0.1156	-3.0700	-1.5545

esperado. O sistema leva 7.8 segundos para alcançar a formação dentro dos 10 *pixels* de tolerância, com a distância entre o agente 1 e o agente 3 sendo a última a alcançar o módulo desejado.

A Figura (4.26) mostra a trajetória dos manipuladores da posição inicial até alcançar a formação, enquanto a Figura (4.29) mostra a orientação. Na primeira figura fica ilustrada a formação do triângulo, quase equilátero a menos da tolerância aceitável em r_d , com um sistema estável. Tal situação é ilustrada pela Figura (4.28), com valores da configuração na tabela (4.9).

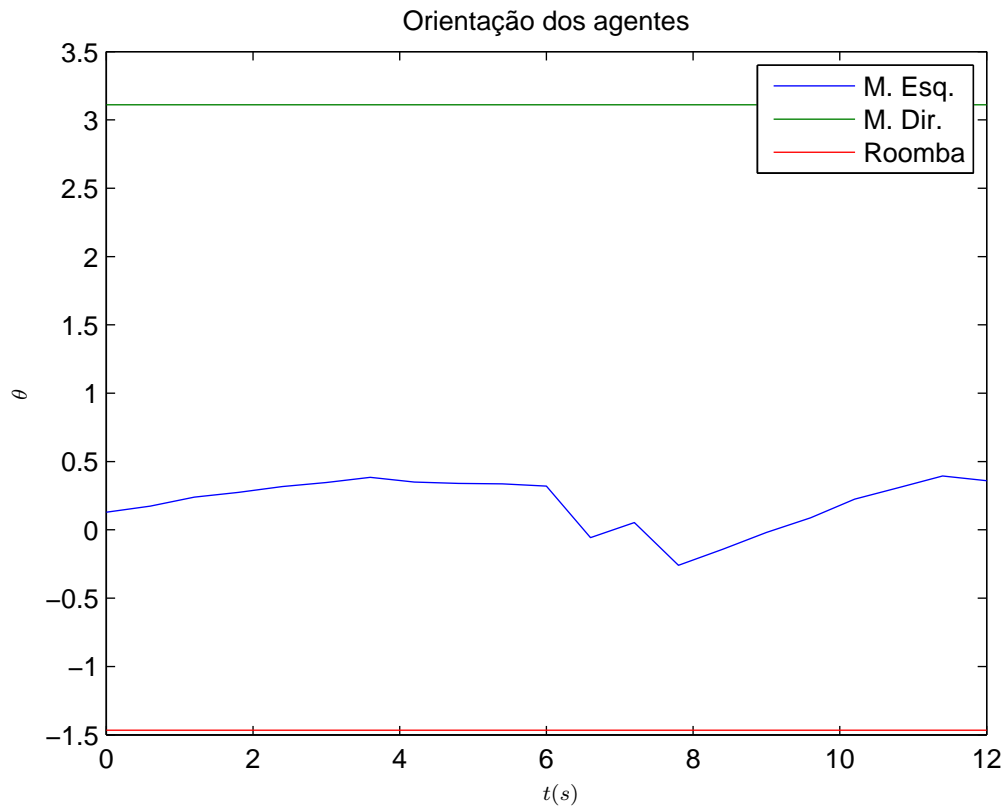


Figura 4.19: Orientação dos agentes - M. Esq.

Tabela 4.9: Posição e orientação final dos agentes - M. Esq./Dir.

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$202.35pixels$	$404.06pixels$	$312.66pixels$
$y(0)$	$-309.38pixels$	$-312.53pixels$	$-132.07pixels$
$\theta(0)$	0.7302	2.4952	-1.5545

Manipulador Esquerdo e Roomba

Os ganhos k e k_v foram mantidos em relação ao experimento anterior. O módulo da distância r_d desejada também não foi alterado, considerando que os agentes tem uma região de colisão de 100 pixels de diâmetro. A Figura (4.30) mostra a configuração inicial do sistema, com valores de posição e orientação descritos tabela (4.10).

- $k_v = 20$,
- $k = 0.5$,
- $r_d = 200 \pm 10 pixels$,



Figura 4.20: Posição e orientação inicial dos agentes - M. Dir.

Tabela 4.10: Posição e orientação inicial dos agentes - M. Esq./Roomba

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$80.46pixels$	$512.53pixels$	$249.97pixels$
$y(0)$	$-344.06pixels$	$-337.99pixels$	$-157.65pixels$
$\theta(0)$	0.0732	-3.0336	-1.5295

- $h = 0.6s$

Dos dados coletados, vê-se que nesse movimento simples, o sistema leva 21.6 segundos para alcançar a formação desejada como mostra a Figura (4.32). A distância entre o roomba e o manipulador esquerdo é a menor entre todos os agentes, e essa distância, como visto, é a que responde mais rápido chegando ao valor desejado em apenas , 4.8 segundos.

Este resultado, junto com a Figura (4.11), mostra um interessante padrão no qual o manipulador esquerdo e o roomba acompanham o movimento um do outro, tentando manter a distância entre os dois. O roomba, ao alcançar a distância desejada com o manipulador direito, tenta circular o mesmo, enquanto o manipulador

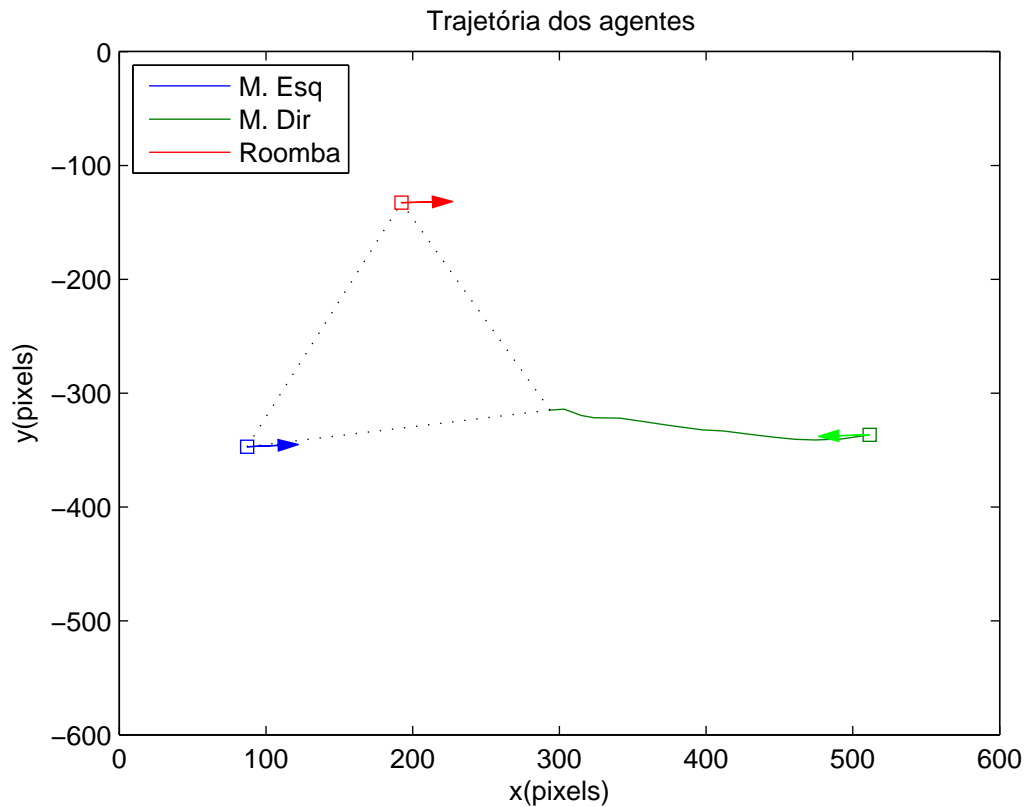


Figura 4.21: Trajetória dos agentes - M. Dir.

Tabela 4.11: Posição e orientação final dos agentes - M. Esq./Roomba

-	M. Esquerdo	M. Direito	Roomba
$x(t)$	$305.07pixels$	$512.53pixels$	$388.06pixels$
$y(t)$	$-362.48pixels$	$-337.99pixels$	$-185.43pixels$
$\theta(t)$	-0.1831	-3.0336	0.4854

esquerdo se aproxima para terminar a formação. Tal situação foi vista na simulação com Três agentes controláveis - Caso 2, como mostra a Figura (3.50) . A Figura (4.34) mostra a como variou a orientação para tal trajetória. A configuração final é ilustrada pela Figura (4.33), com valores da configuração na tabela (4.11).

