

Universidade Federal do Ceará
Centro de Tecnologia
Programa de Pós Graduação em Engenharia Elétrica



**Controle Preditivo Aplicado ao Seguimento de
Trajetória de Robô Móvel com Rodas**

Mariana Akeme Ogawa

Fortaleza
Abril de 2014

Mariana Akeme Ogawa

Controle Preditivo Aplicado ao Seguimento de Trajetória de Robô Móvel com Rodas

Dissertação submetida à Universidade Federal do Ceará como parte dos requisitos para a obtenção do grau de mestre em Engenharia Elétrica.

Orientador:
Prof. Dr. Bismark Claure Torrico

Fortaleza
Abril de 2014

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Ciências e Tecnologia

-
- O28c Ogawa, Mariana Akeme.
 Controle preditivo aplicado ao seguimento de trajetória de robô móvel com rodas / Mariana Akeme Ogawa. – 2014.
 86 f. : il., enc. ; 30 cm.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Elétrica, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2014.
 Área de Concentração: Automação e Controle.
 Orientação: Prof. Dr. Bismark Claure Torrico.
1. Controle preditivo. 2. Robôs - Sistemas de controle. 3. Robótica. I. Título.

CDD 621.3

MARIANA AKEME OGAWA

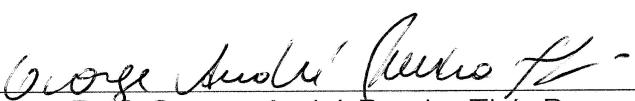
CONTROLE PREDITIVO APLICADO AO SEGUIMENTO DE TRAJETÓRIA
DE ROBÔ MÓVEL COM RODAS

Dissertação apresentada ao
Programa de Pós-Graduação em
Engenharia Elétrica da Universidade
Federal do Ceará, como requisito à
obtenção do título de Mestre em
Engenharia Elétrica. Área de
concentração: Sistema de Energia
Elétrica.

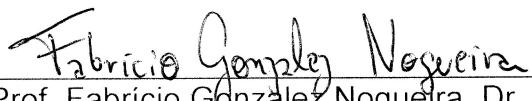
Aprovada em: 29/04/2014

BANCA EXAMINADORA


Prof. Bismark Claure Torrico, Dr. (Orientador)
Universidade Federal do Ceará (PPGEE-UFC)


Prof. George André Pereira Thé, Dr.
Universidade Federal do Ceará (DETI-UFC)


Prof. Paulo Peixoto Praça, Dr.
Universidade Federal do Ceará (PPGEE-UFC)


Prof. Fabrício González Nogueira, Dr.
Universidade Federal do Ceará (UFC)

Dedicatória

Este trabalho é dedicado à Deus, aos meus pais e meus irmãos.

Agradecimentos

Primeiramente a Deus, por tudo.

A minha mãe Maria de Fátima Alencar Ogawa por todo seu apoio, carinho e motivação sempre acreditando em mim, ao meu pai Oscar Ogawa pela sua dedicação. Aos meus irmãos Cynthia e Yukio por sempre estarem ao meu lado quando mais precisei.

Ao Prof. Dr. Bismark Claure Torrico, pela excelente orientação e enorme paciência. Ao Italo pelo projeto do robô móvel com rodas e ao Jehovah pela enorme ajuda na programação do robô, sem eles não seria possível este trabalho.

A todos os meus colegas do GPAR pelas ajudas, esclarecimentos, sugestões. Em especial: Vanier, J. Gleidson, Marcus, Antônio.

Aos meus amigos Juliana, André, Davi, David por todo seu apoio, compreensão e motivação durante esse percurso e aos outros que ficaram na torcida pelo sucesso desse projeto.

E a todas pessoas que por motivo de esquecimento não foram citadas anteriormente, vou deixando neste espaço minhas sinceras desculpas e os meus mais sinceros agradecimentos.

“Educação nunca foi despesa. Sempre foi investimento com retorno garantido”

Arthur Lewis

Resumo

Este trabalho propõe o estudo e aplicação de controladores avançados ao seguimento de trajetórias de robôs móveis com rodas. Este tipo de problema é bastante desafiador do ponto de vista de controle uma vez que o modelo tem duas entradas e três saídas, além disso, trata-se de um modelo não linear. Na literatura existem diversas soluções para o controle de trajetória de robôs móveis, dentre eles tem-se o Controle Preditivo Baseado em Modelo (MPC) por meio de modelos linearizados e um controlador não linear denominado neste trabalho de controlador de Klancar. Os controladores preditivos podem ser aplicados de forma eficiente em plantas com modelos multivariáveis, em situações na qual a trajetória futura de referência é conhecida e em sistemas com restrições nas variáveis de entrada e de saída. Porém, a principal desvantagem do MPC linearizado é o alto custo computacional o que limita as aplicações práticas. Além disso, esse controlador específico utiliza um modelo linearizado da planta. Por outro lado, o controlador de Klancar pode ser mais eficiente que os baseados em modelos lineares, devido às não linearidades inerentes do modelo. No entanto, a sua solução, por definição, não corresponde a critérios ótimos o que pode representar uma desvantagem principalmente em sistemas com restrições e referência futura conhecida. Além disso, neste trabalho é proposta a aplicação do controle preditivo EPSAC (*Extended Prediction Self Adaptive Control*) para o controle de seguimento de trajetórias. Esta estratégia utiliza de forma explícita o modelo não linear do robô, a referência futura, as restrições nas variáveis do robô e solução corresponde a um critério ótimo. Até onde foi pesquisado pelo autor deste trabalho, não existem relatos da utilização do EPSAC na robótica móvel, sendo desta forma uma aplicação inédita. Resultados de simulação são apresentados comparando os controladores estudados, utilizando índices de desempenhos. Além disso, os mesmos foram implementados em um robô móvel.

Número de páginas: 73

Palavras-chave:EPSAC, robô móvel, Controle Preditivo, Seguimento de trajetória.

Abstract

This work proposes a study and application of advanced controller to trajectory tracking of wheeled mobile robots. This kind of problem is a challenger for controllers once its models has two inputs and three outputs and is a non-linear model. In the literature there are various solutions to wheeled mobile robots trajectory tracking, among them the Model Predictive Control (MPC) with linearization model and a non linear control which in this work will be nominated as Klancar Controller. The Predictive Controllers can be applied efficiently in plants which has multiple inputs an multiple outputs, in situation that a future reference trajectory is known and systems with input and output constraints . However, the main disadvantage of MPC is the high computational effort which limits its practical application. Thus, this specific controller uses the plants linearization model. On the other hand, the Klancar Controller may be more efficient than the ones based on linear models, once the model is non linear. However, its solution, by definition, does not match the optimized criteria which can be a disadvantage mainly in systems that has constrains and a known future reference. Furthermore, this work proposes the application of the Predictive Control Extended Prediction Self Adaptive Control (EPSAC) to wheeled mobile robot trajectory tracking. This control strategy uses explicitly the non linear robot model, future reference, constraints on the variables and has a optimized solution. And, to the matter of this work, it has not been found reports of the EPSAC applied in mobile robotics, and is thus an unprecedented application. Simulation results are presented comparing the controllers studied using performance indices. Else, the controllers were applied in a mobile robot.

Number of pages:73

Keywords: EPSAC, mobile robot, Predictive Control, trajectory tracking .

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	1
1.1 Justificativa	4
1.2 Objetivos	4
1.3 Organização do trabalho	5
2 Noções de robótica móvel	6
2.1 Robôs móveis	6
2.2 Robôs móveis com rodas	8
2.2.1 Tarefas básicas de deslocamentos	10
2.2.2 Controle de um robô móvel	11
2.3 Modelo Cinemático do Robô	12
2.3.1 Modelo linearizado	13
2.3.2 Veículos com condução através de unidades diferenciais	14
2.4 Robô utilizado	16
3 Controladores para o seguimento de trajetória	18
3.1 Controlador Klancar	18
3.2 Controle Preditivo	20
3.2.1 Metodologia MPC	22
3.2.2 Elementos do MPC	24
3.2.3 Modelo de predição	25
3.2.4 Função objetivo	25
3.2.5 Obtenção da lei de controle	26
3.2.6 Respostas livre e forçada	27
3.3 Controlador MPC Linearizado	27
4 Extended Prediction Self Adaptive Control - EPSAC	32
4.1 Predição do $x(t+k t)$	33

4.2	Predições para $n(t + k t)$	33
4.3	Resposta base - resposta otimizada	34
4.4	EPSAC não-linear	39
4.5	MIMO EPSAC	40
4.6	EPSAC para um robô móvel com rodas não-holonômico	42
5	Resultados e discussão	45
5.1	Resultados de simulação	45
5.1.1	EPSAC	46
5.1.2	Controlador Klancar	49
5.1.3	MPC linearizado	50
5.1.4	Comparação entre os controladores	52
5.2	Aplicação no robô real	56
5.2.1	Controlador Klancar	57
5.2.2	EPSAC	58
5.2.3	MPC linearizado	61
5.2.4	Comparação entre os controladores	68
6	Conclusão	69
Referências Bibliográficas		71
Apêndice A – Resposta do controlador no Labview		74

Lista de Figuras

1.1	a)Robô Curiosity - NASA	b) Robô Quincy - Fukushima, Japão	2
1.2	a)Robô Rhino	b) Robô humanoide Asimo	2
1.3	a)Robô pessoal Aibo da Sony b) Robô educacional da NI	3	
2.1	Níveis de controle em um robô autônomo	7	
2.2	Sistema básico de controle	8	
2.3	Sistemas de coordenadas para o robô móvel	13	
2.4	Diagrama de blocos - mudanças de velocidades	14	
2.5	Diagrama de blocos - mudanças de velocidades	15	
2.6	Robô National Instruments utilizado	17	
3.1	Robô seguindo um erro de transformação	19	
3.2	Diagrama de Blocos para o controlador Klancar	20	
3.3	Estrutura básica do Controle Preditivo	21	
3.4	Estrutura básica para os controladores MPC	24	
3.5	Diagramas de Blocos MPC linearizado	28	
4.1	Diagrama de blocos Modelo Série-Paralelo	33	
4.2	Ilustração do conceito de resposta base e resposta ótima	36	
4.3	Efeito em um instante de tempo $t + k$ do sinal impulso aplicado no instante t . .	36	
4.4	Efeito em um instante de tempo $t + k$ do sinal impulso aplicado no instante $t + 1$. .	37	
5.1	simulação do WMR utilizando EPSAC a matriz de ponderação Q constante e o horizonte de controle N variando	48	
5.2	a) Erro para o $N = 5$ b) Erro para o $N = 10$	48	
5.3	simulação do WMR utilizando EPSAC a matriz de ponderação Q variando e o horizonte de controle N constante	49	
5.4	Simulação do WMR para o controlador Klancar variando os valores do ganho g . .	50	
5.5	Velocidades obtidas pelo WMR para o Klancar com o $g = 5$	51	
5.6	Simulação do WMR utilizando MPC com a matriz de ponderação Q constante e o horizonte de controle N variando	52	
5.7	Simulação do WMR utilizando MPC a matriz de ponderação Q variando e o horizonte de controle N constante	52	
5.8	Saída do controlador para o robô simulado - EPSAC, Klancar e MPC linearizado	54	

5.9	Valores das velocidades lineares e angulares para cada controlador analisado	55
5.10	Erros de posição e orientação	55
5.11	Erros de posição e orientação - detalhe	56
5.12	Robô no local de teste	58
5.13	Trajetória do robô com o controlador Klancar	59
5.14	Velocidade linear - Klancar	60
5.15	Velocidade angular - Klancar	60
5.16	Comparação entre os resultados da simulação e da implementação no robô para o controlador Klancar	61
5.17	Trajetória do robô - EPSAC	62
5.18	Velocidade linear para o robô utilizando controlador EPSAC	62
5.19	Velocidade linear para o robô utilizando controlador EPSAC	63
5.20	Velocidade angular para o robô utilizando controlador EPSAC	63
5.21	Erro de posição e orientação para o controlador EPSAC	64
5.22	Comparação entre os resultados das simulações e da implementação no robô para o controlador EPSAC	64
5.23	Trajetória do robô - MPC	65
5.24	Velocidade linear para o robô utilizando controlador MPC	65
5.25	Velocidade angular para o robô utilizando controlador MPC	66
5.26	Erro de posição e orientação para o controlador MPC	67
5.27	Comparação entre os resultados das simulações e da implementação no robô para o controlador MPC	67
5.28	Comparação entre os resultados da implementação no robô dos controladores Klancar, EPSAC e MPC	68
A.1	Tela do Labview para acompanhamento do deslocamento do robô	74

Lista de Tabelas

5.1 Índices de desempenho para cada controlador analisado	56
---------------------------------------------------------------------	----

1 Introdução

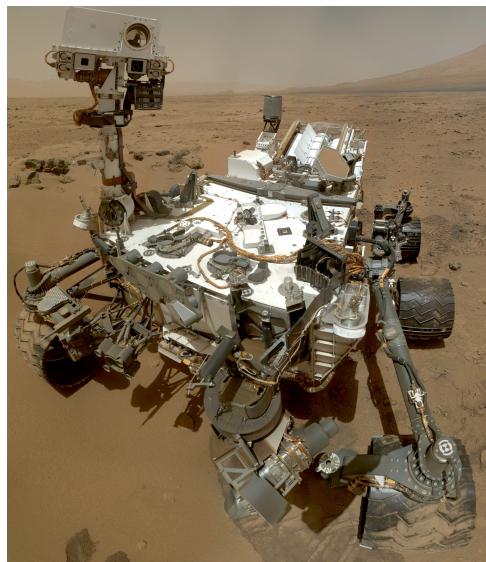
Desde a antiguidade os seres humanos demonstram interesses por máquinas que sejam semelhantes aos seres humanos, não só no aspecto físico, mas que possuam alguma capacidade coletar, interpretar informações sobre o ambiente que os rodeiam e a partir dessas informações sejam capazes de tomar alguma decisão; estas são as características de um robô. Podem-se dividir os robôs em dois grandes grupos: o primeiro grupo engloba os robôs manipuladores, amplamente utilizado nas indústrias realizando tarefas repetitivas, tais como soldar, pintar, apafusar entre outras (CRAIG, 2008). E no segundo grupos tem-se os robôs móveis, que possuem a capacidade de deslocar-se no espaço e esse deslocamento pode ser tanto terrestre, quanto aéreo ou mesmo submarino (SIEGWART; NOURBAKHSH, 2004).

Os robôs móveis podem ser utilizados para explorações de locais inacessíveis aos seres humanos, como por exemplo, ambientes submarinos, vulcões ou o espaço. Na Fig. 1.2(a) é apresentado o robô Curiosity desenvolvido pela NASA enviado a Marte para a realização de estudos (NASA, 2013). Também podem ser utilizado em locais que a proximidade de seres humanos pode ser prejudicial a saúde, tais como, em Nagatani et al. (2013) é apresentado os robôs utilizados para avaliar a situação do reator nuclear em Fukushima devido ao terremoto que atingiu o Japão em 11 de março de 2011 apresentado na Fig. 1.2(b), esses robôs são chamados de Quincy. E podem ser utilizados ainda em locais que correm o risco de acidentes, por exemplo no desarmamento de bombas, ou combate a incêndios.

Além dos robôs para executar tarefas perigosas, existem aqueles usados em ambientes domésticos e de serviços. O robô Rhino, por exemplo, Fig. 1.3(a) desenvolvido pela Universidade de Bonn na Alemanha servia como guia de um museu (BUHMANN *et al.*, 1995). Existe também o Ashimo, apresentado na Fig. 1.3(b) desenvolvido pelo empresa japonesa Honda, sendo o primeiro robô humanóide e a Sony desenvolveu um robô de uso pessoal quadrúpede semelhante a um cachorro, chamado Aibo, Fig. 1.4(a). Algumas empresas trabalham com robôs ou kits com peças para montar robôs utilizados na educação, como é o caso do robô NI LabVIEW Robotics Starter Kit, da National Instruments (NI) Fig. 1.4(b) utilizado nesse trabalho.

Os robôs móveis podem ser divididos de acordo com o ambiente na qual se deslocam. Os robôs móveis terrestres são subdivididos de qual forma esse deslocamento é realizado, dentre

Figura 1.1: a)Robô Curiosity - NASA



(a)

b) Robô Quincy - Fukushima, Japão



(b)

Fonte: (NASA, 2013)

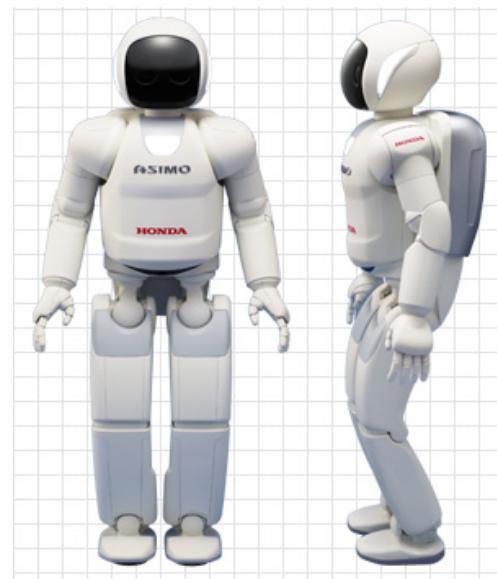
Fonte: (NAGATANI *et al.*, 2013)

Figura 1.2: a)Robô Rhino



(a)

b) Robô humanoide Asimo



(b)

Fonte: (BUHMANN *et al.*, 1995)

Fonte: (MOTOR, 2013)

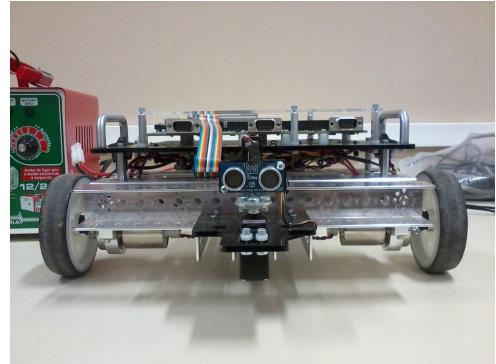
entre eles existem os que se deslocam com rodas. E estes são subdivididos em dois grupos: os omni-direcionais são aqueles que tem a capacidade de se deslocar livremente no plano e os não-holonômicos são os robôs que não podem se locomover em direções laterais em relação às rodas, tendo desta forma, restrições na sua locomoção.