Manipulador Direito e Roomba

Os ganhos k , k_v e o módulo da distância desejada r_d foram não foram alterados em relação ao experimento anterior. A única alteração das condições iniciais do sistema vem da posição e orientação do roomba. Tal configuração inicial está ilustrada na

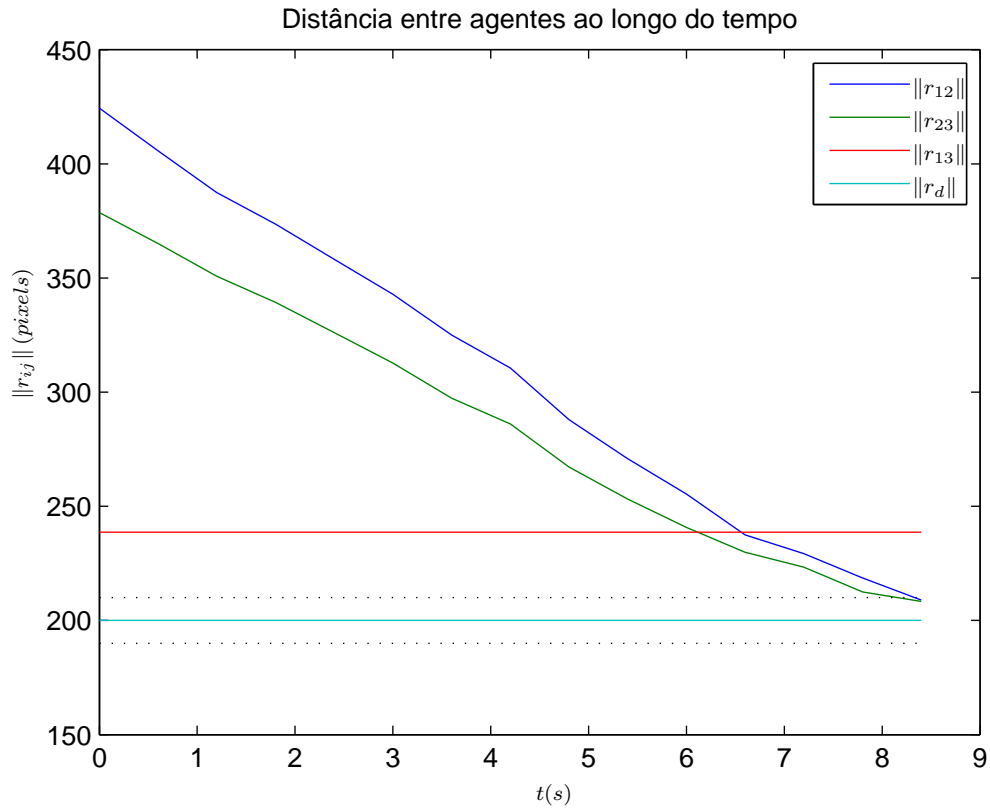


Figura 4.22: Distância entre agentes - M. Dir.

Tabela 4.12: Posição e orientação inicial dos agentes - M. Dir./Roomba

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$79.50pixels$	$509.68pixels$	$336.40pixels$
$y(0)$	$-344.04pixels$	$-337.72pixels$	$-71.03pixels$
$\theta(0)$	0.0377	-3.0869	-1.5433

Figura (4.35), com a posição e orientação de cada agente na tabela (4.12).

- $k_v = 20$,
- $k = 0.5$,
- $r_d = 200 \pm 10 pixels$,
- $h = 0.6s$

Dos dados coletados, nesse experimento, o sistema leva 14.4 segundos para alcançar a formação desejada como mostra a Figura (4.37). Vê-se claramente um



Figura 4.23: Posição e orientação final dos agentes - M. Dir.

Tabela 4.13: Posição e orientação final dos agentes - M. Dir./Roomba

-	M. Esquerdo	M. Direito	Roomba
$x(t)$	$79.50pixels$	$287.21pixels$	$199.62pixels$
$y(t)$	$-344.04pixels$	$-374.20pixels$	$-183.73pixels$
$\theta(t)$	0.0377	-3.0869	-2.6458

grande começando em $t = 7.8s$, causado pela aproximação dos agentes 2 e 3, onde $\|r_{23}\| = 173pixels$, e do passo grande que estava sendo usado no experimento. Tal situação foi escolhida para ilustrar esse problema.

Na Figura (4.36), tem-se a trajetória do agentes de suas posições iniciais até alcançar a formação e na Figura (4.39) sua orientação. Nota-se na trajetória facilmente onde houve o sobressinal mencionado, e como o sistema se corrige após o mesmo, até alcançar a formação. A configuração final é ilustrada pela Figura (4.38), com valores da configuração na tabela (4.13).

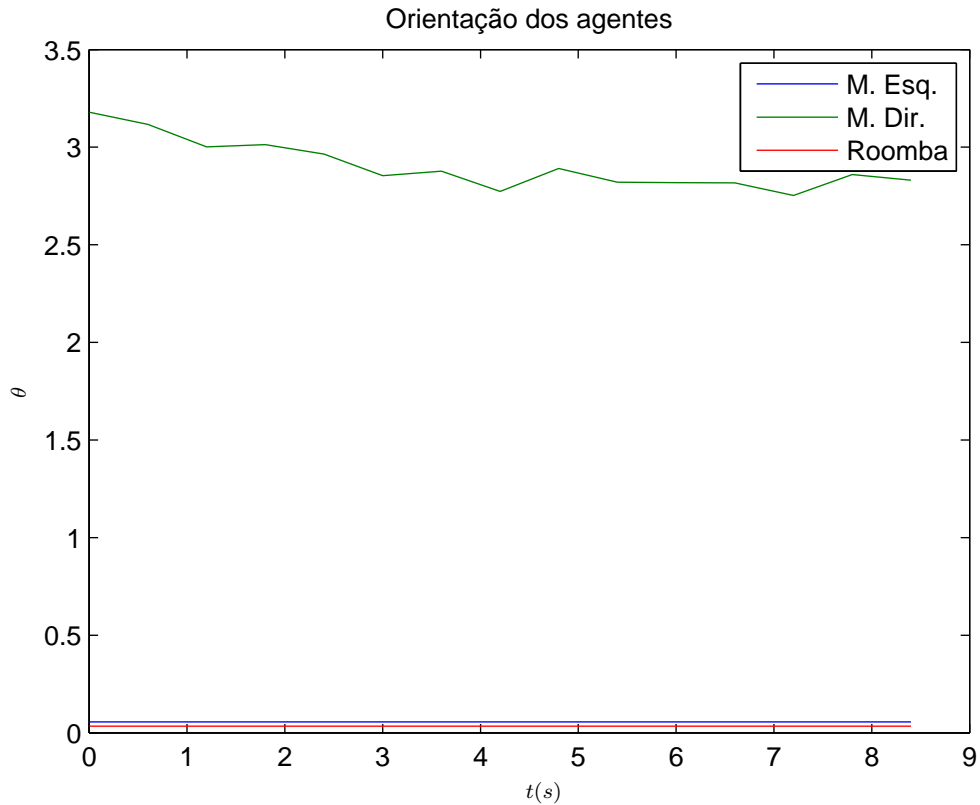


Figura 4.24: Orientação dos agentes - M. Dir.

4.2.3 Três agentes buscando a formação

Nesta última subseção finalmente será vista a ação do controle sobre os três agentes, objetivo final deste projeto. A formação alcançada aqui ainda é um triângulo equilátero e cada par de agentes contribui para o controle dos mesmos.

Os ganhos k e k_v foram mantidos em relação ao experimento anterior. O módulo da distância r_d desejada, igualmente, não foi alterado, para obter-se resultados comparáveis aos experimentos anteriores. A configuração inicial está ilustrada na Figura (4.40), com a posição e orientação de cada agente na tabela (4.14).

- $k_v = 20$,
- $k = 0.5$,
- $r_d = 200 \pm 10 \text{ pixels}$,
- $h = 0.6s$

Detected Target



Figura 4.25: Posição e orientação inicial dos agentes - M. Esq./Dir.

Tabela 4.14: Posição e orientação inicial dos agentes - M. Esq./M. Dir./Roomba

-	M. Esquerdo	M. Direito	Roomba
$x(0)$	$79.29pixels$	$512.66pixels$	$320.99pixels$
$y(0)$	$-345.76pixels$	$-339.95pixels$	$-122.98pixels$
$\theta(0)$	0.0204	3.1329	-1.6479

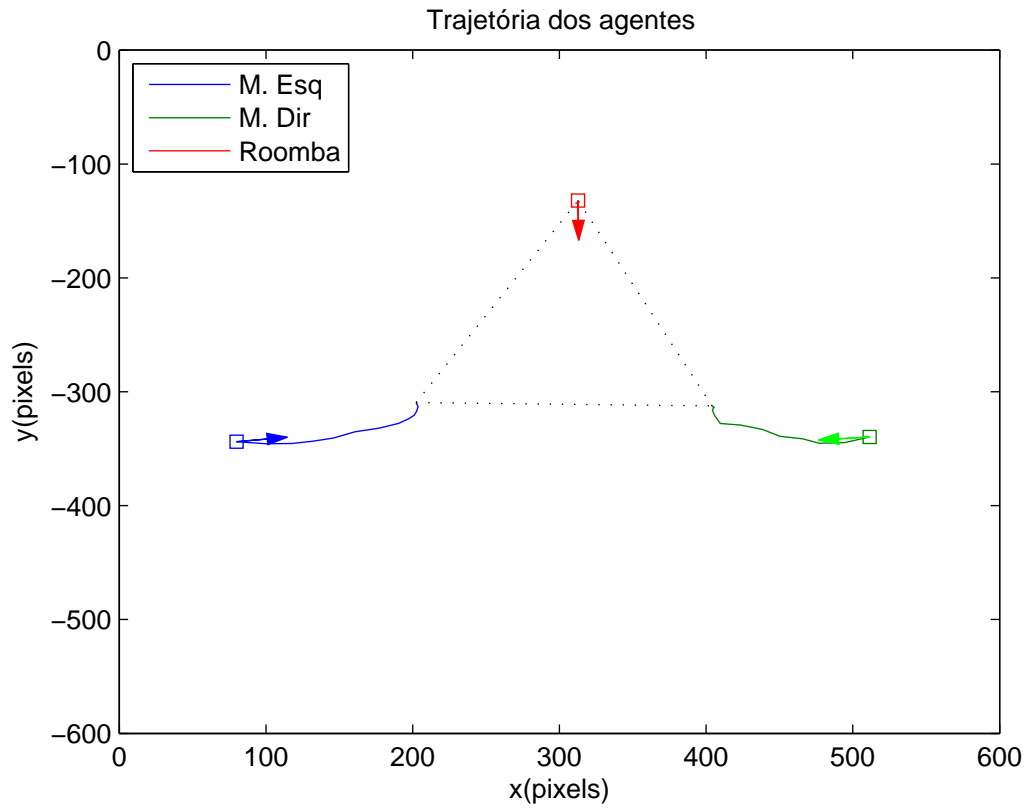


Figura 4.26: Trajetória dos agentes - M. Esq./Dir.

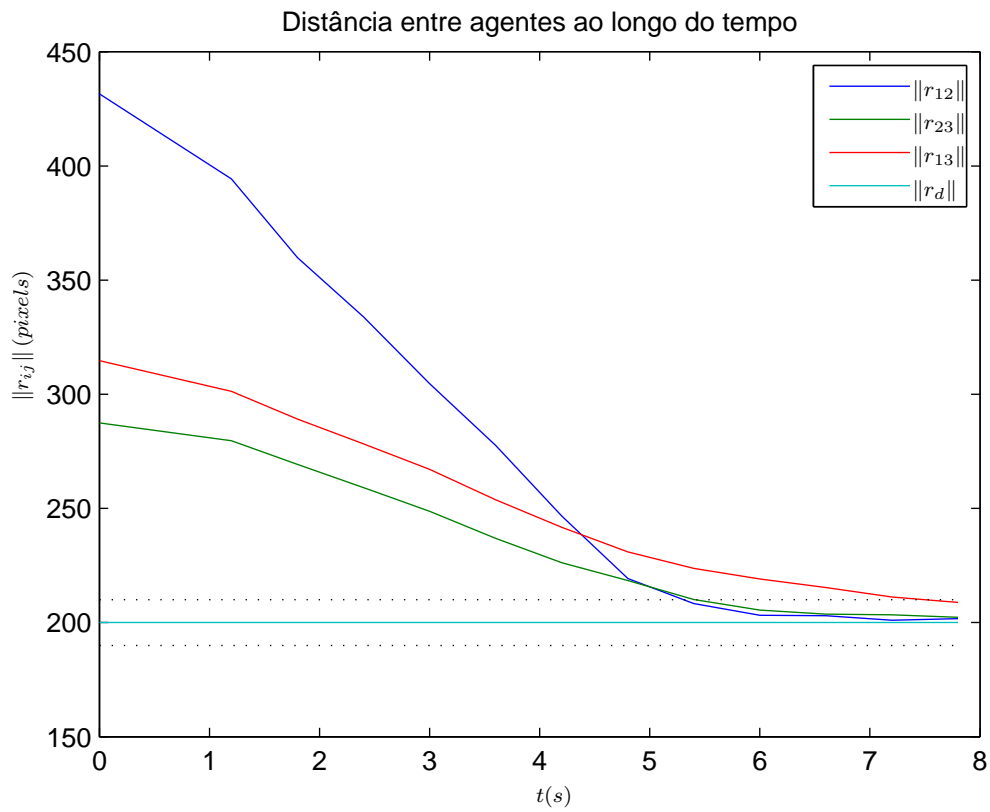


Figura 4.27: Distância entre agentes - M. Esq./Dir.



Figura 4.28: Posição e orientação final dos agentes - M. Esq./Dir.

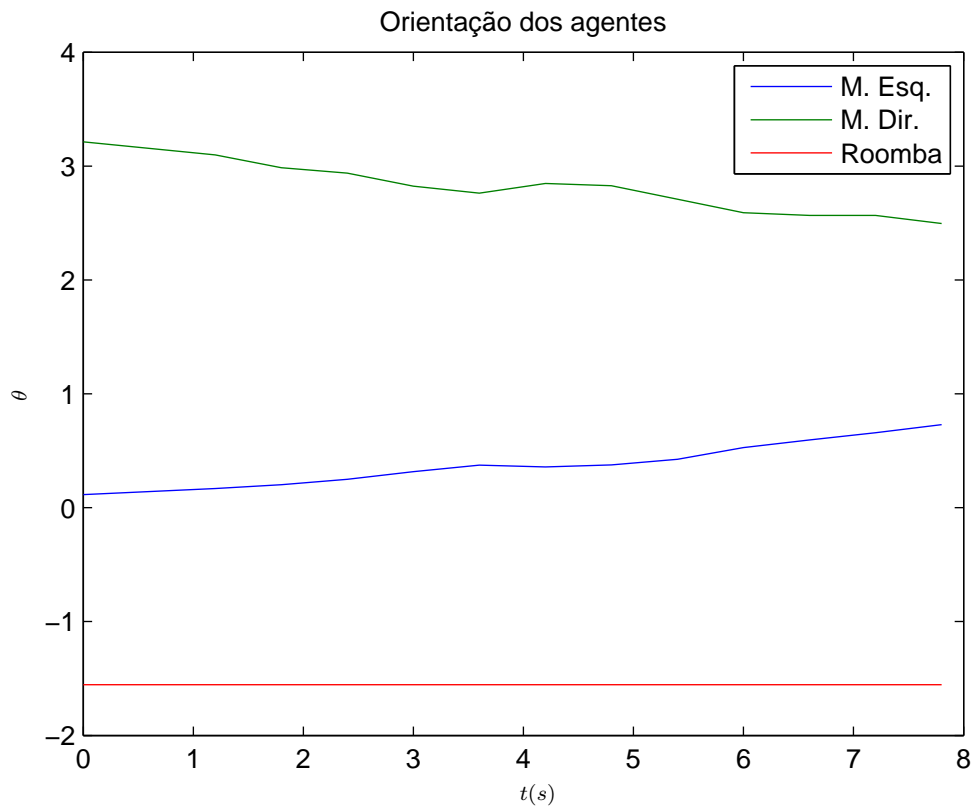


Figura 4.29: Orientação dos agentes - M. Esq./Dir.