O sistemas não-holonômicos estão sendo bastantes estudados, uma vez que são os robôs

Figura 1.3: a)Robô pessoal Aibo da Sony b) Robô educacional da NI



(a)



(b)

Fonte: (AIBO, 2013)

Fonte: Autor

móveis mais populares pela facilidade e o custo menor em relação a outras configurações. Considerando que o ambiente em que o robô se encontra não possua obstáculos, pode-se fazer com que ele se locomova considerando apenas o ponto inicial e o final, ou desejar que o mesmo siga uma trajetória pré-estabelecida.

Na literatura existem diversas propostas para o controle de trajetória de robôs móveis não-holônomicos com rodas (WMR), sendo que a grande maioria das propostas usam o modelo para o robô baseado nas equações da cinemática, sendo denominado modelo cinemático, tanto pela simplicidade quanto pelo fato da maioria dos robôs terem como parâmetro de entrada as velocidades linear e angular e não valores de aceleração ou torque (KLANCAR *et al.*, 2005). Porém, existem outras estruturas a serem utilizadas: Kanayama *et al* (2009) utiliza uma linearização da equação diferencial do sistema, já em Kim e Oh (1999) é utilizado o método de linearização entrada-saída. Em Kühne *et al* (2004) o modelo é linearizado levando em consideração o erro do robô de referência e o robô real.

Entre os controladores propostos na atualidade existem vários ramos de estudos, incluindo a utilização de inteligência artificial com propostas de utilização de lógica de fuzzy e redes neurais como o apresentado em Fierro e Lewis (1995), enquanto em Jang e Chung (2009) é proposto um controlador que combina lógica de fuzzy e redes neurais. Há propostas de controladores PI e PID tanto com estruturas lineares como não linear, em Al-Araji *et al* (2011) é proposto um controlador PID adaptativo, já em Klancar *et al* (2009) o controlador proposto possui características de um controlador PI com termos não-lineares. Em Secchi e Mut (2007) é proposto e analisado a estabilidade de um controlador discreto.

Alguns autores avaliam a utilização de algumas das diversas estratégias de controle predi-

tivo (*Model Predictive Control* - MPC). Em Kühne et al (2004) é utilizado um MPC sendo que o modelo utilizado foi linearizado, já em Hedjar et al (2009) o controlador preditivo calcula o erro previsto a uma amostra adiantada para determinar o esforço do controlador. Em Andrade e Torrico (2011) o controlador aplicado utiliza a metodologia Generalized Predictive Control (GPC).

Dentre as estratégias de controle adaptativo destaca-se *Extended Prediction Self Adaptive Control* (EPSAC) com diversas aplicações, tais como em Pop et. al (2012) o controlador EPSAC é aplicado no controle de um processo de separação de carbonos isótopos, este processo possui várias entradas e saídas, além da presença de atraso, características bastante comuns na indústria química. Até mesmo aplicações na medicina, por exemplo em Hodrea et. al (2008) um controlador EPSAC não linear (NEPSAC) é utilizado para controlar o nível de glicose em pacientes com diabetes tipo I, já em Brito et. al (2010) a estratégia de controle EPSAC é utilizada para controlar a temperatura e umidade de uma incubadora neonatal e em Méndez et. al (2011) é estuda a aplicação do controlador em processos de anestesia.

1.1 Justificativa

Até onde foi pesquisado para a realização desse trabalho, não foram encontrados trabalhos que utilizem o EPSAC para controlar a trajetória de um robô móvel com rodas. O EPSAC possui características que o tornam uma excelente opção para a utilização em aplicações referentes à robótica móvel, em especial no caso de controle de trajetória, destacam-se:

- A capacidade de se trabalhar diretamente com sistemas multivariáveis e com sistemas não-lineares.
- Utilização direta das restrições no algoritmo de controle

Desta forma, a contribuição deste trabalho consiste da aplicação do controlador preditivo EPSAC, para o controle de trajetória de um robô móvel com rodas não-holonômico.

1.2 Objetivos

Esse trabalho tem como principais objetivos:

- Estudo e análise de duas estruturas de controle: um Controle Preditivo que utiliza o modelo cinemático do robô linearizado, proposto por (KÜHNE *et al.*, 2004) e um controlador com termos não linear proposto por (KLANCAR *et al.*, 2005)

- Estudo da estratégia de controle preditivo do tipo EPSAC (CAMACHO; BORDONS, 1998) e a extensão do mesmo para a robótica móvel, especificamente o controle de trajetória.
- Estudo comparativo entre as estratégias de controle apresentadas.
- Aplicação dos controladores estudados em um robô móvel didático da National Instruments e análise dos dados obtidos experimentalmente.

1.3 Organização do trabalho

O primeiro capítulo é dedicado a introdução do trabalho. No segundo capítulo serão apresentado noções e definições sobre robótica móvel, além dos modelos utilizados e as características do robô no qual serão implementado as estratégias de controle. As estratégias de controle estudadas, retiradas da literatura, serão apresentadas no terceiro capítulo. O quarto capítulo será dedicado a apresentação das características do EPSAC com a sua utilização no controle de trajetória de um robô móvel com rodas. Em seguida, no quinto capítulo serão apresentados os resultados e discussão, tanto de simulação quanto os obtidos no robô da National Instruments. O sexto e último capítulo é dedicado as conclusões do trabalho.

2 Noções de robótica móvel

A robótica é uma área do conhecimento multidisciplinar abrangendo conceitos de mecânica, elétrica, computação, automação, inteligência computacional, dentre outras. Os primeiros robôs, os manipuladores industriais, são da década de 1960. Já os primeiros robôs móveis são de 1968 e consistiam principalmente de veículos teleguiados (Automated Guided Vehicles - AGVs), utilizados para transportar ferramentas em ambientes industriais e seguindo uma trajetória predefinida (GARCIA *et al.*, 2007).

2.1 Robôs móveis

Um robô móvel é um dispositivo mecânico montado sobre uma base não fixa, que age sob o controle de um sistema computacional, equipado com sensores e atuadores que o permitem interagir com o ambiente (PIERI, 2002). Esses dispositivos são capazes de movimentar-se no seu ambiente seja utilizando pernas ou rodas. Onde que os mais recentes são capazes de movimentar-se no ar ou embaixo da água. Para o caso específico dos robôs que utilizam rodas, o controle de posição, orientação e velocidade destes robôs são obtidos através de motores elétricos que movimentam as rodas (BEKEY, 2005).

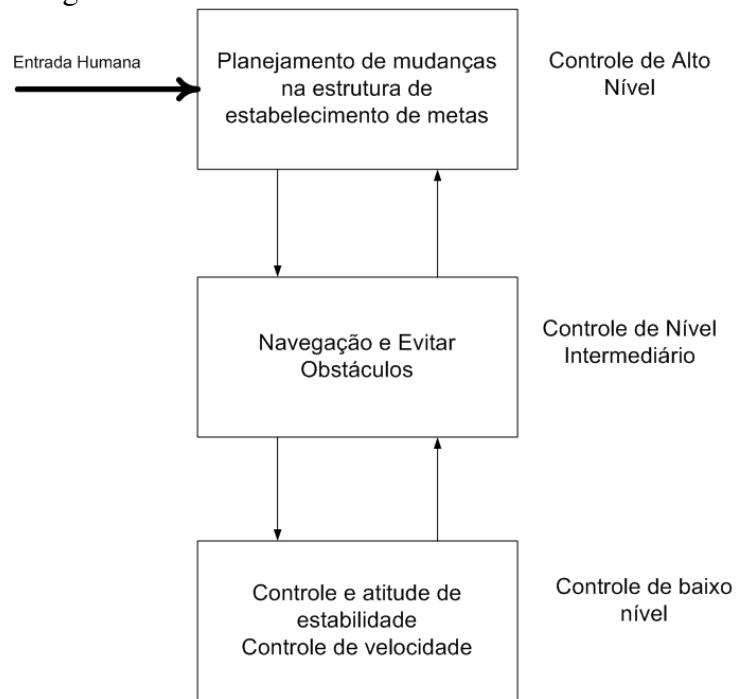
A interação do robô móvel com o ambiente que o cerca ocorre através de ciclos de percepção-ação. Para realizar esses ciclos um robô deve ter sensores, atuadores e capacidade de processamento que emula em alguns aspectos do conhecimento humano (BEKEY, 2005). Para realização de tarefas o robô realiza diversos ciclos de percepção-ação, modificando o estado do ambiente no qual ele se encontra. O ciclo de percepção-ação pode ser descrito em três passos fundamentais (PIERI, 2002):

- Os sensores atuam para obter informações acerca do ambiente;
- O processamento das informações obtidas através dos sensores e decisão de que ações deverão ser executadas;
- Os atuadores executam as ações planejadas.

Na robótica móvel pode-se considerar que existem basicamente três níveis de controle. No mais alto nível de controle é tomada de decisões, planejamento e mudança de metas. No nível mais “baixo” de controle é necessário que os motores de acionamento das rodas alcancem a velocidade desejada de acordo com os requisitos de projeto e limitações físicas da estrutura do robô. Enquanto o nível intermediário é necessário para projetar os robôs de uma forma que não colidam com outros robôs ou com obstáculos, enquanto ao mesmo tempo manter a estabilidade no nível mais baixo (BEKEY, 2005).

Os vários níveis de controle discutidos anteriormente são mostrados na Fig. 2.1. A organização do software é associado com os vários níveis é frequentemente denominado *arquitetura de controle* de um robô. Pode-se observar que os controles de alto nível geram entradas para os níveis mais baixos, mas também existe a realimentação dos níveis inferiores para os níveis mais superiores. Sensores disponibilizam entradas para os níveis inferiores (e algumas vezes de níveis intermediários) (BEKEY, 2005).

Figura 2.1: Níveis de controle em um robô autônomo



Fonte: (BEKEY, 2005)

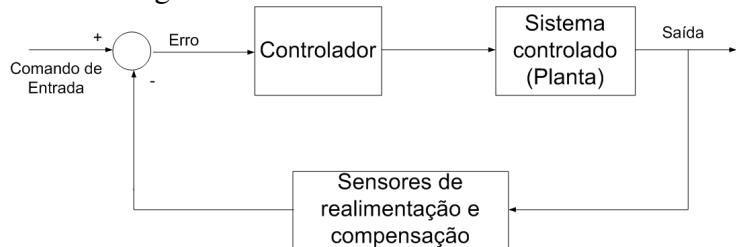
Observe-se que o bloco superior na 2.1 indica a entrada humana envolvida no alto nível de controle. O controle de baixo nível é claramente autônomo, enquanto o controle intermediário é normalmente autônomo nos robôs da atualidade, mas pode ainda envolver algumas entradas humanas (BEKEY, 2005).

O comando de entrada representado na Fig. 2.2 pode representar, por exemplo, a orientação desejada das rodas da frente, considerando que estas são as rodas motorizadas. O erro é a

diferença entre as direções de referência e a direção real. Este sinal de erro é o sinal de entrada do controlador, o qual um de sinal para os motores que irão movimentar as rodas. A direção das rodas é medida pelos sensores e comparada com os comandos de entrada. Sistemas desses tipos são conhecidos como sistemas de controle de realimentação negativa, desde que o sinal de realimentação tem um sinal oposto de comando de entrada.

Existem diversos tipos de robôs móveis, dentre eles existem aqueles que se deslocam em terra utilizando rodas. Esse tipo de robô é o utilizado nesse trabalho e será apresentado com maiores detalhes na sessão seguinte.

Figura 2.2: Sistema básico de controle



Fonte: (BEKEY, 2005)

2.2 Robôs móveis com rodas

Os robôs móveis com rodas são os mais populares entre os robôs móveis terrestres, pois são robôs com estruturas relativamente fáceis de construção mecânica, além de terem custo relativamente baixo comparando com os de esteiras e os que se movimentam através de pernas (PIERI, 2002). As rodas são muito populares não só como meio de locomoção na robótica, mas como nos veículos construídos pelos humanos de uma forma em geral, podendo atingir boas eficiências, e isso com uma implementação relativamente simples (BATLLE; BARJAU, 2009).

Os robôs móveis com rodas podem ser avaliados de acordo com a sua capacidade de manobras no plano de trabalho, sendo subdivididos em dois grupos nos quais é levado em consideração a capacidade de deslocamento: os chamados robôs móveis omnidirecional e os robôs móveis não - holonômicos (BATLLE; BARJAU, 2009) (FIGUEIREDO; JOTA, 2004).

Os robôs denominados de omnidirecional, são aqueles capazes de se movimentar em qualquer direção ao longo do plano do solo (x, y) independente da orientação que o robô encontre-se em relação ao seu eixo vertical. Para essa capacidade de manobras é necessário que as rodas possam se movimentar em mais de uma única direção, desta forma robôs omnidirecionais normalmente utilizam rodas Suecas, ou esféricas alimentadas (BATLLE; BARJAU, 2009).

Robôs não-holonômicos

O robô a ser controlado é um sistema não-holonômico. O termo holonômico é atribuído ao físico Hertz e tem como significado: “universal”, “integral”, “integrável”, sendo literalmente formado pelo prefixo holo que significa o todo, conjunto, totalidade e o sufixo -nomia que é sinônimo de lei. Desta forma, os sistemas não-holonômicos podem ser interpretados como sendo sistemas que não são integráveis (FIGUEIREDO; JOTA, 2004).

Os sistemas não-holonômicos são definidos como sendo os sistemas que possuem dimensão finita tendo algum tipo de restrição imposta a um ou mais estados do mesmo. Estas limitações podem ter diversas causas, seja ela a conservação do momento angular, ou condições impostas pela incapacidade de deslocamento em uma ou mais direções. O efeito da imposição dessas restrições é que durante o projeto do sistema de controle elas devem ser levadas em consideração, uma vez que o sistema não tem atuadores em todas as direções do espaço do problema (FIGUEIREDO; JOTA, 2004).

Apesar da limitação em seus movimentos, os mesmos possuem a capacidade de atingir qualquer configuração no espaço onde estão definidos, desde que estes sistemas sejam controláveis e a configuração atingíveis. Porém, devido as suas características as leis de controle não são simples nem fáceis de serem geradas de forma a garantir a estabilização do sistema; existindo a necessidade de utilização de ferramentas matemáticas mais elaboradas para análise e projeto, tais como a geometria diferencial, a utilização de um controle não-linear ou um controle linear variante no tempo (FIGUEIREDO; JOTA, 2004).

Para a melhoria do projeto de controle de sistemas não-holonômicos é necessário a consideração das restrições ao movimento uma vez que possibilita o projeto de controladores multivariáveis, exponencialmente estáveis de forma integrada. A análise e síntese desse tipo de controladores tem sido um desafio, tendo proporcionado um aumento da utilização das teorias de controle não linear (FIGUEIREDO; JOTA, 2004).

Na robótica o exemplo de um sistema não-holonômico seria um robô que se move com acionamento diferencial. Sendo utilizado nesse caso as mesmas estratégias de controle para sistemas não-holonômicos (FIGUEIREDO; JOTA, 2004).

Sendo uma classe de robôs que alcança uma alta capacidade de manobras, apenas um pouco inferior aqueles que possuem a configuração omnidirecional. Nesses robôs, movimentos em uma direção particular pode requerer uma movimentação rotacional inicial. Com um chassi circular e o eixo de rotação no centro desse robô, esse tipo de robô pode girar sem mudar a sua posição em relação ao solo. O mais popular desse tipo de robô é o de duas rodas com unidade de acionamento diferencial no qual as duas rodas rotacionam em torno de um ponto no centro do robô. Um ou dois pontos adicionais de contato com o solo podem ser utilizados para a

estabilidade, tal escolha é baseada em aplicações específicas (FIGUEIREDO; JOTA, 2004).

As duas rodas acionadas individualmente, também chamadas de rodas motorizadas, são as responsáveis pelo movimento do robô. As mudanças de orientação do robô ocorrem devido às diferenças de velocidade entre as mesmas. O robô possui a capacidade de rotacionar em torno do próprio eixo, na situação que cada roda gira em sentidos contrários, porém com mesmo módulo de velocidade. Geralmente, é considerado que o corpo do robô tem uma forma simétrica, as rodas motorizadas encontram-se equidistantes do eixo de simetria e o centro de massa está no centro geométrico do corpo (FIGUEIREDO; JOTA, 2004). Na continuidade será apresentado as tarefas básicas de deslocamento para esse tipo de robô.