Figura 4.30: Posição e orientação inicial dos agentes - M. Esq./Roomba

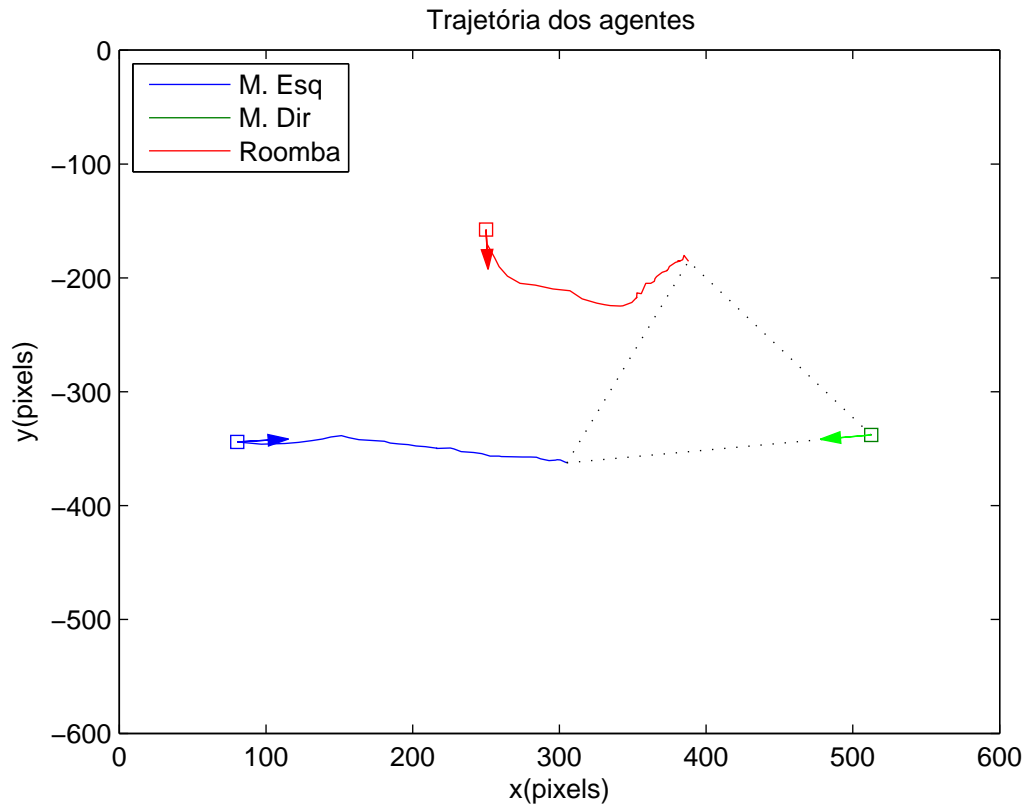


Figura 4.31: Trajetória dos agentes - M. Esq./Roomba

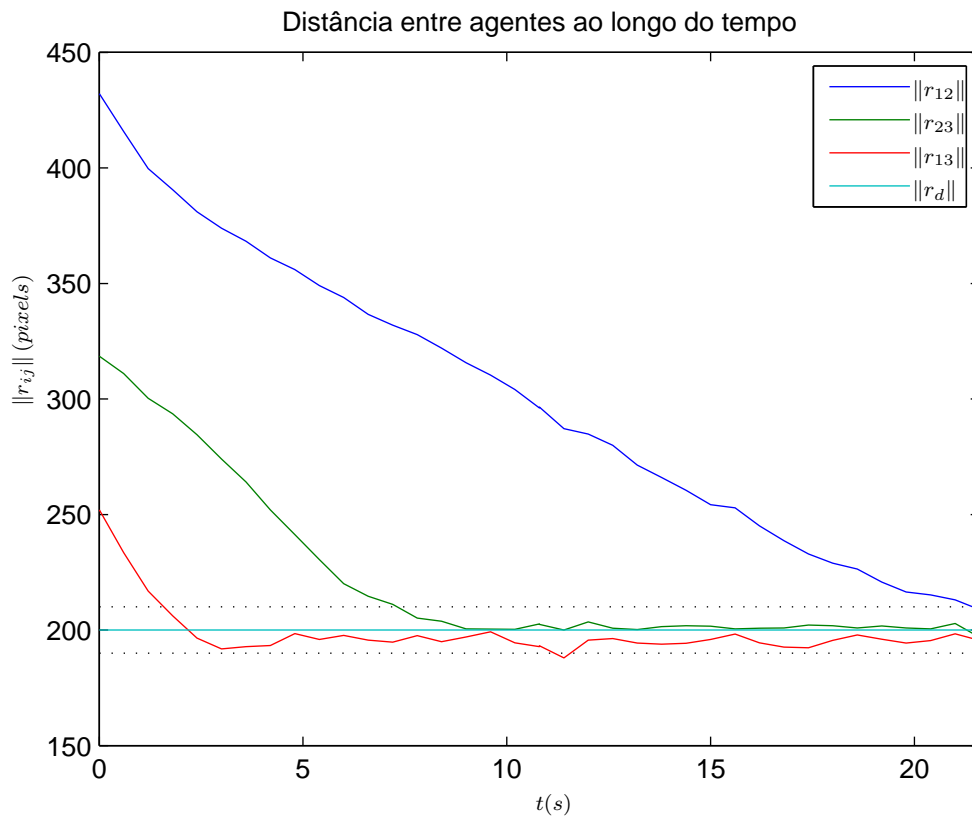


Figura 4.32: Distância entre agentes - M. Esq./Roomba



Figura 4.33: Posição e orientação final dos agentes - M. Esq./Roomba

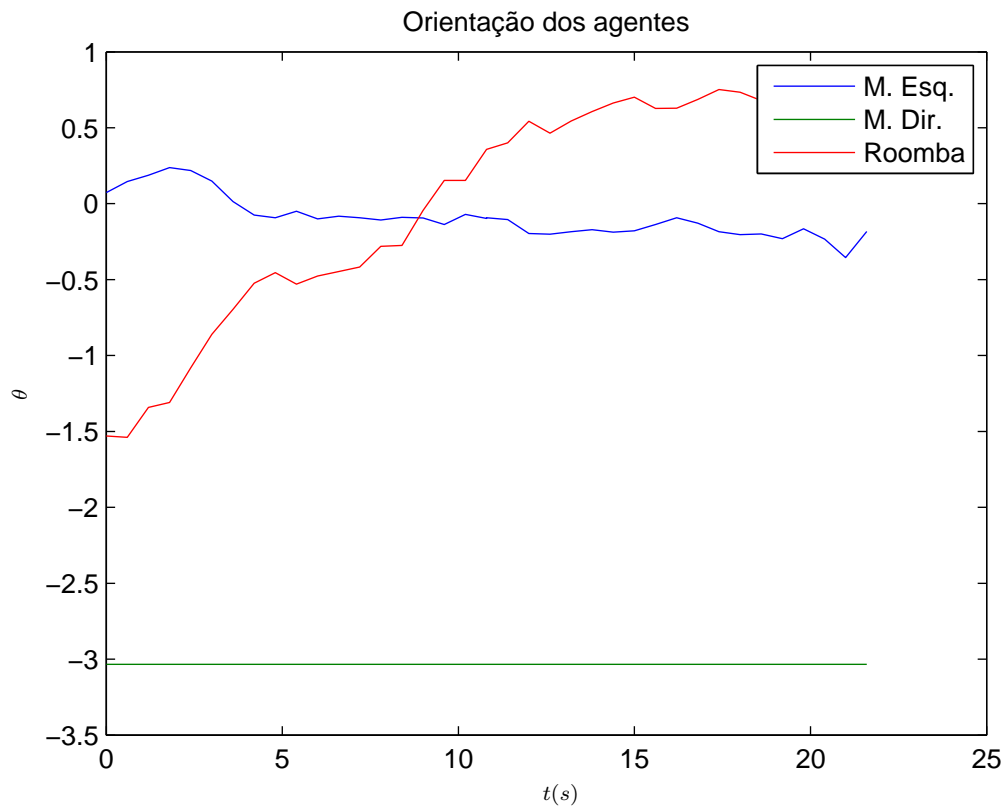


Figura 4.34: Orientação dos agentes - M. Esq./Roomba

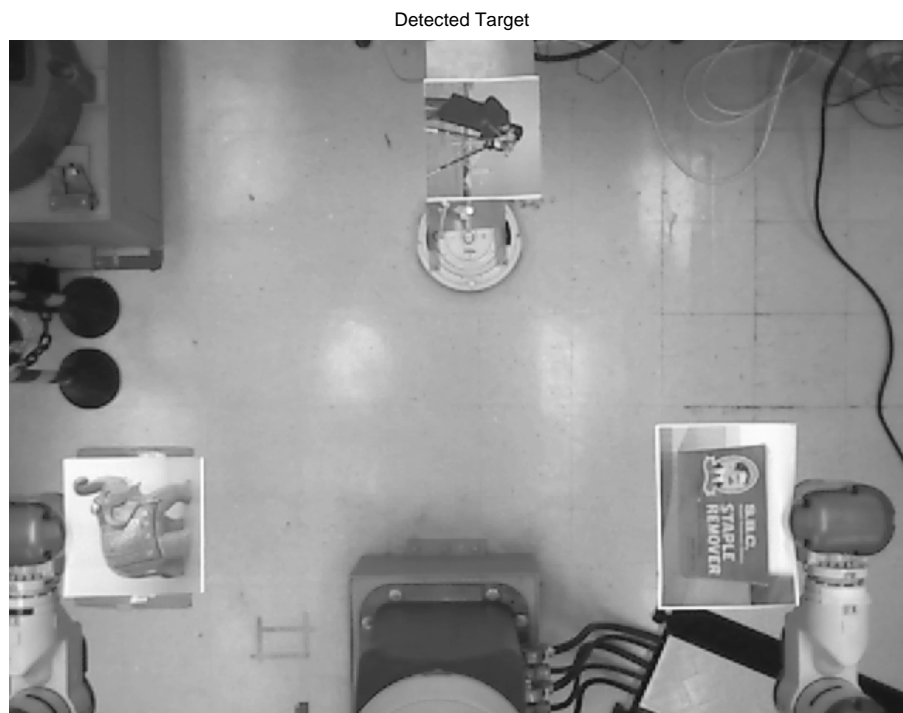


Figura 4.35: Posição e orientação inicial dos agentes - M. Dir./Roomba

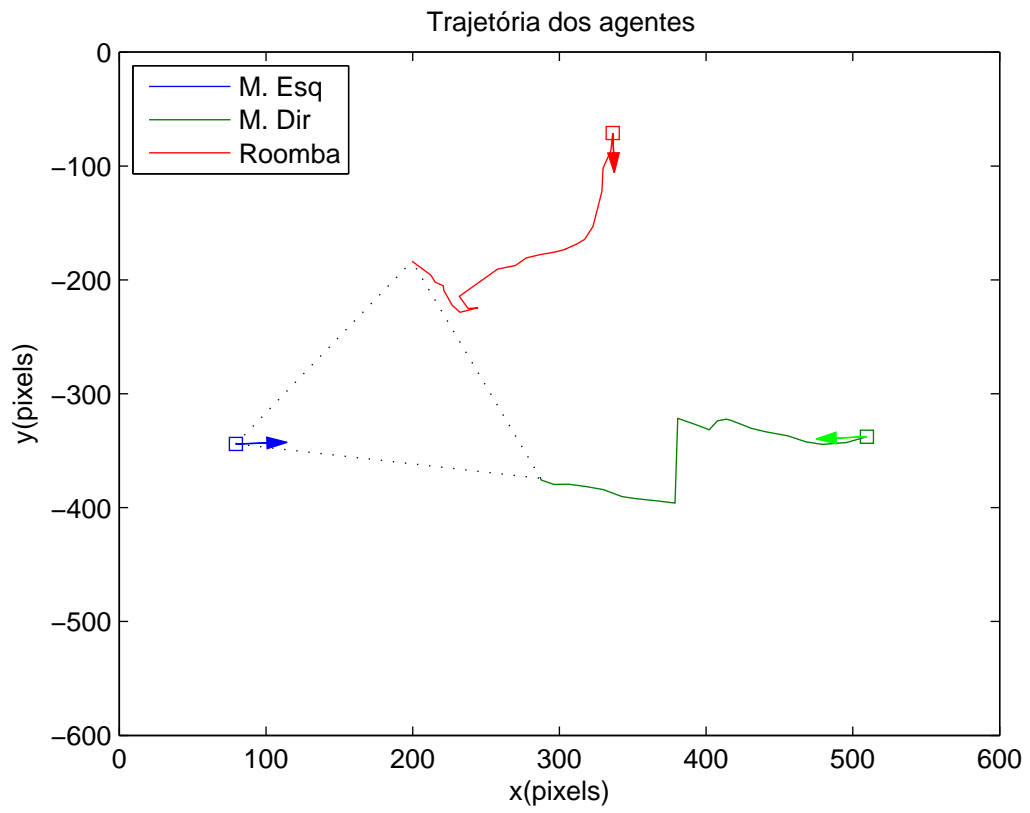


Figura 4.36: Trajetória dos agentes - M. Dir./Roomba

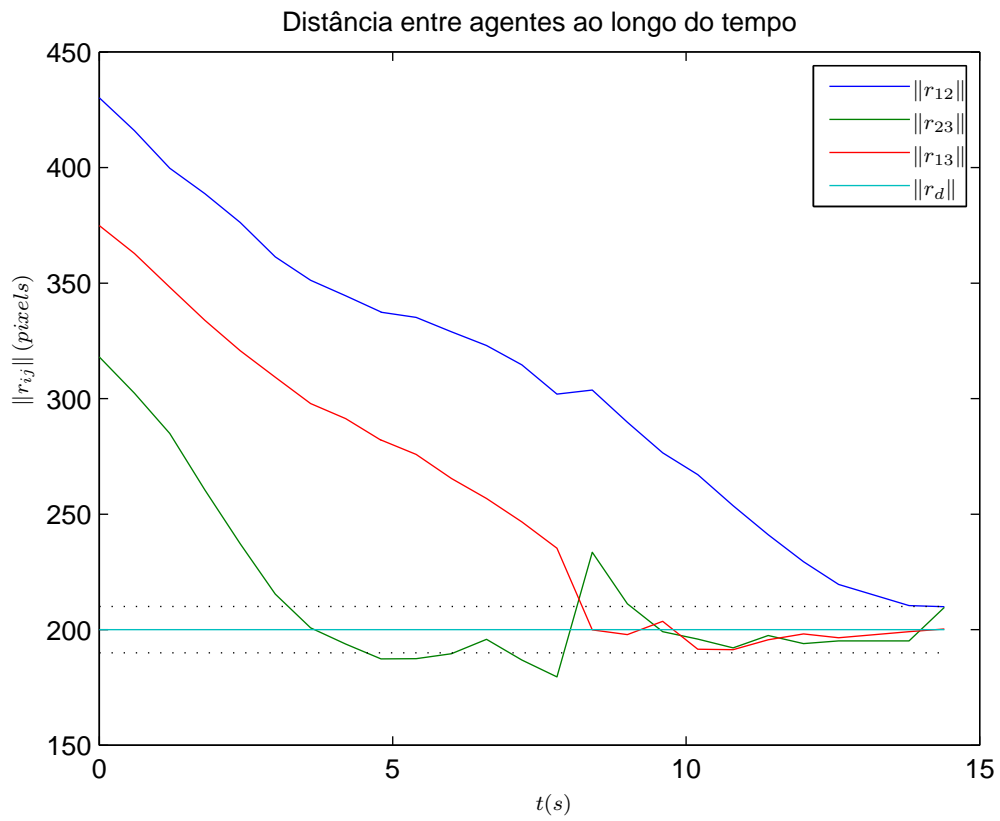


Figura 4.37: Distância entre agentes - M. Dir./Roomba



Figura 4.38: Posição e orientação final dos agentes - M. Dir./Roomba