2.2.1 Tarefas básicas de deslocamentos

Existem duas tarefas básicas de deslocamento para um robô móvel com rodas (WMR - do inglês Wheeled Mobile Robot) quando se encontra em um ambiente no qual não existem obstáculos (ORIOLO *et al.*, 2002):

- Movimento ponto a ponto : o robô deve atingir uma configuração desejada a partir de uma configuração inicial dada.
- Seguimento ou Rastreamento de trajetória: o ponto de referência do robô deve acompanhar uma trajetória no plano cartesiano iniciando em uma configuração inicial dada.

Para realizar essas tarefas pode ser utilizados comandos de pré-alimentação, controle com realimentação ou uma combinação dos dois. Na realidade, as soluções de realimentação apresenta um grau intrínseco de robustez. Entretanto especialmente no caso de deslocamento ponto a ponto, o projeto das leis de controle de realimentação para os sistemas não - holonômicos tem diversas obstruções estruturais. Controladores que superam essas dificuldades podem levar ao desempenho não satisfatório no regime transitório. O projeto de comandos de realimentação é estritamente relacionado ao planejamento de trajetória, e essas soluções levam em consideração a especificidades cinemática da natureza de um robô móvel não-holonômicos (LUCA *et al.*, 2001).

Quando utilizando uma estratégia de realimentação, o deslocamento ponto a ponto leva a um problema denominado estabilização de postura. Sem perdas de generalidade, o objetivo pode ser definido como a origem de uma configuração espacial n-dimensional do robô. Enquanto para o seguimento de trajetória, na presença de um erro inicial, como por exemplo, um início fora da trajetória, o problema de controle de seguimento assintótico consiste na estabilização em zero no $e_p = (e_x, e_y)$, erro cartesiano bidimensional em relação a posição do movimento do robô de referência (LUCA *et al.*, 2001). O objetivo desse trabalho é o controle

para o seguimento de uma trajetória pré-definida. Existem dois tipos de controle nessa situação, o controle cinemático e o dinâmico, que serão apresentados na próxima sessão.

2.2.2 Controle de um robô móvel

É possível levantar os modelos cinemáticos e dinâmicos de um robô móvel com rodas. Para a determinação do modelo cinemático é utilizado álgebra vetorial e matricial (KLANCAR *et al.*, 2005), para representar e descrever a localização de um objeto no espaço tridimensional a partir das equações básicas da cinemática em relação a um sistema de referência. O modelo dinâmico relaciona-se com as forças que atuam no corpo (PIERI, 2002), entre elas o torque e incluem propriedades dinâmicas, tais como a inércia e a massa do robô.

Enquanto o controle cinemático encarrega-se de controlar a trajetória selecionada, porém características dinâmicas do robô, tais como inércia, atritos e folgas impedem que a trajetória de referência e a trajetória real coincidam. O modelo dinâmico do robô é altamente não linear, multivariável, acoplado e de parâmetros variáveis, seu controle é extremamente complexo.

Controle cinemático

O controle cinemático é baseado nos modelos cinemáticos dos sistemas, que podem ser de primeira ordem ou de ordem mais alta. O modelo cinemático de primeira ordem é amplamente utilizado na literatura e na prática por diversas razões, entre elas tem-se (KLANCAR *et al.*, 2005):

1. A dinâmica do sistema normalmente é negligenciável, uma vez que os motores são rápidos e robustos, em especial a velocidades moderadas.
2. O design do robô raramente permite a utilização dos valores do torque ou da aceleração como sendo entrada do sistema.
3. A entrada de um robô móvel normalmente são as velocidades de referências, em alguns casos a linear e angular, em outros equipamentos são as velocidades, seja ela linear ou angular, de cada roda.

A partir do modelo escolhido e das tarefas desejadas que o robô realize é escolhido a estratégia de controle mais adequada que resultem uma otimização. Quando é desejado que o robô percorra uma determinada trajetória através de tempo é utilizado o controle cinemático. Os dados de entrada do controlador são obtidos através de informações provindas do usuário, tais como: o ponto de destino ou a trajetória desenhada. E a partir do modelo cinemático do robô são estabelecidas as velocidades de referência.

Será utilizado neste trabalho o controle cinemático, uma vez que o objetivo é que o robô siga uma trajetória pré-definida, tendo como parâmetros de referências as velocidades lineares e angulares.

2.3 Modelo Cinemático do Robô

O modelo matemático do robô foi determinado a partir das equações da cinemática direta aplicadas ao robô, levando em consideração que o mesmo possui um motor diferencial para cada roda e desta forma o comando de velocidade é realizado individualmente em cada roda motorizada. Para a determinação das equações é considerado que o centro de massa coincide com o centro geométrico do robô. O modelo é dado pela seguinte relação (KLANCAR *et al.*, 2005):

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.1)$$

onde v e ω são a velocidade linear e a angular, respectivamente e também as entradas do sistema $u \triangleq [v \ \omega]^T$. O vetor $x \triangleq [x, y, \theta]^T$ contém os valores de posição e orientação do robô, o ponto médio do eixo das rodas, em relação a um sistema de coordenadas iniciais global $\{O, X, Y\}$, conforme a Fig. 2.3. Reescrevendo a Eq. 2.1 de forma concisa torna-se a seguinte expressão: (KÜHNE *et al.*, 2004)

$$\dot{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.2)$$

As relações apresentadas são para as diferenciais da posição e orientação. Para escrever as rotinas computacionais do Controle Preditivo do tipo EPSAC é necessário um modelo em tempo discreto, dando os valores de posição e orientação a cada instante. Este mesmo modelo é utilizado para o controle PI não-linear, proposto por (KLANCAR *et al.*, 2005), que será apresentado na Seção 3.1. Para a determinação do controlador é necessário a posição e orientação no instante de tempo t , os valores disponíveis são as velocidades linear v e angular ω desta forma o modelo é discretizado utilizando as aproximações de Euler, e as relações para a posição e orientação do robô são as seguinte (ANDRADE; TORRICO, 2011):

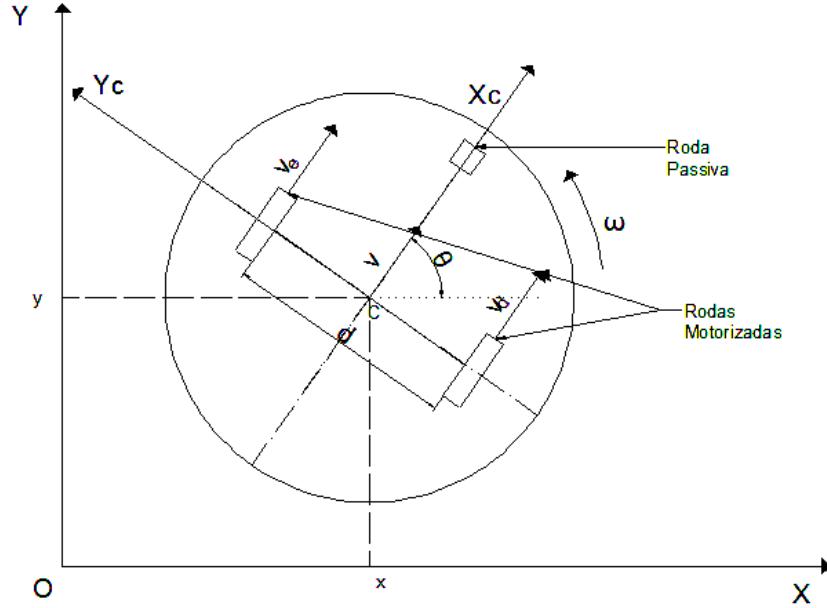
$$x(t+1) = x(t) + T_s v(t) \cos(\theta(t)) \quad (2.3)$$

$$y(t+1) = y(t) + T_s v(t) \sin(\theta(t)) \quad (2.4)$$

$$\theta(t+1) = \theta(t) + T_s \omega(t) \quad (2.5)$$

onde T_s é o período de amostragem.

Figura 2.3: Sistemas de coordenadas para o robô móvel



Fonte: Autor

2.3.1 Modelo linearizado

O controlador proposto por (KÜHNE *et al.*, 2004), apresentado na seção 3.3 utiliza um modelo linearizado. Esse modelo é obtido utilizando o cálculo da diferença dos erros de posição e orientação do robô real em relação ao de referência, considerando que o robô de referência é descrito pela mesma Eq. 2.2, no qual a trajetória é descrita por \mathbf{x}_r e o sinal de controle \mathbf{u}_r , todos os termos que caracterizam o robô de referência terão com subscrito o índice r (KÜHNE *et al.*, 2004) .

$$\dot{\mathbf{x}}_r = \mathbf{f}(\mathbf{x}_r, \mathbf{u}_r) \quad (2.6)$$

Expandindo o termo a direita da Eq. 2.2 utilizando a Série de Taylor em relação ao ponto $(\mathbf{x}_r, \mathbf{u}_r)$ e descartando os termos de ordem mais altas, tem-se que (KÜHNE *et al.*, 2004):

$$\dot{\mathbf{x}} = f(x_r, u_r) + \frac{\partial f(x, u)}{\partial x} \bigg|_{\substack{x=x_r \\ u=u_r}} (x - x_r) + \frac{\partial f(x, u)}{\partial u} \bigg|_{\substack{x=x_r \\ u=u_r}} (u - u_r) \quad (2.7)$$

reescrevendo de forma simplificada:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_r, \mathbf{u}_r) + \mathbf{f}_{x,r}(\mathbf{x} - \mathbf{x}_r) + \mathbf{f}_{u,r}(\mathbf{u} - \mathbf{u}_r) \quad (2.8)$$

onde os termos $\mathbf{f}_{x,r}$ e $\mathbf{f}_{u,r}$ são jacobianos de \mathbf{f} levando em consideração \mathbf{x} e \mathbf{u} , respectivamente, estimados em torno do ponto de referência $(\mathbf{x}_r, \mathbf{u}_r)$. Subtraindo a Eq. 2.6 da Eq. 2.8, resulta em

(KÜHNE *et al.*, 2004):

$$\dot{\tilde{\mathbf{x}}} = \mathbf{f}_{\mathbf{x},\mathbf{r}}\tilde{\mathbf{x}} + \mathbf{f}_{\mathbf{u},\mathbf{r}}\tilde{\mathbf{u}} \quad (2.9)$$

sendo $\tilde{\mathbf{x}} \triangleq \mathbf{x} - \mathbf{x}_r$ o termo referente ao erro do robô real em relação ao robô de referência, e o termo $\tilde{\mathbf{u}} \triangleq \mathbf{u} - \mathbf{u}_r$ é associado a perturbação da entrada no controle do sistema (KÜHNE *et al.*, 2004). Utilizando diferenças avançadas para a aproximação de \dot{x} , tem-se o seguinte modelo em tempo discreto:

$$\tilde{\mathbf{x}}(\mathbf{k} + 1) = \mathbf{A}(\mathbf{k})\tilde{\mathbf{x}}(\mathbf{k}) + \mathbf{B}(\mathbf{k})\tilde{\mathbf{u}}(\mathbf{k}) \quad (2.10)$$

sendo:

$$\mathbf{A}(k) = \begin{bmatrix} 1 & 0 & -v_r(k) \sin \theta_r(k) T_s \\ 0 & 1 & v_r(k) \cos \theta_r(k) T_s \\ 0 & 0 & 1 \end{bmatrix}$$

e :

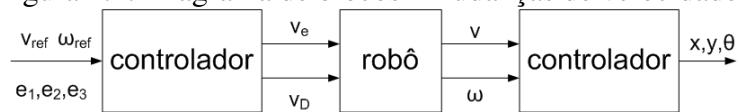
$$\mathbf{B}(k) = \begin{bmatrix} \cos \theta_r(k) T_s & 0 \\ \sin \theta_r(k) T_s & 0 \\ 0 & T_s \end{bmatrix}$$

2.3.2 Veículos com condução através de unidades diferenciais

Para os WMR existem dois tipos: os veículos com condução pelas rodas dianteiras e os com condução através de unidades diferenciais. As equações cinemáticas são desenvolvidas com propriedades básicas geométricas do movimento alcançado (COOK, 2011). Como o robô utilizado é um com unidades diferenciais nas rodas traseiras, o primeiro tipo não será abordado neste trabalho.

A entrada do robô utilizado são as velocidades lineares das rodas direita (v_d) e esquerda (v_e) já os controladores trabalham com as velocidades lineares e angulares v e ω desta forma é necessário a conversão de v e ω para v_d e v_e e depois retornar para v e ω conforme o apresentado na Fig. 2.4.

Figura 2.4: Diagrama de blocos - mudanças de velocidades



Fonte: Autor

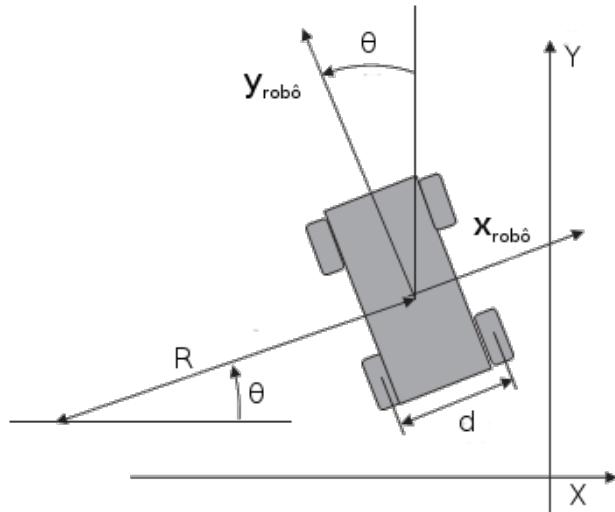
Neste tipo de WMR as rodas de um lado do robô são controladas independentemente. Coordenando as duas velocidades diferentes, pode-se fazer diferentes tipos de movimentos, tais

como: girar em torno de si mesmo, movimentar-se em linha reta, movimentar-se em um caminho circular ou acompanhar uma trajetória pré-definida (COOK, 2011).

As equações de movimento para um robô conduzido através de rodas com unidades diferenciais, serão definidas a seguir. Seja R o raio instantâneo de curvatura da trajetória de robô. d A largura do veículo (o espaçamento entre as rodas), como o apresentado na Fig. 2.5. Lembrando que:

$$v = \omega R \quad (2.11)$$

Figura 2.5: Diagrama de blocos - mudanças de velocidades



Fonte: (COOK, 2011)

A distância da roda esquerda para o eixo de coordenadas é $R_e = R - d/2$, enquanto para a roda direita é $R_d = R + d/2$. Desta forma a velocidade linear da roda esquerda:

$$v_e = \omega(R - W/2) \quad (2.12)$$

e da roda direita é:

$$v_d = \omega(R + W/2) \quad (2.13)$$

Subtraindo a Eq. 2.13 de Eq. 2.12, resulta em:

$$v_d - v_e = \omega d \quad (2.14)$$

então a velocidade angular do robô é:

$$\omega = \frac{v_d - v_e}{d}. \quad (2.15)$$

Resolvendo para o raio de curvatura instantâneo, tem-se que:

$$R = \frac{v_e}{\omega} + \frac{d}{2}$$

ou:

$$R = \frac{v_e}{\frac{v_r - v_l}{d}} + \frac{d}{2} = \frac{d}{2} \frac{v_d + v_e}{v_d - v_e}$$

Isto resulta na seguinte expressão para velocidade em relação ao eixo longitudinal do robô, o eixo paralelo a direção das rodas:

$$v_{yrob\hat{o}} = \omega R = \frac{v_d - v_e}{d} \frac{d}{2} \frac{v_d + v_e}{v_d - v_e} = \frac{v_d + v_e}{2} \quad (2.16)$$

Como não existe a movimentação em direções laterais as rodas do robô:

$$v_{xrob\hat{o}} = 0, \quad (2.17)$$

e

$$\omega_{rob\hat{o}} = \frac{v_d - v_e}{d}. \quad (2.18)$$

Convertendo para coordenadas globais, tem-se as seguintes expressões:

$$\dot{x} = -\frac{v_d + v_e}{2} \sin \theta, \quad (2.19)$$

$$\dot{y} = \frac{v_d + v_e}{2} \cos \theta, \quad (2.20)$$

e

$$\omega = \frac{v_r - v_l}{d}. \quad (2.21)$$

Pode-se ser levado em consideração que as velocidades não podem mudar instantaneamente. Desta forma, é introduzido como variáveis de controles as taxas de velocidades (COOK, 2011):

$$\dot{v}_d = u_1 \quad (2.22)$$

e

$$\dot{v}_e = u_2 \quad (2.23)$$

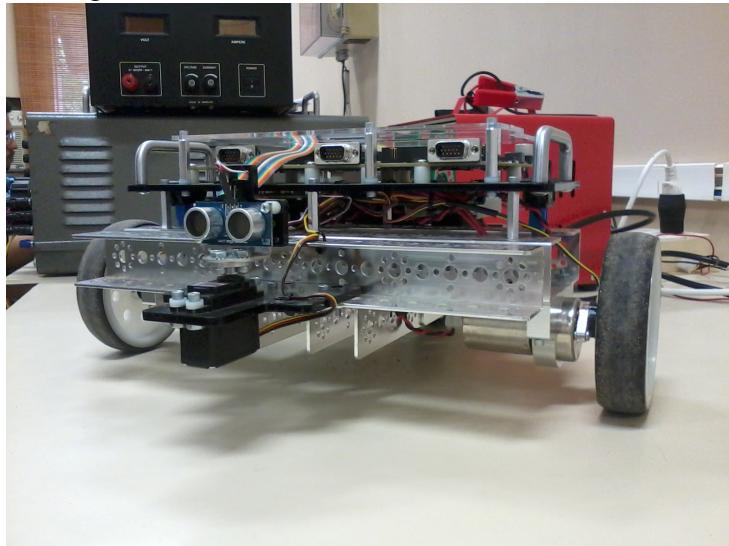
2.4 Robô utilizado

O robô móvel utilizado para a realização dos experimentos, foi um equipamento da empresa National Instruments, modelo NI LabVIEW Robotics Starter Kit ®, mostrado na Fig. 2.6.

O robô possui as seguintes características (INSTRUMENTS, 2012):

- 2 motores Pitsco Education de 12 V, cada um com 152 rpm de rotação e um torque

Figura 2.6: Robô National Instruments utilizado



Fonte: Autor

aproximado de 2.1 N.m (300 oz-in);

- Encoder ótico com 400 pulsos por revolução;
- Sensor de distância ultrassônico, com alcance que varia entre 0,02 até 3 m.
- Duas rodas motorizadas com diâmetro de 4 polegadas.
- Uma roda omni direcional que é utilizada para o direcionamento do robô.

A programação do robô é baseada na ferramenta:LabVIEW Robotics. Existem bibliotecas disponíveis, ou pode ser feito um algoritmo próprio utilizando o *software* Labview. Já vem incorporado ao equipamento um algoritmo de desvio de obstáculos, um dispositivo de controle de velocidades das rodas e um dispositivo de aquisição de dados em tempo real.

O programa utilizado para comunicação robô-computador é o Labview. O Labview tem uma interface gráfica semelhante a um diagrama de blocos, existem bibliotecas, contendo funções e também blocos prontos de sensores e atuadores, que são incluídas na instalação básica dos programas e outras a partir de Download do sítio na Internet da National Instruments, também existe a disponibilidade de diagramas de blocos completos. É possível fazer a simulação do comportamento do robô a partir do Labview, e também criar um bloco no qual contenha linhas de códigos próprias para o Mathscript.

3 Controladores para o seguimento de trajetória

Existem na literatura diversas propostas de controladores para promover o seguimento de trajetória de um robô móvel. O objetivo desse trabalho é verificar a possibilidade da utilização da estratégia de controle preditivo EPSAC para o controle de trajetória e comparar seu desempenho com dois controladores existentes na literatura. Um desses controladores escolhidos foi proposto por (KLANCAR *et al.*, 2005) e tem como característica a presença de termos não lineares e apenas um grau de liberdade no controlador, sendo denominados nesse trabalho de controlador Klancar em homenagem ao autor. O outro controlador escolhido foi proposto por (KÜHNE *et al.*, 2004) e utiliza uma estratégia de controle MPC, o qual trabalha com um modelo do robô linearizado em torno da trajetória de referência, sendo por este motivo denominado de MPC linearizado.

3.1 Controlador Klancar

A estratégia baseia-se no seguinte conceito: quando um robô é controlado para seguir um caminho de referência normalmente existem erros entre a posição e orientação do robô real e o determinado pela referência, sendo a posição e orientação ideal, como o mostrado na Fig. 3.1, os erros podem ser escritos da seguinte forma:

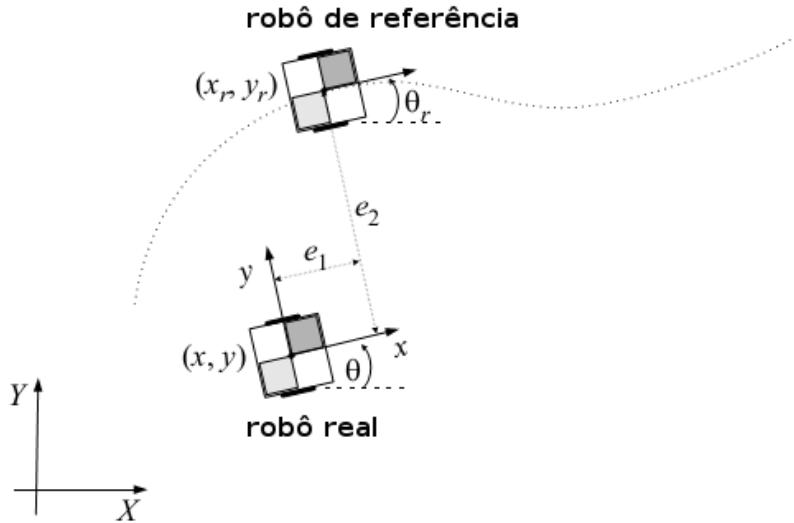
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (3.1)$$

Derivando a função erro, tem-se o seguinte modelo cinemático:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.2)$$

As entradas do robô podem ser expressa, a partir de uma transformação não linear, da seguinte forma:

Figura 3.1: Robô seguindo um erro de transformação

(KLANCAR *et al.*, 2005)

$$u_1 = v_r \cos e_3 - v$$

$$u_2 = \omega_r - \omega \quad (3.3)$$

O controlador em malha fechada é definido como sendo, na forma matricial:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -\text{sign}(v_r)k_2 & -k_3 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (3.4)$$

ou escrito de forma simplificada:

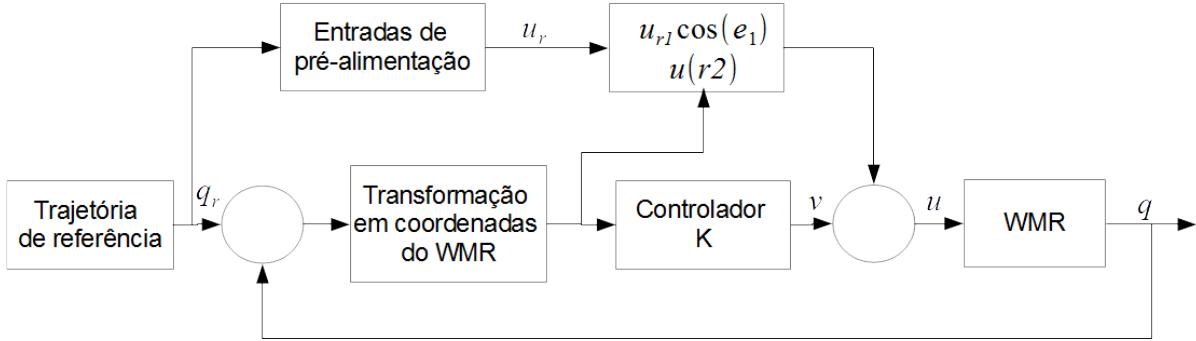
$$\mathbf{v} = \mathbf{K} \cdot \mathbf{e} \quad (3.5)$$

O sistema tem três estados e duas entradas, desta forma a dimensão da matriz de ganho \mathbf{K} é 2×3 . A estrutura do controlador é apresentado na Fig. 3.2, em forma de diagrama de blocos. A partir da trajetória de referência são determinadas as entradas de pré-alimentação, essas entradas são as velocidades linear e angulares de referência. Esses valores determinam o sinal de controle aplicado nos motores de acionamento das rodas do robô.

Para reduzir o erro na direção de movimento e_1 a velocidade tangencial deve mudar correspondentemente, da mesma forma a orientação do erro e_3 pode ser manipulada através da velocidade angular do robô. Já o erro da direção ortogonal do movimento pode ser reduzido

mudando a velocidade angular (KLANCAR *et al.*, 2005).

Figura 3.2: Diagrama de Blocos para o controlador Klancar



(KLANCAR *et al.*, 2005)

A determinação dos ganhos do controlador é feito através da comparação entre os polinômios real e desejado característico. Para um sistema de segunda ordem o coeficiente de amortecimento desejado $\zeta \in (0, 1)$ e a frequência natural $\omega_n > 0$. Um polo extra em $s = -2\zeta\omega_n$ aumenta o tempo de subida e diminui o overshoot do sistema. (KLANCAR *et al.*, 2005)

$$\det(s\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{K}) = s^3 + (k_1 + k_3)s^2 + (k_1k_3 + k_2v_r + \omega_r^2)s + k_1k_2ev_r + k_3\omega_r^2 \quad (3.6)$$

Desta forma, tem-se os seguintes coeficientes:

$$\begin{aligned} k_1 &= k_3 = 2\zeta\omega_n(t) \\ k_2 &= g \cdot |v_r(t)| \end{aligned} \quad (3.7)$$

sendo o parâmetro g o termo que corresponde ao grau de liberdade no projeto do controle, parâmetro este que deve ser positivo, $g > 0$ (KLANCAR *et al.*, 2005). Quanto maior este parâmetro mais rápida é a resposta do sistema.

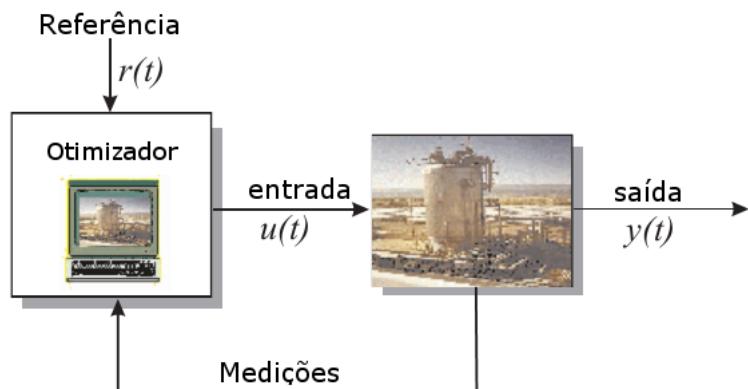
3.2 Controle Preditivo

O Controle Preditivo Baseado em Modelos, ou simplesmente Controle Preditivo (MBPC ou MPC- do inglês Model (Based) Predictive Control) é uma metodologia de controle que utiliza o modelo do processo *on-line*, ou seja dentro do cálculo do controle, para fazer as previsões dos valores de saída futuros da planta e desta forma otimizar as ações de controle futuro (CAMACHO; BORDONS, 1998). Sendo que o conceito de um controlador MPC não se trata de uma estratégia de controle específica e sim um conjunto de metodologias de controle que possuem algumas características em comum (KEYSER, 2003). A estrutura conceitual do MPC é

apresentada na Fig. 3.3.

O Controle Preditivo também é conhecido como controle de horizonte retroativo ou controle de horizonte em movimento. É utilizado a dinâmica explícita do modelo da planta para prever os efeitos da reação futura das variáveis manipuladas na saída e o sinal de controle obtido pela minimização de uma função custo. O desempenho do controlador depende de como a dinâmica do sistema é representada pelo modelo entrada-saída utilizado no projeto do controlador (HOLKAR; WAGHMARE, 2010). Essas estratégias estão sendo amplamente aplicadas na indústria como uma maneira efetiva para tratar os problemas de controles multivariáveis com restrições (BEMPORAD; MORARI, 1999).

Figura 3.3: Estrutura básica do Controle Preditivo



(BEMPORAD; MORARI, 1999)

As características básicas das estratégias de controle que compõem a família de controladores MPC são (CAMACHO; BORDONS, 1998):

- A utilização explícita das previsões das saídas do modelo de processo em instantes de tempo futuro, denominado horizonte.
- O valor do controle é determinado pela minimização de uma função custo, esta função pode incluir restrições.
- Estratégia retroativa, trata de uma estratégia na qual a cada instante o horizonte é deslocado para o futuro envolvendo a aplicação do valor do primeiro controle calculado em cada passo.

As diferenças existentes entre os algoritmos do MPC são: o modelo utilizado para representar o processo e o ruído, a função custo a ser minimizada, se a mesma leva ou não em consideração a existência de restrições para a determinação do controlador (CAMACHO; BORDONS, 1998). Existe uma ampla utilização dessas estratégias em vários meios tanto no meio acadêmico quanto na indústria, também sendo utilizado no campo médico.

No MPC, os modelos de processo podem ser dos tipos lineares ou não lineares. São utilizados para avaliar o comportamento do sistema. O movimento futuro das variáveis manipuladas é determinado pela minimização do erro previsto ou da função objetiva (HOLKAR; WAGHMARE, 2010).

Os controladores MPC possuem algumas vantagens em relação a outros métodos de controle. Entre elas: a ampla variedade de processos nos quais o controle MPC pode ser aplicado, sendo esses com dinâmica simples ou complexas, sistemas com um atraso grande, fase não mínima, instáveis, não lineares e multivariáveis. As restrições são incluídas diretamente do projeto do controlador.

A principal desvantagem da utilização de um controlador preditivo é o custo computacional uma vez que suas derivações são mais complexas do que o controle clássico PID. Se a dinâmica do processo não sofrer alterações as derivações do controle podem ser determinadas anteriormente, mas no caso do controle adaptativo, todos os cálculos devem ser realizados a cada tempo de amostra. Quando restrições são consideradas eleva-se o custo computacional envolvido (CAMACHO; BORDONS, 1998).

Com os recursos computacionais existentes na atualidade isto não é necessariamente um problema, mas deve ser levado em consideração que o computador não é utilizado apenas para o algoritmo de controle, mas também para comunicação entre os operadores, alarmes gravações entre outros (CAMACHO; BORDONS, 1998).

A disseminação do MPC só foi possível a partir do desenvolvimento e melhorias de tecnologias, que atualmente estão atingindo um certo estado de maturidade, essas tecnologias são necessárias para a aplicação das estratégias de controle MPC. Entre elas tem-se (CAMACHO; BORDONS, 1998):

1. **Identificação e modelamento:** existem diversas e poderosas técnicas, sendo possível trabalhar em ambientes adversos, presença de ruídos.
2. **Computadores:** São equipamentos mais rápidos, confiáveis e acessível, capaz de processar algoritmos *on-line* complexos.

3.2.1 Metodologia MPC

A seguinte notação será utilizada durante o texto (CAMACHO; BORDONS, 1998):

- t = índice de tempo discreto $t = 0, 1, 2, 3, \dots$;
- $u(t)$ = entrada do processo (variável manipulada - VM);
- $y(t)$ = saída do processo (variável controlada VC);

- $w(t)$ = setpoint ou referência;
- $u(t + k|t)$ =valores futuros de entrada, postulados em um tempo t ;
- $y(t + k|t)$ =valores previstos para a saída, baseado em:
 - Medidas disponíveis no tempo t : $y(t), y(t - 1), \dots, u(t - 1), u(t - 2), \dots$
 - Valores futuros postulados para a entrada : $u(t|t), u(t + 1|t), \dots$

O conjunto de algoritmos que pertencem ao MPC, representados pelo diagrama de blocos da Fig. 3.4, tem como característica a estratégia descrita a seguir (CAMACHO; BORDONS, 1998):

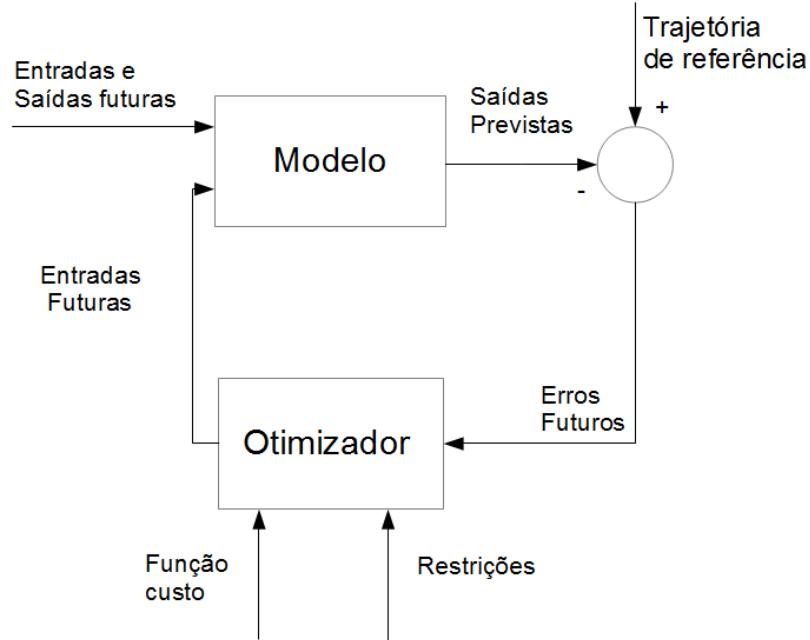
1. O horizonte N determina as saída futuras, sendo denominado de horizonte de predição e determinada para cada instante de tempo t utilizando o modelo de processo. Esses valores previstos para a saída $y(t + k|t)$ para $k = 1 \dots N$ depende dos valores conhecidos, saídas e entradas passadas, no instante t e os sinais de controle futuro $u(t + k|t)$, $k = 0 \dots N - 1$, que serão enviadas para o sistema e calculados.
2. O conjunto de sinais de controle futuro é determinado pelo cálculo da otimização determinado pelo critério em ordem de manter o processo o mais próximo possível da trajetória de referência $w(t + k)$, esta trajetória pode ser o próprio setpoint ou uma aproximação do mesmo.
3. O sinal de controle $u(t|t)$ é enviado para o processo enquanto os valores calculados para o próximo sinal de controle são rejeitados, pois a nova amostra de tempo $y(t + 1)$ já é conhecida e o passo 1 é repetido com este novo valor e todas as sequencias são atualizadas. E o $u(t + 1|t + 1)$ é calculado utilizando o conceito de horizonte retrocedente.

O modelo utilizado para predizer as saídas futuras da planta, sendo baseado nos valores passados e atuais e nos valores propostos de ações de controle ótimo futuro. Essas ações são calculadas levando em consideração a função custo, que leva em consideração o seguimento de erro futuro, como também restrições (CAMACHO; BORDONS, 1998).