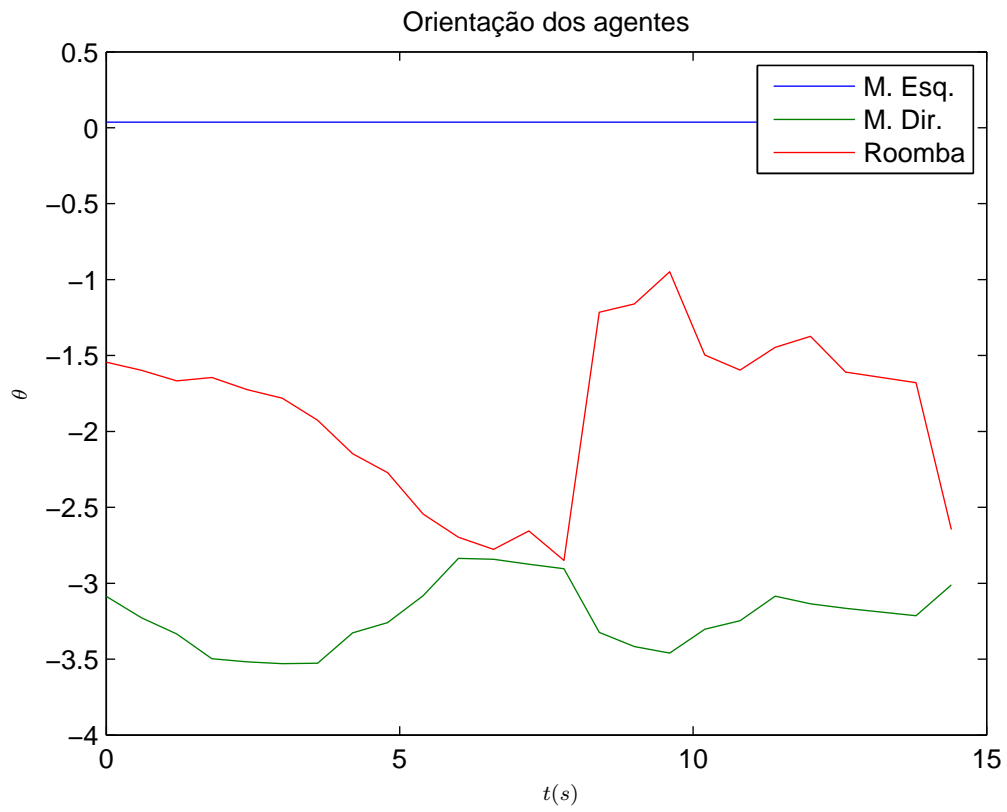


Figura 4.39: Orientação dos agentes - M. Dir./Roomba



Figura 4.40: Posição e orientação inicial dos agentes - M. Esq./M. Dir./Roomba

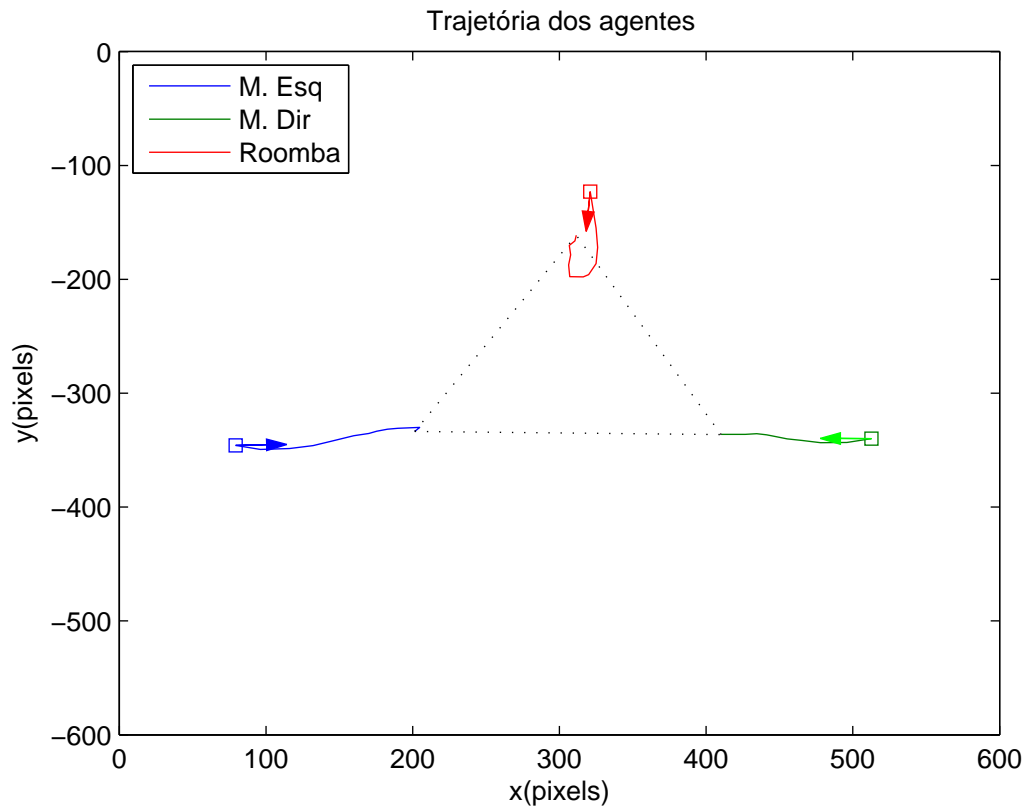


Figura 4.41: Trajetória dos agentes - M. Esq./M. Dir./Roomba

Observa-se a trajetória do sistema, na Figura (4.41) e sua orientação na Figura (4.44). Note que o roomba faz uma trajetória com curvatura mais fechada, trajetória qual seria mais difícil, até mesmo impossível, para qualquer dos manipuladores realizar. Embora curvas fechadas na trajetória em geral indiquem sobressinal, observa-se que os demais agentes não tem qualquer sinal de tal aproximação em suas trajetórias, diferente do que foi visto no experimento anterior (Figura 4.36). Vendo a Figura (4.42), tal sobressinal de fato não