Consequentemente, tem-se a importância do modelo do processo em relação ao controlador. O modelo escolhido deve ser capaz de apresentar a dinâmica do processo para desta forma ser capaz de predizer precisamente as saídas futuras e também deve ser um modelo de simples entendimento e também de implementação (CAMACHO; BORDONS, 1998).

Em resumo, temos os algoritmos pertencentes ao MPC tem as seguintes características em comum:

Figura 3.4: Estrutura básica para os controladores MPC



(CAMACHO; BORDONS, 1998)

- Predição através de um modelo do processo,
- Determinação de uma referência para a trajetória ou de *set-point*,
- Estruturação de uma lei futura de controle,
- Definição de uma função custo e restrições,
- Determinação da otimização do controle

3.2.2 Elementos do MPC

Existem três elementos comuns entre os algoritmos MPC, diferentes opções para cada um desses elementos podem ser escolhidos de acordo com o algoritmo escolhido, estes elementos são (CAMACHO; BORDONS, 1998):

- O modelo de predição
- A função objetiva
- O procedimento para a determinação da lei de controle.

3.2.3 Modelo de predição

O modelo de predição deve ser capaz de capturar a dinâmica do processo, permitir que as predições sejam calculadas, ser intuitivo e permitir análises teóricas. Essa utilização é necessário para o cálculo da predição da saída do processo em um instante futuro $\hat{y} + k|t$, esses modelos representam as relações sendo que as diferentes estratégias MPC podem utilizar diferentes modelos de predições. Um modelo de perturbações pode ser levado em consideração de forma a descrever o comportamento que não é aparente no modelo do processo, tais como: efeitos de entradas não mensuráveis, ruído e erros do modelo. Desta forma, o modelo pode ser separado em duas partes necessárias para a predição: o modelo atual do processo e o modelo de perturbações (CAMACHO; BORDONS, 1998).

Modelo do processo

As estratégias de controle MPC podem utilizar diversos tipos de modelos variando de acordo com a formulação específica para o algoritmo, sendo os mais comumente utilizados: resposta ao impulso, resposta ao degrau, funções de transferência, espaço de estado pode ser utilizados modelos não lineares, redes neurais e lógica de Fuzzy entre outras formas de representação (CAMACHO; BORDONS, 1998).

Modelos de perturbações

A escolha do modelo utilizado para representar as perturbações é tão importante quanto a escolha do modelo do processo. O modelo de perturbações é a diferença entre a saída medida (CAMACHO; BORDONS, 1998).

3.2.4 Função objetivo

Os variados algoritmos MPC propõem diferentes tipos de função custo para obter a lei de controle. O objetivo é que a saída futura (y) no horizonte de controle considerado deve seguir a um sinal de referência determinado (w) e ao mesmo tempo o esforço de controle (δu) necessário para a realização dessa tarefa deve ser penalizado. A expressão geral para uma função objetivo desta forma é (CAMACHO; BORDONS, 1998):

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j)[\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=N_1}^{N_u} \lambda(j)[\delta u(t+j-1)]^2 \quad (3.8)$$

Em uma função custo é possível a consideração de (CAMACHO; BORDONS, 1998):

- **Parâmetros:** N_1 e N_2 são horizontes de custo mínimo e máximo, respectivamente e N_u é o horizonte de controle que não necessariamente coincide com o horizonte máximo. Os

significados de N_1 e N_2 são intuitivos, marcando os limites de instantes no qual é desejado que a saída siga a referência.

- **Trajetória de referência:** uma das vantagens do controle preditivo é que as evoluções futuras do processo são conhecidas previamente, e desta forma o sistema pode reagir antes que a mudança seja efetivamente realizada, desta forma evitando-se os efeitos de atraso na resposta do processo, a evolução futura da referência $r(t + k)$ é conhecida antecipadamente em diversas aplicações, como por exemplo na robótica, servos ou em processos descontínuo. A trajetória de referência pode ser utilizada para especificar um comportamento de ciclo de malha fechada; essa idéia é utilizada nos controladores GPC e EPSAC.
- **Restrições** Na prática todos os processos são sujeitos a restrições. Os atuadores tem uma faixa limitada de campo de ação assim como uma determinada taxa de variação. Razões de construção, segurança ou ambiental, até a amplitude dos sensores pode causar um limite em variáveis, como por exemplo, níveis em um tanque, fluxo em tubulações entre outros. Por essas razões é necessário a introdução de restrições nas funções a serem minimizadas. Normalmente os limites na amplitude e as taxas de variações do sinal de controle e os limites na saída será considerado:

$$\begin{aligned} u_{min} &\leq u(t) \leq u_{max} & \forall t \\ \delta u_{min} &\leq u(t) - u(t-1) \leq \delta u_{max} & \forall t \\ y_{min} &\leq y(t) \leq y_{max} & \forall t \end{aligned}$$

Adicionando essas restrições as funções objetivas na minimização se tornam ainda mais complexas, então a solução não pode ser obtidas explicitamente como no caso sem restrições (CAMACHO; BORDONS, 1998).

3.2.5 Obtenção da lei de controle

Para obter os valores do controle a ser aplicado na planta $u(t + k|t)$ é necessário a minimização da função objetivo J Eq. 3.8. Para fazer isso os valores das previsões de saída $\hat{y}(t + k|t)$ são calculados como uma função dos valores passados de entrada e saída e dos valores de sinais de controle, fazendo uso do modelo escolhido e o substituindo na função custo, obtendo expressões as quais as suas minimização tendem aos valores procurados (CAMACHO; BORDONS, 1998).

3.2.6 Respostas livre e forçada

Uma característica típica da maioria dos controladores MPC é a utilização dos conceitos de resposta livre e resposta forçada. A idéia é expressar a seqüência de controle como sendo a adição de dois sinais (CAMACHO; BORDONS, 1998):

$$u(t) = u_{livre} + u_{for} \quad (3.9)$$

O sinal u_{livre} corresponde as entradas passadas e é mantido constante e igual ao último valor da manipulação de variáveis em um instante de tempo futuro. Enquanto o sinal u_{for} é igualado a zero no passado e feito igual ao próximo movimento do controle futuro (CAMACHO; BORDONS, 1998).

A previsão da sequência de saída é separada em duas partes. Um parte refere-se a resposta livre ($y_{livre}(t)$) correspondendo a previsão se saída quando a variável de processo manipulada é feita igual a $u_{livre}(t)$, a outra, a resposta forçada ($y_{for}(t)$) corresponde a previsão da saída do processo quando a sequência de controle é feita igual a $u_{foc}(t)$. A resposta livre corresponde a evolução do processo devido ao estado presente, enquanto a resposta forçada ocorre devido aos movimentos de controle futuro (CAMACHO; BORDONS, 1998).

3.3 Controlador MPC Linearizado

O segundo controlador para seguimento de trajetória de um robô móvel com rodas foi proposto em Kühne et al.(2004) e também utiliza um estratégia de controlador MPC na qual o método consiste em otimizar as previsões do comportamento do processo em relação a sequência de entradas do controlador futuro. Além da função custo a ser minimizada, essa estratégia tem como peculiaridade a utilização do modelo do robô linearizado, conforme foi apresentado na seção 2.3.1. A função objetiva a ser minimizada é a função quadrática dos estados e das entradas dos controladores (KÜHNE *et al.*, 2004):

$$\Phi(k) = \sum_{j=1}^N \tilde{\mathbf{x}}^T(k+j|k) \mathbf{Q} \tilde{\mathbf{x}}(k+j|k) + \tilde{\mathbf{u}}^T(k+j-1|k) \mathbf{R} \tilde{\mathbf{u}}(k+j-1|k) \quad (3.10)$$

onde:

N é o horizonte de previsão;

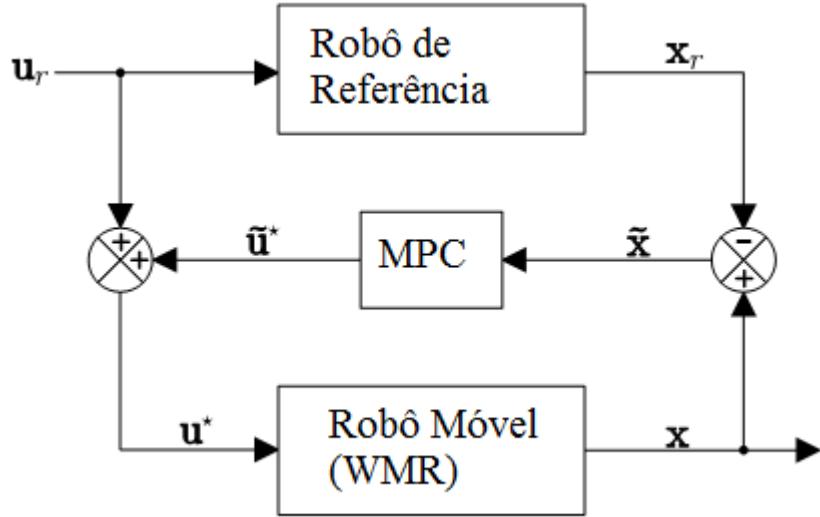
\mathbf{Q} e \mathbf{R} sendo matrizes de peso, onde : $\mathbf{Q} \geq 0$ e $\mathbf{R} > 0$.

O problema de otimização pode ser definido para determinar $\tilde{\mathbf{u}}^*$:

$$\tilde{\mathbf{u}}^* = \arg \min \Phi(k) \quad (3.11)$$

A cada interação k é calculado a minimização da Eq. 3.10, fornecendo uma sequência de controle ótimo $\tilde{\mathbf{u}}^*(k|k), \dots, \tilde{\mathbf{u}}^*(N-1|k)$ e o custo ótimo $\Phi^*(k)$, sendo que a lei de controle é o primeiro elemento da sequência do controle ótimo, ou seja $\tilde{\mathbf{u}}^*(k|k)$. O diagrama de bloco que representa a estrutura do controle é apresentado na Fig. 3.5. Para a solução da otimização dois vetores são definidos (KÜHNE *et al.*, 2004):

Figura 3.5: Diagramas de Blocos MPC linearizado



Fonte: (KÜHNE *et al.*, 2004)

$$\tilde{\mathbf{x}}(k+1) \triangleq \begin{bmatrix} \tilde{x}(k+1|k) \\ \tilde{x}(k+2|k) \\ \vdots \\ \tilde{x}(k+N|k) \end{bmatrix} \quad (3.12)$$

$$\tilde{\mathbf{u}}(k) \triangleq \begin{bmatrix} \tilde{x}(k|k) \\ \tilde{x}(k+1|k) \\ \vdots \\ \tilde{x}(k+N-1|k) \end{bmatrix} \quad (3.13)$$

Desta forma a Eq.3.10 pode ser reescrita como:

$$\Phi(k) = \tilde{\mathbf{x}}^T * (k+1) \tilde{\mathbf{Q}} \tilde{\mathbf{x}}(k+1) + \tilde{\mathbf{u}}^T(k) \tilde{\mathbf{R}} \tilde{\mathbf{u}}(k) \quad (3.14)$$

Sendo:

$$\tilde{\mathbf{Q}} \triangleq \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{Q} \end{bmatrix} \quad (3.15)$$

e:

$$\tilde{\mathbf{R}} \triangleq \begin{bmatrix} \mathbf{R} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{R} \end{bmatrix} \quad (3.16)$$

Pode-se escrever a expressão $\tilde{\mathbf{x}}(k+1)$, como sendo:

$$\tilde{\mathbf{x}}(k+1) = \tilde{\mathbf{A}}(k)\tilde{\mathbf{x}}(k|k) + \tilde{\mathbf{B}}(k)\tilde{\mathbf{u}}(k) \quad (3.17)$$

sendo as matrizes $\tilde{\mathbf{A}}$ e $\tilde{\mathbf{B}}$ definido como:

$$\tilde{\mathbf{A}}(k) \triangleq \begin{bmatrix} \mathbf{A}(k|k) \\ \mathbf{A}(k|k)\mathbf{A}(k+1|k) \\ \vdots \\ \alpha(k, 0) \end{bmatrix} \quad (3.18)$$

$$\tilde{\mathbf{B}}(k) \triangleq \begin{bmatrix} \mathbf{B}(k|k) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}(k+1|k)\mathbf{B}(k|k) & \mathbf{B}(k+1|k) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha(k, 1)\mathbf{B}(k|k) & \alpha(k, 2)\mathbf{B}(k+1|k) & \dots & \mathbf{B}(k+N-1|k) \end{bmatrix} \quad (3.19)$$

sendo $\alpha(k, j)$ definido como:

$$\alpha(k, j) \triangleq \prod_{i=j}^{N-1} \mathbf{A}(k+i|k) \quad (3.20)$$

Reescrevendo a Eq. 3.10, a partir da Eq. 3.14 com as matrizes definidas em Eq. 3.15 e 3.16, a função quadrática se torna:(KÜHNE *et al.*, 2004)

$$\Phi(k) = \frac{1}{2} \tilde{\mathbf{u}}^T(k) \mathbf{H}(K) \tilde{\mathbf{u}}(k) + \mathbf{f}^T(k) \tilde{\mathbf{u}}(k) + \mathbf{d}(k) \quad (3.21)$$

com:

$$\mathbf{H}(k) \triangleq 2(\tilde{\mathbf{B}}(k)^T(k) \tilde{\mathbf{Q}} \tilde{\mathbf{B}}(k) + \tilde{\mathbf{R}})$$

$$\mathbf{f}(k) \triangleq 2\tilde{\mathbf{B}}^T(k) \tilde{\mathbf{Q}} \tilde{\mathbf{A}}(k) \tilde{\mathbf{x}}(k|k)$$

$$\mathbf{d}(k) \triangleq \tilde{\mathbf{x}}^T(k|k) \tilde{\mathbf{A}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{A}}(k) \tilde{\mathbf{x}}(k|k)$$

A Matriz \mathbf{H} é denominada de matriz Hessiana e deve ser uma matriz definida positiva. Ela refere-se a parte quadrática da função objetiva. Enquanto \mathbf{f} descreve a parte linear e o termo \mathbf{d} é a parte independente de $\tilde{\mathbf{u}}$ e não influencia na determinação de \mathbf{u}^* . Para resolver o problema de otimização deve se considerado que o controle fique dentro de uma faixa, expressando os valores das restrições do controlador (KÜHNE *et al.*, 2004):

$$\mathbf{u}_{min}(k) \leq \mathbf{u}(k) \leq \mathbf{u}_{max}(k) \quad (3.22)$$

Desta forma, a Eq. 3.11 pode ser reescrita para encontrar $\tilde{\mathbf{u}}^*$, na qual :(KÜHNE *et al.*, 2004)

$$\tilde{\mathbf{u}}^* = \arg \min_{\tilde{\mathbf{u}}} \Phi(k) \quad (3.23)$$

sujeito a :

$$\mathbf{D}\tilde{\mathbf{u}} \leq \mathbf{d} \quad (3.24)$$

onde $\Phi(k)$ é a função objetivo e $\tilde{\mathbf{u}}$ é a variável livre de otimização. A inequação, apresentada na Eq.3.24, é a forma geral com a qual descrever restrições em variáveis de controle. Se for considerado apenas restrições no controle como na Eq. 3.27, tem-se:(KÜHNE *et al.*, 2004)

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \tilde{\mathbf{u}} \leq \begin{bmatrix} \tilde{\mathbf{u}}_{max} \\ -\tilde{\mathbf{u}}_{min} \end{bmatrix} \quad (3.25)$$

Podendo ser reescrito da seguinte forma:

$$\tilde{\mathbf{u}}_{min} \leq \tilde{\mathbf{u}} \leq \tilde{\mathbf{u}}_{max} \quad (3.26)$$

Como a variável de otimização livre é $\tilde{\mathbf{u}}(k)$, reescrevendo a Eq. 3.26 levando em consideração a variável livre, em forma vetorial:(KÜHNE *et al.*, 2004)

$$\bar{\mathbf{u}}_{min}(k) - \bar{\mathbf{u}}_r(k) \leq \bar{\mathbf{u}}(k) \leq \bar{\mathbf{u}}_{max}(k) - \bar{\mathbf{u}}_r(k) \quad (3.27)$$

com:

$$\bar{\mathbf{u}}_{min}(k) \triangleq \begin{bmatrix} \mathbf{u}_{min}(k) \\ \mathbf{u}_{min}(k+1) \\ \vdots \\ \mathbf{u}_{min}(k+N-1) \end{bmatrix} \quad (3.28)$$

$$\bar{\mathbf{u}}_{max}(k) \triangleq \begin{bmatrix} \mathbf{u}_{max}(k) \\ \mathbf{u}_{max}(k+1) \\ \vdots \\ \mathbf{u}_{max}(k+N-1) \end{bmatrix} \quad (3.29)$$

$$\bar{\mathbf{u}}_r(k) \triangleq \begin{bmatrix} \mathbf{u}_r(k) \\ \mathbf{u}_r(k+1) \\ \vdots \\ \mathbf{u}_r(k+N-1) \end{bmatrix} \quad (3.30)$$

Como o estado de predição é a função da sequência ótima a ser calculada, é fácil mostrar que o estado de restrições pode ser escrito genericamente pela Eq. 3.24. Além disso, restrições nas taxas de mudança de controle e estados pode ser formuladas de maneira similar (KÜHNE *et al.*, 2004).

4 Extended Prediction Self Adaptive Control - EPSAC

O EPSAC é um dos diversos algoritmos pertencente à família dos Controles Preditivos. Como tal, possui as características em comum a todos os algoritmos pertencentes ao MPC, como a utilização do modelo de processo para obter uma predição dos valores futuros, a determinação de uma referência para a trajetória. Devido a essas características é um algoritmo utilizado em uma ampla variedade de processos industriais e sendo pesquisado para processos específicos que vão desde o controle em uma torre de destilação (POP *et al.*, 2012) , até a aplicação na medicina, como por exemplo o controle de temperatura e umidade em uma incubadora neonatal(BRITO *et al.*, 2010).

Entre as características que tornaram o EPSAC uma escolha para o controle de trajetória de um robô móvel é a possibilidade de se trabalhar com a trajetória de referência para o processo e a outra é a capacidade de trabalhar diretamente tanto com sistemas não lineares quanto sistemas MIMO (sistema com múltiplas entradas e múltiplas saídas, do inglês *Multi Input and Multi Output*) para robô móvel a ser controlado trata-se de um sistema com duas entradas as velocidades lineares e angulares v e w , respectivamente e três saídas a posição no plano cartesiano x e y e a orientação do robô θ .

Para a utilização de um controlador baseado na metodologia MPC, como é o caso do EP-SAC, um dos aspectos fundamentais é a predição da saída do processo $y(t+k|t), k = 1 \dots N_2$. Essa predição é baseada nas medidas disponíveis no instante de tempo t : $y(t), y(t-1), \dots, u(t-1), u(t-2), \dots$ e nos valores futuros estimados para a entrada do sistema: $u(t|t), u(t+1|t), \dots$ (KEYSER, 2003).

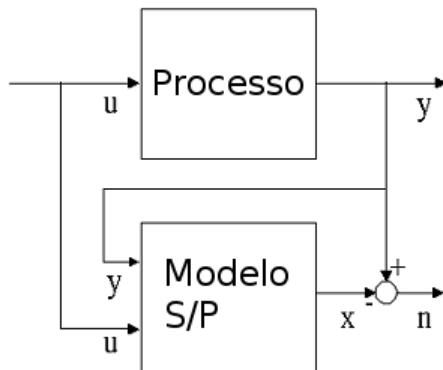
Para o EPSAC, um modelo genérico é utilizado,a solução desse problema de medição multi passos é feita através da utilização de técnicas de filtragem. Reescrevendo o modelo do processo no tempo, tem-se que (KEYSER, 2003):

$$y(t+k|t) = x(t+k|t) + n(t+k|t) \quad (4.1)$$

4.1 Predição do $x(t+k|t)$

Para a determinação da predição dos valores de $x(t+k|t)$ é utilizado à recursividade do modelo do sistema. Sendo, duas as possibilidades de configuração: modelo em série e o modelo série-paralelo. O modelo paralelo, também chamado de modelo independente só pode ser utilizado para processos estáveis, enquanto o modelo em série-paralelo, ou modelo realinhado, pode ser utilizado também para processos instáveis (KEYSER, 2003). A predição de $x(t+k|t)$ para aplicação a do robô foi o utilizado o modelo série - paralelo, sendo que o diagrama de blocos desse modelo é apresentado na Fig. 4.1

Figura 4.1: Diagrama de blocos Modelo Série-Paralelo



Fonte: (KEYSER, 2003)

Para cada instante de amostragem t , a recursão inicia no instante $k = 0$ e calcula-se o valor de $x(t|t)$ utilizando o vetor de entrada do modelo $[x(t-1)x(t-2)x(t-3)\dots u(t-1)u(t-2)u(t-3)\dots]$, que contém os valores passados conhecidos até o tempo t , disponível no banco de dados do computador. O valor $x(t) \equiv x(t|t)$ deve ser salvo no banco de dados do computador para a utilização nas próximas amostras. Desta forma, o valor $x(t|t)$ anteriormente calculado é utilizado como entrada do modelo para determinar os valores de $x(t+1|t)$ e outras variáveis (KEYSER, 2003).

4.2 Predições para $n(t+k|t)$

Utilizando as saídas do processo medidas $y(t)$ calcula-se o valor atual do ruído $n(t)$ com os valores do modelo genérico do processo Eq. 4.1: $n(t) = y(t) + x(t)$. Os valores anteriores $n(t-1), n(t-2), \dots$ estão disponíveis no banco de dados, uma vez que eles foram determinados em instantes de tempo anteriores. Sendo que para calcular o sinal filtrado de distúrbio é utilizada a seguinte expressão (KEYSER, 2003):

$$n_f(t) = \frac{D(q^{-1})}{C(q^{-1})} n(t), \quad \text{com :} \quad (4.2)$$

$$n_f(t) = -c_1 n_f(t-1) - c_2 n_f(t-2) - \dots \quad (4.3)$$

$$\dots + n(t) + d_1 n(t-1) + d_2 n(t-2) + \dots \quad (4.4)$$

Como é desejado que o sistema rejeite sinais de perturbações constantes o modelo de perturbações dado por:

$$n(t) = \frac{C(q^{-1})}{D(q^{-1})} e(t) \quad (4.5)$$

onde $C(q^{-1})$ é um filtro, o termo $D(q^{-1})$ refere-se a um integrador e $e(t)$ é modelado como sendo um ruído branco. A conclusão é que o sinal $n_f(t)$ pode ser calculado como ruído branco com média zero : $n_f(t) = e(t)$. O motivo dessa consideração é que o ruído branco por definição é não-correlacionado e sua melhor predição é o valor médio, desta forma (KEYSER, 2003):

$$n_f = (t+k|t) \equiv 0, k = 1 \dots N_2 \quad (4.6)$$

Então a predição para a perturbação é obtida da seguinte forma:

$$n(t+k|t) = \frac{C(q^{-1})}{D(q^{-1})} n_f(t+k|t) \quad (4.7)$$

$$n(t+k|t) = -d_1 n(t+k-1|t) - d_2 n(t+k-2|t) - \dots + n_f(t+k|t) + c_1 n_f(t+k-1|t) + \dots \quad (4.8)$$

A recursão vai de $k = 1 \dots N_2$. Para o instante de tempo $k = 1$, o valor do sinal do lado direito da equação $n(t|t), n(t-1|t), \dots, n_f(t|t), n_f(t-1|t), \dots$ são conhecidos do banco de dados, enquanto $n_f(t+1|t) \equiv 0$ de acordo com a Eq. 4.6. O valor calculado $n(t+1|t)$ é utilizado no lado direito da equação, junto com o valor de $n_f(t+2|t) \equiv 0$, em ordem para calcular $n(t+2|t)$ e assim por diante (KEYSER, 2003).

4.3 Resposta base - resposta otimizada

A resposta futura $y(t+k|t)$ é definida conceitualmente e baseada no conceito de sistemas lineares de sobreposição como sendo (KEYSER, 2003):

$$y(t+k|t) = y_{base}(t+k|t) + y_{opt}(t+k|t) \quad (4.9)$$

Sendo as que as contribuições têm origens diferentes. O $y_{base}(t+k|t)$ tem as seguintes

contribuições, os efeitos do (KEYSER, 2003):

- Perturbações futuras previstas $n(t+k|t)$
- Controle passado $u(t-1), u(t-2), \dots$
- O cenário do controle futuro básico $u_{base}(t+k|t), k = 0 \dots N_2 - 1$

Sendo esse controle futuro básico definido, *a priori*, para modelos básicos. A escolha desse vetor de controle básico é irrelevante, qualquer escolha seja $u_{base}(t+k|t) \equiv 0, k = 0 \dots N_2 - 1$ ou $u_{base}(t+k|t) \equiv u(t-1), k = 0 \dots N_2 - 1$, terá como resposta o mesmo cenário de controle ótimo. Para sistemas não-lineares, essa escolha deve ser feita de uma forma particular (KEYSER, 2003).

A componente $y_{base}(t+k|t)$ pode ser calculado como mostrado anteriormente, sendo à entrada do modelo o $u_{base}(t+k|t)$.

Para o $y_{opt}(t+k|t)$ a contribuição é referente ao efeito da ação da otimização do controle futuro $\delta u(t+k|t), k = 0 \dots N_2 - 1$ com $\delta u(t+k|t) = u(t+k|t) - u_{base}(t+k|t)$, o termo $u(t+k|t)$ é o controle ótimo. (KEYSER, 2003) É uma prática comum no MPC estruturar um cenário de controle futuro, desta forma reduzindo os graus de liberdade do vetor de controle $u(t+k|t), k = 0 \dots N_2 - 1$. Isso pode ser realizado definindo um horizonte de controle N_u (com $1 \leq N_u \leq N_2$) depois a estratégia de controle postulada se torna constante: $u(t+k|t) = u(t+N_u-1|t), k \geq N_u$, desta forma é reduzido os graus de liberdade de N_2 para N_u . Em diversas aplicações práticas, uma versão extremamente simplificada com $N_u = 1$ tem bons resultados.

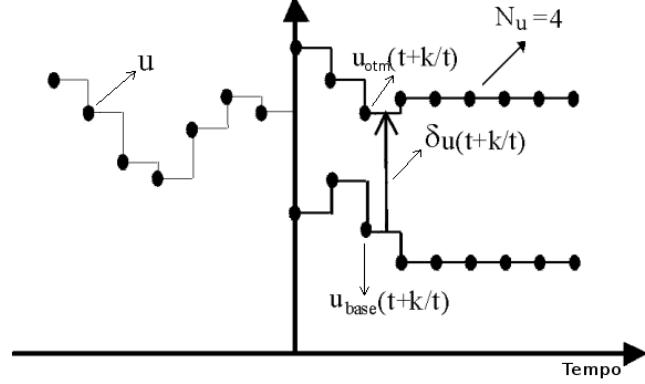
A ilustração dos conceitos de estratégias de controle básico e ótimo são mostrados na Fig. 4.2. Na qual o termo $u_{opt}(t+k|t)$ é o controle desejado e composto de duas parcelas o termo $u_{base}(t+k|t)$ cenário de controle básico no futuro diretamente relacionado ao modelo não linear, enquanto $(t+k|t)$ é o efeito da ação de controle futuro relacionado a parte linear do modelo, desta forma é feita diversas iterações para que esse termo se aproxime de zero (KEYSER, 2003).

O conceito de horizonte de controle implica que $u_{base}(t+k|t) = u_{base}(t+N_u-1|t), k \geq N_u$ e $\delta u(t+k|t) = \delta u(t+N_u-1|t), k \geq N_u$, como o mostrado na figura. A figura indica que a componente $y_{opt}(t+k|t)$ da Eq. 4.9 é o resultado de $\delta u(t+k|t)$, sendo o efeito cumulativo da série de entradas impulso e degrau (KEYSER, 2003).

Um impulso com amplitude $\delta u(t|t)$ no instante de tempo t , resulta na contribuição $h_k \delta u(t|t)$ para a saída do sistema no instante de tempo $t+k$, sendo k a quantidade de amostras posteriores, como o ilustrado na Fig. 4.3 (KEYSER, 2003).

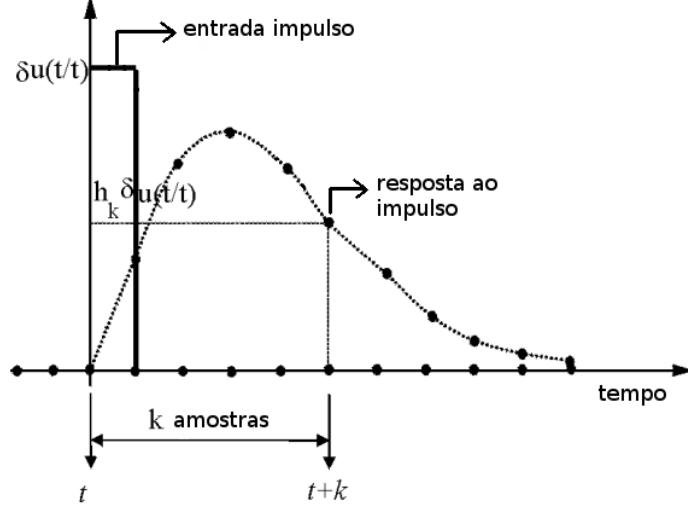
Para um impulso com amplitude $\delta u(t+1|t)$ no tempo $t+1$ resulta na contribuição $h_{k-1} \delta u(t+1|t)$

Figura 4.2: Ilustração do conceito de resposta base e resposta ótima



Fonte: (KEYSER, 2003)

$1|t)$ para a saída do sistema no instante $t + k$, ou seja, $k - 1$ amostras posteriores, como o observado na Fig.4.4. E assim, sucessivamente (KEYSER, 2003).

Figura 4.3: Efeito em um instante de tempo $t + k$ do sinal impulso aplicado no instante t 

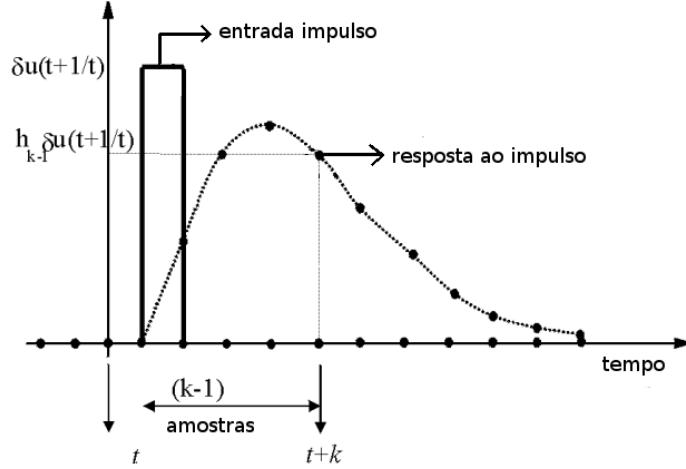
Fonte: (KEYSER, 2003)

A aplicação de um degrau com amplitude $\delta u(t + N_u - 1|t)$ no instante de tempo $t + N_u - 1$ resulta em uma contribuição $g_{k-N_{u+1}} \delta u(t + N_u - 1|t)$ na saída do sistema no instante de tempo $t + k$, sendo $k - N_u + 1$ amostras posteriores (KEYSER, 2003).

Sendo que o efeito acumulativo dos efeitos na saída do sistema no instante $t + K$ de todos os impulsos e do degrau aplicados pode ser escrito da seguinte forma (KEYSER, 2003):

$$y_{otm} = (t + k|t) = h_k \delta u(t|t) + h_{k-1} \delta u(t + 1|t) + \dots + g_{k-N_{u+1}} \delta u(t + N_u - 1|t) \quad (4.10)$$

Os parâmetros $h_1, h_2, \dots, h_k \dots h_{N_2}$ são os coeficientes de resposta ao degrau do sistema.

Figura 4.4: Efeito em um instante de tempo $t + k$ do sinal impulso aplicado no instante $t + 1$ 

Fonte: (KEYSER, 2003)

Enquanto os parâmetros $g_1, g_2, \dots, g_k, \dots, g_{N_2}$ são os coeficientes de resposta ao degrau do sistema. Esses valores podem ser obtidos de duas maneiras a primeira é a forma direta, ou seja, aplicando um sinal de impulso ou degrau no sistema, isso pode ser aplicado tanto para sistemas lineares quanto para sistemas não-lineares. Para a segunda, é uma forma analítica a partir da função de transferência ou do modelo de espaço de estado (KEYSER, 2003).

Sendo que $h_0 = h_{-1} = h_{-2} = \dots = g_0 = g_{-1} = g_{-2} = \dots = 0$ e que os coeficientes da resposta ao impulso podem ser facilmente calculados a partir dos valores dos coeficientes de resposta ao degrau e *vice versa*: $h_k = g_k - g_{k-1}$, desta forma apenas uma das duas respostas deve ser calculadas (KEYSER, 2003).

Uma das características dos algoritmos pertencentes a famílias de controle do MPC é que eles consistem em aplicar uma sequência de controle que minimize uma função custo multi-estágio da forma (KEYSER, 2003):

$$J = \sum_{k=N_1}^{N_2} [y(t+k|t) - w(t+k)]^2 + \sum_{k=0}^{N_u-1} \lambda [\Delta u(t+k|t)]^2 \quad (4.11)$$

Sujeito a:

$$\begin{aligned} y_{\min} &\leq y(t+k|t) \leq y_{\max} \quad \forall k = 1, \dots, N_2, \\ u_{\min} &\leq u(t+k|t) \leq u_{\max} \quad \forall k = 0, \dots, N_u - 1, \\ \Delta u_{\min} &\leq \Delta u(t+k|t) \leq \Delta u_{\max} \quad \forall k = 0, \dots, N_u - 1 \end{aligned} \quad (4.12)$$

onde N_1 e N_2 são os horizontes de custos mínimos e máximos, N_u é o horizonte de controle, λ é o peso do controle, $w(t+k)$ é o valor nominal ou a seqüência de referência, $\Delta u(t)$ é a ação de controle incremental, $\Delta = 1 - q^{-1}$ e $y(t+k|t)$ é o valor ótimo k a saída do sistema um passo adiante, $y(t)$ dados até o momento t (KEYSER, 2003).