ocorre. Na verdade o roomba se move para abrir espaço para o manipulador direito se aproxime do esquerdo e entre em formação, análogo ao que aconteceu como ocorreu no caso Manipulador Esquerdo - Roomba (Figura 4.31).

Da mesma Figura (4.42), obtém-se que o tempo que o sistema leva para alcançar a formação é de 6.6 segundos. A disposição da configuração final no espaço de trabalho pode ser vista na Figura (4.43), com posição e orientação na tabela (4.15).

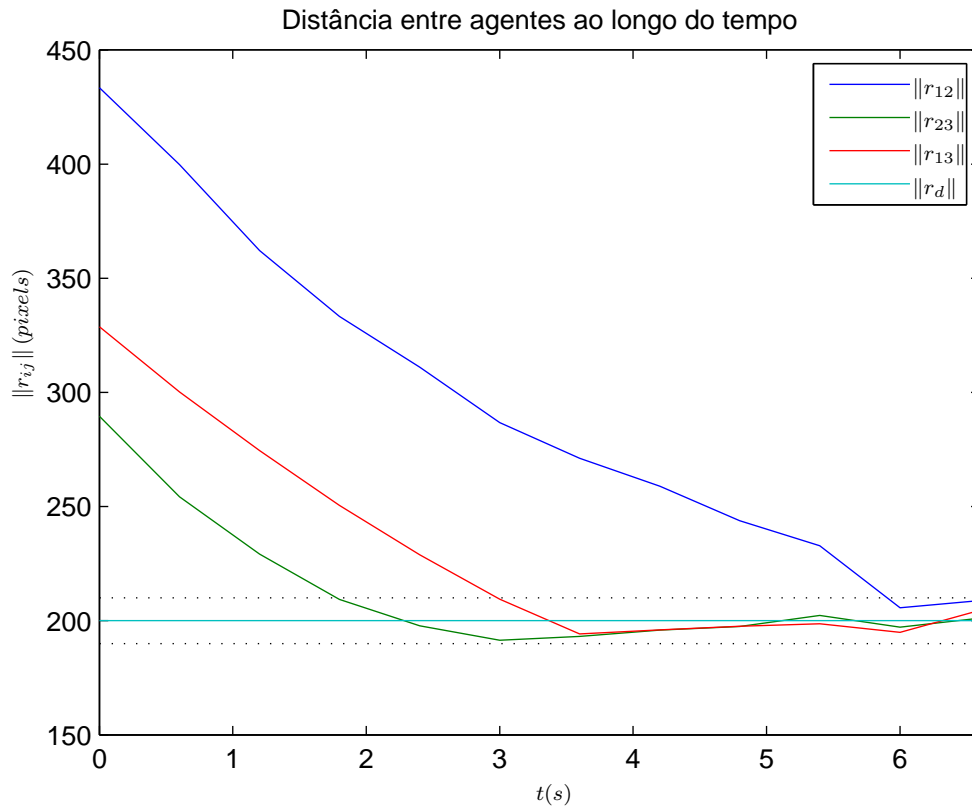


Figura 4.42: Distância entre agentes - M. Esq./M. Dir./Roomba

Tabela 4.15: Posição e orientação final dos agentes - M. Esq./M. Dir./Roomba

-	M. Esquerdo	M. Direito	Roomba
$x(t)$	$201.38pixels$	$410.20pixels$	$311.48pixels$
$y(t)$	$-333.64pixels$	$-336.45pixels$	$-161.22pixels$
$\theta(t)$	-0.2289	-2.9025	1.8022

Detected Target

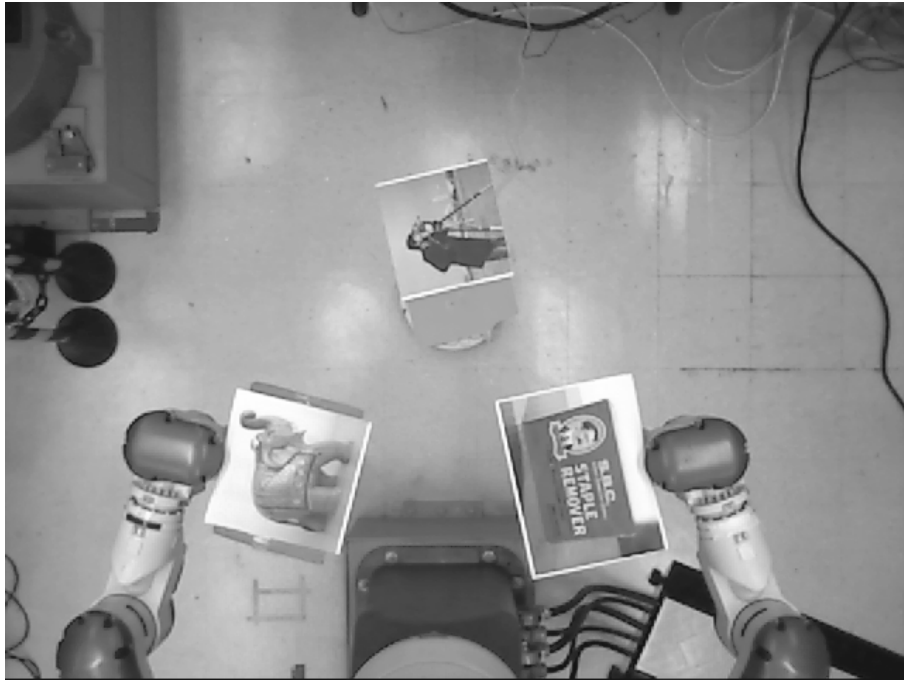


Figura 4.43: Posição e orientação final dos agentes - M. Esq./M. Dir./Roomba

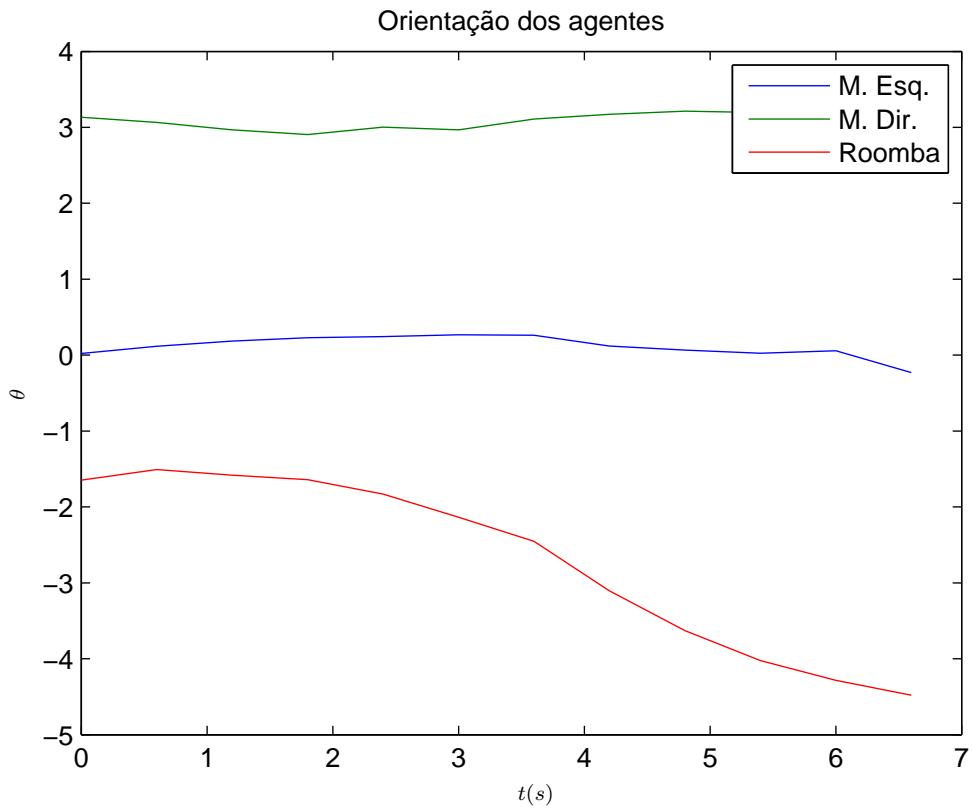


Figura 4.44: Orientação dos agentes - M. Esq./M. Dir./Roomba

Capítulo 5

Conclusões

Neste trabalho foi realizada a aplicação de uma lei de controle por função potencial saturada e deslocada para um conjunto de agentes não-holonômicos em um espaço de trabalho real. Até que se chegasse ao objetivo final, foram implementadas diversas etapas intermediárias. Foram modelados todos os elementos que constituem o sistema. O roomba como robô não-holonômico, os manipuladores o qual tiveram que ter seu movimento restringido para atuar como robôs não-holonômicos, a câmera que realiza a captura da posição e orientação dos agentes, e até mesmo a própria lei de controle para entender como a mesma atua sobre um sistema.

Com o modelo pronto, simulou-se exaustivamente o sistema em diversas condições. Alterou-se o número de agentes sob influência da lei de controle, a posição e orientação inicial dos agentes, a rotação da câmera, e praticamente todos os resultados mostram a perfeita atuação da lei de controle no sentido de garantir a formação desejada de forma estável. E mesmo os problemas que surgiram não foram causados por falha de controle, projetado para robôs não-holonômicos do tipo unicycle, mas em geral pela restrição não-holonômica imposta aos manipuladores, que já tem espaço de trabalho limitado. Tais problemas foram resolvidos com uma escolha arbitrária de parâmetros e posições iniciais adequadas, obtidas com grande número de simulações realizadas, até que foram obtidas trajetórias adequadas ao sistema, com todas as suas limitações. Esse fato mostra o quão importante foi realizar todas as simulações, antes de levar o controle ao caso real propriamente dito.

Por último, de posse de posições e parâmetros razoavelmente adequados, obtidos com as simulações, resta levar o controle ao espaço de trabalho real. Os resulta-

dos apresentados condizem com aquilo que era esperado, visto as simulações, até mesmo quando sugerida uma trajetória com curvas mais abruptas, que foi o caso do experimento com o Manipulador Direito e Roomba controláveis. Um resultado interessante foi mostrado com o último experimento, no qual todos os três agentes estavam sob ação da lei de controle, onde, apesar de a trajetória do roomba ter sofrido uma mudança de orientação de π radianos em relação à sua orientação original, foi visto que o módulo de suas distâncias ao longo do tempo não apresenta variações após alcançar a formação. O que significa que todo o movimento do roomba foi feito acompanhando o manipulador direito em toda sua trajetória. Em suma, o controle proposto guia os agentes a formação de forma estável, com ressalva para movimentos impossíveis no espaço de trabalho limitado.

5.1 Trabalhos Futuros

Visto que neste trabalho, a formação é obtida no plano da imagem, uma continuação natural para futuros projetos seria levar em consideração o modelo da câmera e fazer com que a formação agora seja feita no plano real, e não no plano da imagem. Outra importante idéia, vem do problema em gerar uma trajetória adequada para o sistema. Sendo o problema o fato de que o controle foi gerado para robôs não-holonômicos, seria extremamente interessante, em um próximo trabalho, fazer alterações no controle de modo que ele conte com as restrições de espaço de trabalho e movimento, não havendo mais necessidade de planejamento de trajetória. Por último, ainda há de se analisar a diferença entre a posição do padrão-alvo e o roomba, como mostrado no capítulo anterior. Seria interessante analisar, num trabalho futuro, os efeitos que este erro tem sobre o sistema, e como sumir com tais erros.

Referências Bibliográficas

- [1] SCHARF, D., HADAEGH, F., PLOEN, S. “A survey of spacecraft formation flying guidance and control. Part II: control”. In: *American Control Conference, 2004. Proceedings of the 2004*, v. 4, pp. 2976–2985 vol.4, June 2004.
- [2] REN, W., BEARD, R. “A decentralized scheme for spacecraft formation flying via the virtual structure approach”. In: *American Control Conference, 2003. Proceedings of the 2003*, v. 2, pp. 1746–1751, June 2003. doi: 10.1109/ACC.2003.1239847.
- [3] CAI, H., HUANG, J. “The leader-following consensus of multiple rigid spacecraft systems”. In: *American Control Conference (ACC), 2013*, pp. 824–829, June 2013. doi: 10.1109/ACC.2013.6579938.
- [4] SERRANI, A. “Robust coordinated control of satellite formations subject to gravity perturbations”. In: *American Control Conference, 2003. Proceedings of the 2003*, v. 1, pp. 302–307 vol.1, June 2003. doi: 10.1109/ACC.2003.1238957.
- [5] MARTIN, M., KLUPAR, P., KILBERG, S., et al. “Techsat 21 and revolutionizing space missions using microsattellites”, *Fifteenth AIAA/USU*, 2001.
- [6] STILWELL, D., BISHOP, B. “Platoons of underwater vehicles”, *Control Systems, IEEE*, v. 20, n. 6, pp. 45–52, Dec 2000. ISSN: 1066-033X. doi: 10.1109/37.887448.
- [7] QUINTERO, S. A. P., COLLINS, G. E., HESPANHA, J. P. “Flocking with Fixed-Wing UAVs for Distributed Sensing: A Stochastic Optimal Control Approach”. In: *American Control Conference (ACC), 2013*, June 2013.

- [8] ZHANG, M., LIU, H. “Formation flight of multiple fixed-wing unmanned aerial vehicles”. In: *American Control Conference (ACC), 2013*, pp. 1614–1619, June 2013. doi: 10.1109/ACC.2013.6580066.
- [9] BENDER, J. “An overview of systems studies of automated highway systems”, *Vehicular Technology, IEEE Transactions on*, v. 40, n. 1, pp. 82–99, Feb 1991. ISSN: 0018-9545. doi: 10.1109/25.69977.
- [10] WANG, Y., ZENGJIN, H. “Stability of an automated vehicle platoon”. In: *American Control Conference, 1998. Proceedings of the 1998*, v. 2, pp. 950–954 vol.2, Jun 1998. doi: 10.1109/ACC.1998.703548.
- [11] HOWARD, A., PARKER, L. E., SUKHATME, G. S. “Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection”, *The International Journal of Robotics Research*, v. 25, n. 5-6, pp. 431–447, 2006.
- [12] LI, S.-M., BOSKOVIC, J., SEEREERAM, S., et al. “Autonomous hierarchical control of multiple unmanned combat air vehicles (UCAVs)”. In: *American Control Conference, 2002. Proceedings of the 2002*, v. 1, pp. 274–279 vol.1, 2002. doi: 10.1109/ACC.2002.1024816.
- [13] REYNOLDS, C. W. “Flocks, Herds and Schools: A Distributed Behavioral Model”, *SIGGRAPH Comput. Graph.*, v. 21, n. 4, pp. 25–34, ago. 1987. ISSN: 0097-8930. doi: 10.1145/37402.37406. Disponível em: <<http://doi.acm.org/10.1145/37402.37406>>.
- [14] DAI, Y., LEE, S.-G. “The leader-follower formation control of nonholonomic mobile robots”, *International Journal of Control, Automation and Systems*, v. 10, n. 2, pp. 350–361, 2012. ISSN: 1598-6446. doi: 10.1007/s12555-012-0215-x. Disponível em: <<http://dx.doi.org/10.1007/s12555-012-0215-x>>.
- [15] NECOARA, I., CLIPICI, D., OLARU, S. “Distributed model predictive control of leader-follower systems using an interior point method with efficient computations”. In: *American Control Conference (ACC), 2013*, pp. 1697–1702, June 2013. doi: 10.1109/ACC.2013.6580080.