Utilizando o modelo genérico do processo, as previsões de valores para a saída são:(KEYSER, 2003)

$$y(t+k/t) = x(t+k/t) + n(t+k/t) \quad (4.13)$$

Considerando a resposta futura como sendo resultado cumulativo de dois efeitos. E o princípio do EPSAC é baseado na minimização dos erros entre a trajetória da referência futura e a saída prevista do processo, calculada como (KEYSER, 2003):

$$y(t+k/t)_i = y_{livre_i}(t+k/t) + y_{for_i}(t+k/t) \quad (4.14)$$

Sendo y_{livre_i} a resposta livre determinada pela resposta do modelo genérico, enquanto y_{for_i} a resposta forçada do sistema é determinada pela sequência de entrada degrau. Desta forma, a resposta forçada pode ser escrita em forma de matriz (KEYSER, 2003):

$$\mathbf{Y}_{\text{for}} = \mathbf{GU} \quad (4.15)$$

e a equação para a lei de controle torna-se:

$$\mathbf{Y} = \mathbf{Y}_{\text{base}} + \mathbf{Y}_{\text{opt}} = \bar{\mathbf{Y}} + \mathbf{GU} \quad (4.16)$$

sendo:

$$\begin{aligned} \mathbf{Y} &= [y(t+N_1/t) \dots y(t+N_2/t)]^T \\ \bar{\mathbf{Y}} &= [y_{free}(t+N_1/t) \dots y_{free}(t+N_2/t)]^T \\ \mathbf{U} &= [\Delta u(t/t) \dots \Delta u(t+N_u-1/t)]^T \end{aligned}$$

$$G = \begin{bmatrix} g_{N_1} & 0 & \dots & \dots & 0 \\ g_{N_1+1} & g_{N_1} & 0 & \dots & 0 \\ g_{N_1+2} & g_{N_1+1} & g_{N_1} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & g_{N_2-2} & \dots & g_{N_2-N_u+1} \end{bmatrix}$$

sendo o controle ótimo a primeira linha do vetor:

$$u_0 = -H^{-1}b \quad (4.17)$$

onde :

$$H = 2(P^T Q_\lambda P + G^T G) \text{ e}$$

$$b^T = 2((Pu_b + u_i)^T Q_\lambda P + (y_b - \omega)^T G)$$

4.4 EPSAC não-linear

É necessário que algumas considerações sejam feitas para utilizar o algoritmo proposto para o EPSAC em modelos não lineares, o chamado NEPSAC (EPSAC não-linear), uma vez que para definir os conceitos de resposta e controle de base e otimizado, foi utilizado o princípio de superposição, então esse conceitos são válidos apenas para sistemas lineares. Porém, para sistemas não-lineares, através da seleção de uma estratégia de controle base $u_{base}(t+k|t)$ apropriado, o termo $y_{opt}(t+k|t)$ na Eq. 4.9 pode chegar à zero de uma forma gradual e iterativa. Desta forma, os resultados para a solução ótima é válido também para sistemas não-lineares, uma vez que o princípio de superposição não é mais utilizado. O procedimento do EPSAC, utilizado tanto para sistemas lineares quanto sistemas não-lineares, é descrito da seguinte forma, para cada amostra de tempo (KEYSER, 2003):

1. Selecione um vetor $u_{base}(t+k|t), k = 0 \dots N_u - 1$. Para os modelos lineares, essa escolha é irrelevante para as soluções. Entretanto, para modelos não-lineares, o objetivo deve ser obter de forma iterativa uma política de controle $u_{base}(t+k|t)$ que é o mais próximo possível da estratégia de controle ótimo $u(t+k|t)$. Desta maneira tornando as ações de controle ótimo $\delta u(t+k|t)$ e o termo correspondente $y_{opt}(t+k|t)$ próximo de zero. Para minimizar o número de iterações deve-se fazer uma boa estimativa inicial para o valor de $u_{base}(t+k|t)$. Uma escolha simples e efetiva é iniciar com $u_{base}(t+k|t) \equiv u^*(t+k|t-1)$, ou seja, a política de controle ótimo é derivada da amostra anterior.
2. Uma vez que foi escolhido $u_{base}(t+k|t), k = 0 \dots N_u - 1$ é calculado os resultados de $\delta u(t+k|t), k = 0 \dots N_u - 1$ e de $u(t+k|t) = u_{base}(t+k|t) + \delta u(t+k|t), k = 0 \dots N_u - 1$, para um modelo linear é determinado da maneira conforme foi Seção 4.3. Para os modelos não-lineares estes não são o controle ótimo uma vez que não é utilizado o princípio de superposição. Porém, é esperado que o resultado de $u(.|.)$ é mais próximo do valor de que o valor u_{base} anterior estimado. Para um modelo não-linear é sugerido que:
 - continue o procedimento, em uma mesma amostra de tempo, onde $u_{base}(t+k|t) = u_{base}(t|t) + \delta u(t|t)$ e após retorna-se ao passo 2.

Continuando esse procedimento iterativo é esperado que o $u_{base}(.|.)$ irá convergir para o $u(.|.)$ ótimo. A cada intervalo de tempo o $u_{base}(.|.)$ é mais próximo de $u(.|.)$, isso significa que os valores de $\delta u(.|.)$ são menores. Como o mostrado na Fig.4.2, e desta forma, o termo $y_{opt}(.|.)$ se torna menor. O princípio de superposição utilizado na determinação da Eq. 4.9, mas

é invalido para sistemas não-lineares. Sendo que a cada iteração com a redução do valor de $\delta u(\cdot|\cdot)$ também é reduzido o impacto do princípio da superposição. Finalmente, quando $\delta u(\cdot|\cdot)$ são praticamente zero, o princípio de superposição não é mais envolvido e o sinal de controle calculado será ótimo também para sistemas não-lineares (KEYSER, 2003).

A metodologia abordada acima para o NEPSAC é uma das diversas metodologias utilizadas o Controle Preditivo não-linear (NMPC - *Nonlinear Model Predictive Control*), sendo que nas abordagens NMPC o modelo nunca é linearizado. O modelo não linear é utilizado diretamente para os cálculos, em todas as iterações, da resposta base $y_{base}(\cdot|\cdot)$ e também utilizado para determinar os coeficientes da resposta ao degrau e/ou impulso g_i, h_i estes são os únicos valores requeridos para obter a solução do controle (KEYSER, 2003).

4.5 MIMO EPSAC

Foi apresentado nas sessões anteriores foi apresentada metodologia EPSAC para os sistemas com apenas uma entrada e uma saída (SISO - Single Input Single Output). Nessa seção é apresentada a expansão dos conceitos para os sistemas com múltiplas entradas e múltiplas saídas (MIMO - Mutiple Input Mutiple Output). Por simplicidade, será apresentado um sistema com duas entradas e duas saídas, a expansão para um sistema com n_u entradas e n_y saídas é feita de forma direta. Para um processo com 2 entradas e 2 saídas, o modelo genérico do processo Eq. 4.1, torna-se (KEYSER, 2003):

$$\begin{cases} y_1(t) = x_1(t) + n_1(t) \\ y_2(t) = x_2(t) + n_2(t) \end{cases} \quad (4.18)$$

Para uma estrutura de modelo paralelo:

$$\begin{cases} x_1(t) = f_1[x_1(t-1), x_1(t-2), \dots, u_1(t-1), u_1(t-2), \dots]u_2(t-1), u_2(t-2), \dots \\ x_2(t) = f_2[x_2(t-1), x_2(t-2), \dots, u_1(t-1), u_1(t-2), \dots]u_2(t-1), u_2(t-2), \dots \end{cases} \quad (4.19)$$

Para uma estrutura série-paralelo:

$$\begin{cases} x_1(t) = f_1[y_1(t-1), y_1(t-2), \dots, u_1(t-1), u_1(t-2), \dots]u_2(t-1), u_2(t-2), \dots \\ x_2(t) = f_2[y_2(t-1), y_2(t-2), \dots, u_1(t-1), u_1(t-2), \dots]u_2(t-1), u_2(t-2), \dots \end{cases} \quad (4.20)$$

As funções $f_1[\cdot]$ e $f_2[\cdot]$ são os modelos do processo e são conhecidas, que podem ter qualquer estrutura em geral (linear, não-linear, redes neurais, ...). Neste caso, será assumido que se trata de modelos lineares. As perturbações podem ser modeladas como processos de ruídos coloridos (e_1 e e_2 são sinais de ruído branco) (KEYSER, 2003):

$$n_1(t) = \frac{C_1(q^{-1})}{D_1(q^{-1})} e_1(t) \quad (4.21)$$

e:

$$n_2(t) = \frac{C_2(q^{-1})}{D_2(q^{-1})} e_2(t) \quad (4.22)$$

Um preditor de múltiplos passos foi apresentado. E este preditor é diretamente aplicado no modelo de processo com diversas entradas u_1, u_2 e pode ser aplicada para calcular $y_1(t+k|t)$ e $y_2(t+k|t)$, com $k = 1 \dots N_2$. Utilizando os conceitos de respostas base e otimizada apresentada 4.3, tem -se as seguintes relações (KEYSER, 2003):

$$\begin{cases} y_1(t+k|t) = y_{1base}(t+k|t) + y_{1opt}(t+k|t) \\ y_2(t+k|t) = y_{2base}(t+k|t) + y_{2opt}(t+k|t) \end{cases} \quad (4.23)$$

com:

$$\begin{bmatrix} y_{1otm}(t+1|t) \\ y_{1otm}(t+2|t) \\ \vdots \\ y_{1otm}(t+N_2|t) \end{bmatrix} = \begin{bmatrix} h_1^{11} & 0 & 0 & \dots & 0 \\ h_2^{11} & h_1^{11} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{N_2}^{11} & h_{N_2-1}^{11} & h_{N_2-2}^{11} & \dots & g_{N_2-N_u+1}^{11} \end{bmatrix} \begin{bmatrix} \delta u_1(t|t) \\ \delta u_1(t+1|t) \\ \vdots \\ \delta u_1(t+N_u-1|t) \end{bmatrix} + \dots$$

$$\dots + \begin{bmatrix} h_1^{12} & 0 & 0 & \dots & 0 \\ h_2^{12} & h_1^{12} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{N_2}^{12} & h_{N_2-1}^{12} & h_{N_2-2}^{12} & \dots & g_{N_2-N_u+1}^{12} \end{bmatrix} \begin{bmatrix} \delta u_2(t|t) \\ \delta u_1(t+1|t) \\ \vdots \\ \delta u_2(t+N_u-1|t) \end{bmatrix}$$

Observações (KEYSER, 2003):

- y_{1otm} indica a parte da saída prevista do processo $y_1(t+k|t)$ vinda tanto da ação de controle ótimo $\delta u_1(t+k|t)$ e $\delta u_2(t+k|t)$.
- Para um sistema 2×2 , 4 resposta ao degrau ou impulso podem ser definidas, as quais descrevem o efeito da mudança para cada uma das duas entradas em relação as duas saídas. Os coeficientes da resposta ao impulso da entrada j para a saída i tem a seguinte notação: $h_1^{ij}, h_2^{ij}, h_3^{ij}, \dots$
- Uma expressão similar a 4.5 pode ser obtida para a segunda saída $y_{2otm}(t+k|t)$.
- O horizonte de predição N_2 pode ser diferente para as duas saídas.

- O horizonte de controle N_u pode ser diferente para as duas entradas.

$$\begin{cases} \mathbf{Y}_1 = \bar{\mathbf{Y}}_1 + \mathbf{G}_{11} \cdot \mathbf{U}_1 + \mathbf{G}_{12} \cdot \mathbf{U}_2 \\ \mathbf{Y}_2 = \bar{\mathbf{Y}}_2 + \mathbf{G}_{21} \cdot \mathbf{U}_1 + \mathbf{G}_{22} \cdot \mathbf{U}_2 \end{cases} \quad (4.25)$$

A expressão é equivalente 4.16 para os sistemas SISO. (KEYSER, 2003)

4.6 EPSAC para um robô móvel com rodas não-holonômico

O EPSAC está sendo utilizado e estudado em diversas áreas, porém não existe a literatura aplicação do mesmo para um controle de trajetória de um robô móvel com rodas. O controlador EPSAC possui características que o torna uma opção viável para o controle de trajetória de um WMR, tais como:

- A necessidade do controlador de um set point ou uma referência, que para WMR se trata da trajetória de referência.
- A possibilidade de trabalhar com sistemas MIMO e não-linear, lembro que o robô móvel é um sistema MIMO com duas entradas, as velocidades linear e angular, e três saídas: posição x , y e a orientação θ . Além da presença de funções trigonométricas caracterizando um sistema não-linear.
- A possibilidade de restrições no controlador, uma vez que o robô possui limites de velocidades que o mesmo pode atingir, esses valores funcionam como restrições no controlador, para um controlador do tipo EPSAC pode ser incluído diretamente nos cálculos.

O controlador tem como objetivo fazer com que o robô siga uma trajetória de referência pré-determinada. Podendo ser formulado da seguinte forma:

$$X(k) - X_r(k) = 0, \quad \text{sendo} \quad X_r = \begin{bmatrix} x_r(k) \\ y_r(k) \\ \theta_r(k) \end{bmatrix} \quad (4.26)$$

na qual $X_r(k)$ é a trajetória de referência. É comumente associado a trajetória de referência um robô de referência *virtual* (KLANCAR *et al.*, 2005), ou seja, um robô que seguiria de forma perfeita a trajetória. Sendo o modelo cinemático desse robô o mesmo do robô a ser controlado:

$$\begin{cases} x_r(k+1) = x_r(k) + v_r(k) \cos \theta_r(k) T_s \\ y_r(k+1) = y_r(k) + v_r(k) \sin \theta_r(k) T_s \\ \theta_r = \theta_r + \omega_r T_s \end{cases} \quad (4.27)$$

onde T_s é o tempo de amostragem. Pode ser reescrita de forma mais compacta como sendo:

$$x_r = (k+1) = f(x_r(k), u_r(k)) \quad (4.28)$$

sendo $u_r(k) = [v_r(k), \omega_r(k)]^T$ são as entradas do controle de referência.

A função custo a ser utilizada é:

$$J = \sum_{k=N_1}^{N_2} [y(t+k|t) - w(t+k)]^2 + \sum_{k=0}^{N_u-1} \lambda [\Delta u(t+k|t)]^2 \quad (4.29)$$

sujeito a:

$$\begin{aligned} y_{\min} &\leq y(t+k|t) \leq y_{\max} \quad \forall k = 1, \dots, N_2, \\ u_{\min} &\leq u(t+k|t) \leq u_{\max} \quad \forall k = 0, \dots, N_u - 1, \\ \Delta u_{\min} &\leq \Delta u(t+k|t) \leq \Delta u_{\max} \quad \forall k = 0, \dots, N_u - 1 \end{aligned} \quad (4.30)$$

onde N_1 e N_2 são os horizontes mínimos e máximo, N_u é o horizonte de controle, λ o peso do controle, $w(t+k)$ *set-point* futuro ou a trajetória de referência, $\Delta u(t)$ é a ação de controle incremental, $\Delta = 1 - q^{-1}$ e $y(t+k|t)$ é a predição k -passo a frente da saída do sistema $y(t)$ no instante de tempo t .

É assumido que a estratégia de controle para o robô móvel é constante, desta forma é definido o horizonte de controle N_u como sendo $N_u = N$, reduzindo os graus de liberdade do controlador.

O cálculo para a determinação do controlador não será feito utilizando as ferramentas de otimização dos programas de cálculos, uma vez que os cálculos de otimização aumentam o custo computacional. Para a simplificação será considerado que o controle ótimo é a primeira linha do vetor:

$$u_0 = -H^{-1}b \quad (4.31)$$

onde :

$$\begin{aligned} H &= 2(P^T Q \lambda P + G^T G) \text{ e} \\ b^T &= 2((P u_b + u_i)^T Q \lambda P + (y_b - \omega)^T G) \end{aligned}$$

Os valores dos termos é apresentado na seção 4.3. Os parâmetros de controle considerados são os seguintes o horizonte de controle N e a matriz de ponderação Q que possui a seguinte forma:

$$\mathbf{Q} = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix}$$

As restrições utilizadas serão os valores máximo e mínimo para as velocidades linear e angular. Sendo os limites para a velocidade linear $v_{min} = -0.4m/s$ e $v_{max} = 0.4m/s$ e para a velocidade angular $\omega_{min} = -0.4rad/s$ e $\omega_{max} = 0.4rad/s$. Desta forma, utilizando as equações do modelo acima e o procedimento apresentado nesse capítulo foi escrito o algoritmo para o controle de trajetória do robô móvel com rodas.

5 Resultados e discussão

Os resultados serão apresentados em duas partes: a primeira contém os resultados obtidos pela simulação computacional de cada controle sugerido, comparações entre os mesmo e os devidos comentários; a segunda parte será apresentada os resultados do controle para o WMR real.

5.1 Resultados de simulação

O controle sugerido foi implementado em uma rotina computacional para o software Matlab®. Uma trajetória foi pré-definida e é uma figura semelhante ao desenho do algoritmo que representa o número oito, sendo composto por duas circunferências de mesmo raio e o mesmo igual a dois $r_c = 2m$ e são tangentes no ponto $(0,0)$. Os valores das velocidades lineares v_{ref} e ω_{ref} utilizados para realizar a trajetória no tempo requerido são os valores de referência, ou seja $u_{base} \triangleq [v_{ref} \ \omega_{ref}]$. A velocidade linear v_{ref} é constante durante toda a trajetória sendo $v_{ref} = 0,3m/s$, a velocidade angular tem o mesmo módulo durante toda a trajetória, porém na metade tem seu sentido invertido, de 0 a $125s$ a velocidade angular $\omega_{ref} = 0,15rad/s$ e a partir de $125s$ até o final com $\omega_{ref} = -0,15rad/s$. A mudança do sinal da velocidade angular indica que a primeira metade da trajetória, correspondente a primeira circunferência, é realizado em sentido inverso ao da segunda metade.