- [16] LEE, D., VISWANATHAN, S., HOLGUIN, L., et al. “Decentralized guidance and control for spacecraft formation flying using virtual leader configuration”. In: *American Control Conference (ACC), 2013*, pp. 4826–4831, June 2013. doi: 10.1109/ACC.2013.6580585.
- [17] CAO, Z., TAN, M., WANG, S., et al. “The optimization research of formation control for multiple mobile robots”. In: *Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on*, v. 2, pp. 1270–1274 vol.2, 2002. doi: 10.1109/WCICA.2002.1020786.
- [18] REN, W. “Consensus strategies for cooperative control of vehicle formations”, *Control Theory & Applications, IET*, v. 1, n. 2, pp. 505–512, 2007.
- [19] GOUVÊA, J. A. *Controle de formação de robôs não-holonômicos com restrição de curvatura usando função potencial*. Tese de Doutorado, COPPE, UFRJ, Rio de Janeiro, 2011.
- [20] BALCH, T., ARKIN, R. “Behavior-based formation control for multirobot teams”, *Robotics and Automation, IEEE Transactions on*, v. 14, n. 6, pp. 926–939, Dec 1998. ISSN: 1042-296X. doi: 10.1109/70.736776.
- [21] YUN, X., ALPTEKIN, G., ALBAYRAK, O. “Line and circle formation of distributed physical mobile robots”, *Journal of Robotic Systems*, v. 14, n. 2, pp. 63–76, 1997. ISSN: 1097-4563. doi: 10.1002/(SICI)1097-4563(199702)14:2<63::AID-ROB2>3.0.CO;2-R. Disponível em: <[http://dx.doi.org/10.1002/\(SICI\)1097-4563\(199702\)14:2<63::AID-ROB2>3.0.CO;2-R](http://dx.doi.org/10.1002/(SICI)1097-4563(199702)14:2<63::AID-ROB2>3.0.CO;2-R)>.
- [22] MONTEIRO, S., BICHO, E. “A dynamical systems approach to behavior-based formation control”. In: *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, v. 3, pp. 2606–2611, 2002. doi: 10.1109/ROBOT.2002.1013624.
- [23] ANTONELLI, G., ARRICHIELLO, F., CHAKRABORTI, S., et al. “Experiences of formation control of multi-robot systems with the Null-Space-based Behavioral Control”. In: *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1068–1073, April 2007. doi: 10.1109/ROBOT.2007.363126.

- [24] REIF, J. H., WANG, H. “Social potential fields: A distributed behavioral control for autonomous robots”, *Robotics and Autonomous Systems*, v. 27, n. 3, pp. 171–194, 1999.
- [25] OLFATI-SABER, R., FAX, J., MURRAY, R. “Consensus and Cooperation in Networked Multi-Agent Systems”, *Proceedings of the IEEE*, v. 95, n. 1, pp. 215–233, Jan 2007. ISSN: 0018-9219. doi: 10.1109/JPROC.2006.887293.
- [26] REN, W., BEARD, R., ATKINS, E. “A survey of consensus problems in multi-agent coordination”. In: *American Control Conference, 2005. Proceedings of the 2005*, pp. 1859–1864 vol. 3, June 2005. doi: 10.1109/ACC.2005.1470239.
- [27] OLFATI-SABER, R., MURRAY, R. “Flocking with obstacle avoidance: cooperation with limited communication in mobile networks”. In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, v. 2, pp. 2022–2028 Vol.2, Dec 2003. doi: 10.1109/CDC.2003.1272912.
- [28] OKOLOKO, I. “Consensus based distributed motion planning on a sphere”. In: *American Control Conference (ACC), 2013*, pp. 6132–6137, June 2013. doi: 10.1109/ACC.2013.6580799.
- [29] ZHOU, B., LIN, Z. “Consensus of high-order multi-agent systems with input and communication delays-state feedback case”. In: *American Control Conference (ACC), 2013*, pp. 4027–4032, June 2013. doi: 10.1109/ACC.2013.6580456.
- [30] CAO, Y., CASBEER, D., SCHUMACHER, C. “Reaching consensus in the sense of probability”. In: *American Control Conference (ACC), 2013*, pp. 5415–5420, June 2013. doi: 10.1109/ACC.2013.6580684.
- [31] BEARD, R., LAWTON, J., HADAEGH, F. “A feedback architecture for formation control”. In: *American Control Conference, 2000. Proceedings of the 2000*, v. 6, pp. 4087–4091 vol.6, 2000. doi: 10.1109/ACC.2000.876990.
- [32] TAN, K.-H., LEWIS, M. “Virtual structures for high-precision cooperative mobile robotic control”. In: *Intelligent Robots and Systems '96, IROS 96, Pro-*

ceedings of the 1996 IEEE/RSJ International Conference on, v. 1, pp. 132–139 vol.1, Nov 1996. doi: 10.1109/IROS.1996.570643.

- [33] YUAN, J., TANG, G.-Y. “Formation control for mobile multiple robots based on hierarchical virtual structures”. In: *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pp. 393–398, June 2010. doi: 10.1109/ICCA.2010.5524078.
- [34] DONG, W., BEN GHALIA, M., CHEN, C., et al. “Teleoperation of a cluster of mobile robots subject to model uncertainty”. In: *American Control Conference (ACC), 2013*, pp. 6436–6441, June 2013. doi: 10.1109/ACC.2013.6580848.
- [35] DESAI, J., OSTROWSKI, J., KUMAR, V. “Controlling formations of multiple mobile robots”. In: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, v. 4, pp. 2864–2869 vol.4, May 1998. doi: 10.1109/ROBOT.1998.680621.
- [36] TAKAHASHI, H., NISHI, H., OHNISHI, K. “Autonomous decentralized control for formation of multiple mobile robots considering ability of robot”, *Industrial Electronics, IEEE Transactions on*, v. 51, n. 6, pp. 1272–1279, Dec 2004. ISSN: 0278-0046. doi: 10.1109/TIE.2004.837848.
- [37] JAFARIAN, M., DE PERSIS, C. “Exact formation control with very coarse information”. In: *American Control Conference (ACC), 2013*, pp. 3026–3031, June 2013. doi: 10.1109/ACC.2013.6580295.
- [38] QUINTERO, S., COLLINS, G., HESPANHA, J. “Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach”. In: *American Control Conference (ACC), 2013*, pp. 2025–2031, June 2013. doi: 10.1109/ACC.2013.6580133.
- [39] MAGEE, C., OBRADOVIC, B., FREGENE, K. “Toward energy-optimizing rotary wing MAV formations”. In: *American Control Conference (ACC), 2013*, pp. 25–30, June 2013. doi: 10.1109/ACC.2013.6579808.
- [40] NECOARA, I., CLIPICI, D., OLARU, S. “Distributed model predictive control of leader-follower systems using an interior point method with efficient compu-

- tations”. In: *American Control Conference (ACC), 2013*, pp. 1697–1702, June 2013. doi: 10.1109/ACC.2013.6580080.
- [41] YAN, J., BITMEAD, R. “Coordinated control and information architecture”. In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, v. 4, pp. 3919–3923 vol.4, Dec 2003. doi: 10.1109/CDC.2003.1271762.
- [42] LADINO, A., PATINO, D. “On the stability of predictive controllers for linear systems with variable time delays”. In: *American Control Conference (ACC), 2013*, pp. 3254–3259, June 2013. doi: 10.1109/ACC.2013.6580333.
- [43] LI, H., SHI, Y. “Distributed receding horizon control of constrained linear systems with communication delays”. In: *American Control Conference (ACC), 2013*, pp. 1703–1708, June 2013. doi: 10.1109/ACC.2013.6580081.
- [44] HAFEZ, A., MARASCO, A., GIVIGI, S., et al. “Encirclement of multiple targets using model predictive control”. In: *American Control Conference (ACC), 2013*, pp. 3147–3152, June 2013. doi: 10.1109/ACC.2013.6580315.
- [45] MURRAY, R. M., LI, Z., SASTRY, S. S. “A mathematical introduction to robotic manipulation”. CRC press, 1994.
- [46] TANNER, H., JADBABAIE, A., PAPPAS, G. “Stable flocking of mobile agents part II: dynamic topology”. In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, v. 2, pp. 2016–2021 Vol.2, Dec 2003. doi: 10.1109/CDC.2003.1272911.
- [47] *iRobot Roomba 500 Open Interface (OI) Specification*. iRobot, 2007.
- [48] RIBEIRO, G. C. D. M. *Teleoperação bilateral de múltiplos robôs aplicada ao transporte de carga*. Tese de Mestrado, COPPE, UFRJ, Rio de Janeiro, 2013.
- [49] BAY, H., TUYTELAARS, T., VAN GOOL, L. “Surf: Speeded up robust features”. In: *Computer vision–ECCV 2006*, Springer, pp. 404–417, 2006.
- [50] TORR, P. H., ZISSERMAN, A. “MLESAC: A new robust estimator with application to estimating image geometry”, *Computer Vision and Image Understanding*, v. 78, n. 1, pp. 138–156, 2000.