Para todos os estudos de simulação: cada um dos três controladores separados com variações no seus respectivos parâmetros de controle e para uma avaliação conjunta, o robô estará em uma condição de erro de estado inicial, ou seja, o ponto de partida do robô não é a origem da trajetória. No caso será utilizado como ponto de partida o ponto $X_0 = [0; -1; \pi/2]$, posicionado a $\theta = 90^\circ$ em relação ao eixo de referência.

Sendo apresentados, os gráficos da trajetória simulada do robô, da variação das velocidades e do comportamento dos erros para cada valor de saída, x , y e θ . Além da análise gráfica foi definido quatro parâmetros de índices de desempenho para comparar os algoritmos de controle.

Um índice de desempenho é uma medida quantitativa de desempenho de um sistema de controle, sendo um número positivo ou nulo. Então o melhor sistema é definido como o sistema que minimiza esses índices. Para ter o máximo de informações possíveis do desempenho dos

controladores utilizados foram utilizados quatro índices de desempenho, sendo eles:

- **Integral do quadrado do Erro - ISE (Integral of the Square of the Error).** Este índice irá discriminar sistemas excessivamente superamortecidos dos subamortecidos:

$$ISE = \int_0^T e^2(t) dt$$

- **Integral do valor absoluto do erro - IAE (Integral of the Absolute magnitude of the Error).** Índice utilizado em simulações computacionais:

$$IAE = \int_0^T |e(t)| dt$$

- **Integral do tempo multiplicado pelo valor absoluto do erro - ITAE (Integral of Time multiplied by Absolute of the Error).** Utilizado para reduzir a contribuição de grandes erros iniciais no valor da integral de desempenho, enfatizando os erros que acontecem mais tarde na resposta.

$$ITAE = \int_0^T t|e(t)| dt$$

- **Integral do tempo multiplicado pelo quadrado do erro - ITSE (Integral of Time multiplied by the Squared Error).** Este índice fornece a melhor seletividade dentre os índices de desempenho, uma vez que o valor mínimo da integral é prontamente discernível ao serem variados os parâmetros do sistema.

$$ITSE = \int_0^T te^2(t) dt$$

5.1.1 EPSAC

O controlador EPSAC possui duas variáveis de ajuste de controle o horizonte de controle N e a matriz de ponderação Q , o horizonte de controle deve ser um número inteiro maior que um, ($N \in \mathbb{N}, N > 1$) sendo que Q uma matriz composta de zeros exceto na sua diagonal principal que contém os termos de parâmetros de ajuste de controle, os parâmetros da diagonal da matriz q_{11}, q_{22}, q_{33} podem assumir qualquer valor positivo diferente de zero ($q_{nn} > 0, q_{nn} \in \mathbb{R}$) . Ambos os parâmetros quando aumentados melhoram a resposta do sistema.

A avaliação do comportamento dos parâmetros de ajuste do controlador EPSAC foi dividida em duas partes, sendo cada parâmetro avaliado de forma independente. Na primeira etapa será avaliado o comportamento do horizonte de controle N enquanto a segunda parte é dedicada a avaliação da matriz de ponderação Q .

A primeira etapa será avaliado o comportamento da resposta do sistema em relação ao horizonte de controle N , então a matriz de ponderação Q será mantida constante enquanto será

ajustado apenas o valor de N . Na segunda etapa, o horizonte de controle será mantido constante enquanto será ajustado o valor da matriz de ponderação Q , para simplificação apenas o valor de ponderação referente a q_θ será utilizado como parâmetro de ajuste.

$$Q = \begin{bmatrix} q_x & 0 & 0 \\ 0 & q_y & 0 \\ 0 & 0 & q_\theta \end{bmatrix}$$

Avaliação do comportamento do controlador em relação ao horizonte de predição N

O comportamento do EPSAC em relação ao horizonte de controle N será avaliado para quatro valores, lembrando que $N > 0, N \in \mathbb{N}$, os valores de N avaliado serão $N = 2, N = 3, N = 5$ e $N = 10$. A matriz de ponderação Q foi mantida com seus valores constantes e iguais:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0,1 \end{bmatrix}$$

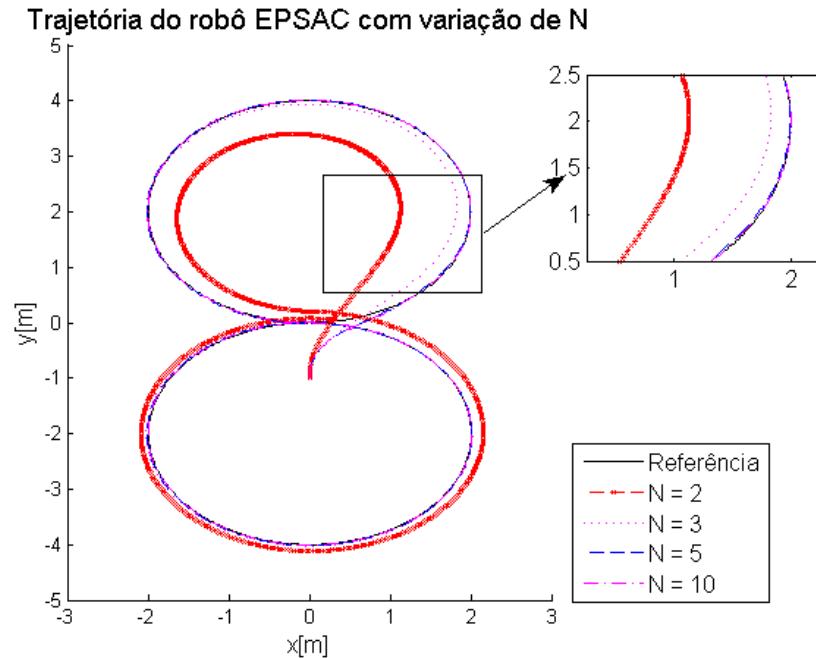
Nota-se na Fig 5.1 a melhora de desempenho do controlador quando comparado com o obtido para $N = 2$ quando os valores de são $N = 3, N = 5$ e $N = 10$, confirmando a hipótese de que a melhoria de resposta do sistema quando utilizando o horizonte de controle como parâmetro de controle depende diretamente do valor de N quanto maior o valor de N melhor será a resposta e maior o tempo necessário para realizar os cálculos.

Para o horizonte de controle $N = 5$ tem-se que após aproximadamente 12s, o robô simulado consegue corrigir o erro inicial e acompanhar a referência, como o apresentado na Fig.5.3(a). Enquanto para $N = 10$ tem-se que o erro tende a zero em aproximadamente 6s, conforme o observado na Fig.5.3(b). Porém, o tempo necessário para serem realizados os cálculos é de 0,92s quando $N = 10$ contra 0,505s necessários sendo o horizonte de controle é $N = 5$.

Avaliação do comportamento do controlador em relação a matriz de ponderação Q

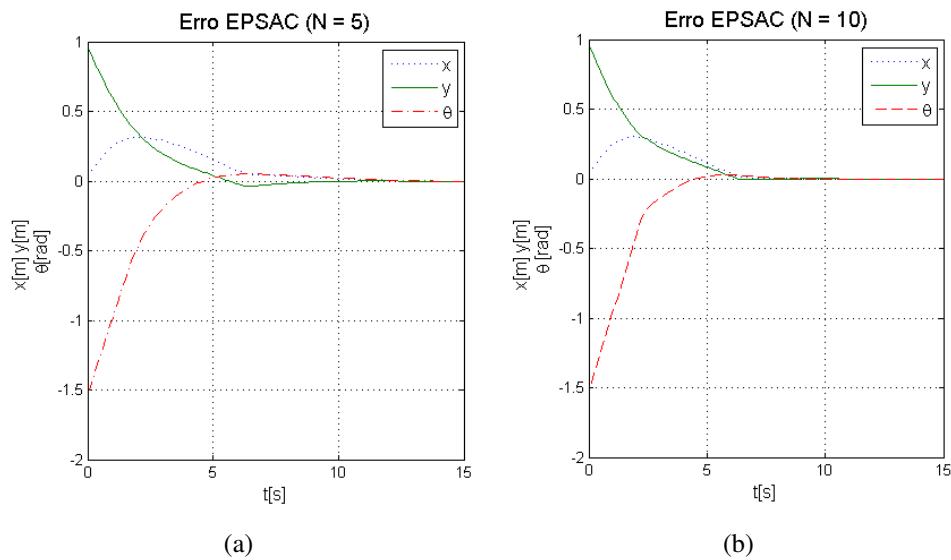
O horizonte de controle N é mantido constante, enquanto é avaliada a influência da variação dos valores na matriz Q , será alterado apenas um elemento dentro da matriz, por motivo de facilitar as comparações futuras e a utilização do controlador, esse parâmetro alterado é denominado como sendo $q_{avaliado}$. O $q_{avaliado}$ é correspondente a ponderação na variável de saída θ , a ponderação das outras duas saídas q_x w q_y será mantida constante e igual a 1. Desta forma, para o estudo do comportamento dos parâmetros de controle do EPSAC, a matriz Q pode ser descrita da seguinte forma:

Figura 5.1: simulação do WMR utilizando EPSAC a matriz de ponderação Q constante e o horizonte de controle N variando



Fonte: Autor

Figura 5.2: a) Erro para o $N = 5$ b) Erro para o $N = 10$



Fonte: Autor

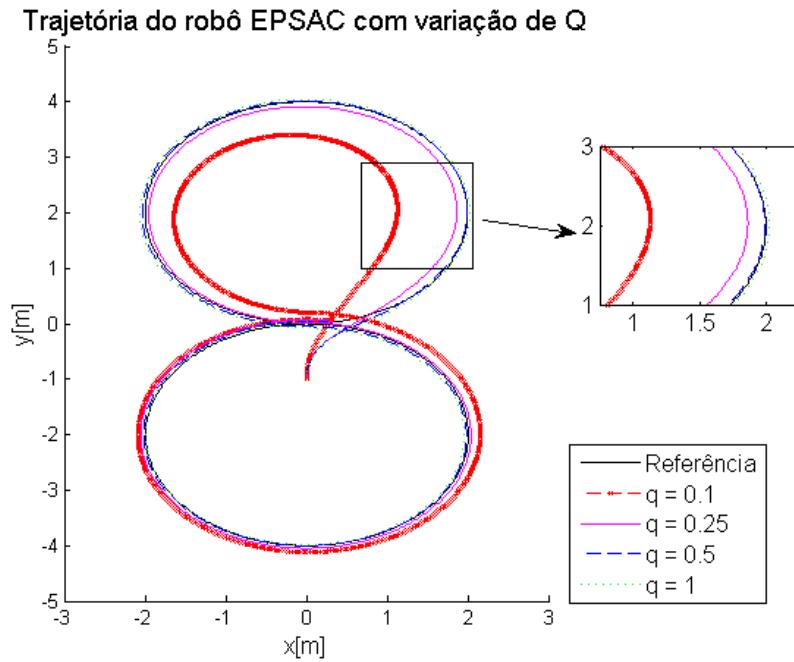
Fonte: Autor

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & q_{avaliado} \end{bmatrix}$$

O valor do horizonte de controle $N = 2$. Os valores avaliados serão $q_{avaliado} =$

0, 1, 0, 25, 0, 5 e 1. O comportamento da matriz de ponderação Q é empírico, em alguns caso quanto maior o valor melhor a resposta em outros ocorre o contrário. A ponderação afeta diretamente a variável de saída correspondente, no caso o θ ; para os três primeiros testes, com o aumento do q_{avaliado} ocorre uma melhoria da resposta, porém quando $q_\theta = 1$ acarreta em um erro maior do que comparado a resposta para $q_\theta = 0,5$, isso ocorre pois o erro passa a ser influenciado pelas outras duas variáveis. O resultado é apresentado na Fig. 5.3.

Figura 5.3: simulação do WMR utilizando EPSAC a matriz de ponderação Q variando e o horizonte de controle N constante



Fonte: Autor

5.1.2 Controlador Klancar

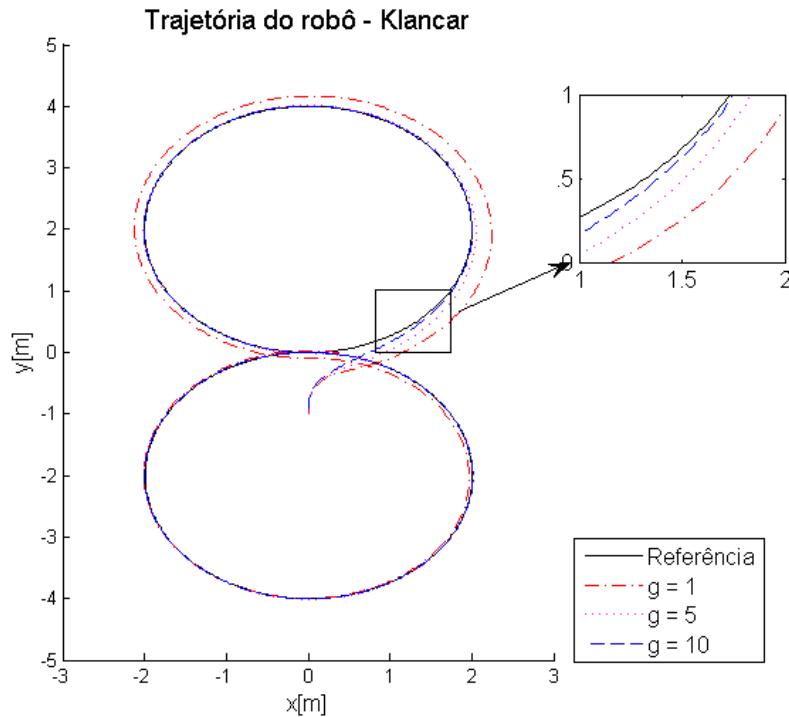
O controlador apresentado na seção 3.1. Tem como parâmetro de controle o ganho g que a única restrição é que g deve ser positivo $g > 0$, para melhorar a resposta do sistema deve modificar o valor de g . Quanto maior o valor de g melhor a resposta do sistema, essa melhoria ocorre até um determinado valor do ganho g , após atingido esse valor o aumento do ganho não acarreta uma melhor resposta do sistema. Esse valor limite depende das condições iniciais do sistema.

Os valores do coeficiente de amortecimento ζ e a frequência natural ω_n são mesmo valores utilizados por Klancar et al (2005) no artigo de referência, $\zeta = 0,6$ e $\omega_n = 2$.

O valor inicial do robô é o mesmo utilizado para simular o controlador EPSAC, sendo a posição inicial a mesma utilizada para o controlador EPSAC $x = [0; -1; \pi/2]$, foi avaliado o g para três valores $g_1 = 1$, $g_2 = 5$ e $g_3 = 10$ e pode-se observar que quanto maior o valor do g

melhor é a resposta do sistema, até um certo valor de ganho g após esse valor não há melhorias consideráveis na resposta, comprovando com a simulação o comportamento da variação do ganho g na resposta do controlador. Os resultados são apresentados para $g = 1, g = 5, g = 10$ na Fig. 5.4.

Figura 5.4: Simulação do WMR para o controlador Klancar variando os valores do ganho g .



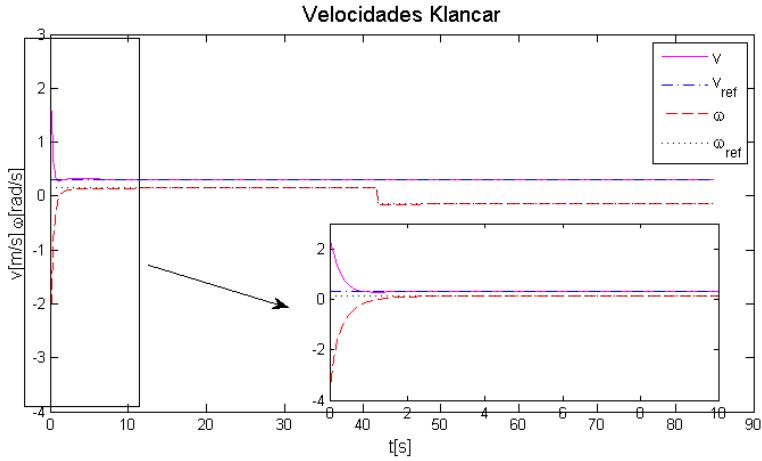
Fonte: Autor

Porém o controlador Klancar atinge velocidades maiores do que foram determinadas como limites para o WMR, na Fig. 5.5 é apresentado as velocidades para $g = 5$ um valor de ganho que possui uma resposta boa do sistema para a correção do erro inicial, porém é observado uma velocidade inicial alta.

Conforme foi mencionado anteriormente, ocorre uma variação no valor da velocidade angular, na metade da trajetória devido a mudança de sentido.

5.1.3 MPC linearizado

O procedimento para avaliar o controlador MPC linearizado proposto por Kühne et al (2004) foi o mesmo utilizado para a avaliação do EPSAC, primeiramente, a matriz de ganho Q é mantido constante e variado o valor do horizonte de controle N , após apenas um valor da matriz de ganho Q será variado, depois o valor de N é mantido constante e será variado apenas um valor da matriz Q pela mesma razão que o procedimento foi realizado para o EPSAC a

Figura 5.5: Velocidades obtidas pelo WMR para o Klancar com o $g = 5$.

Fonte: Autor

questão do custo computacional.

Observando que ambas as estratégias a EPSAC e o MPC linearizado são algoritmos pertencentes ao grupo de controladores MPC e por essa razão apresentam os mesmos parâmetros de controle, a matriz de ganho Q e horizonte de controle N e por uma questão de comparação futura entre os controladores os valores utilizados para a simulação do MPC linearizado inicialmente serão os mesmos que foram simulados para o EPSAC.

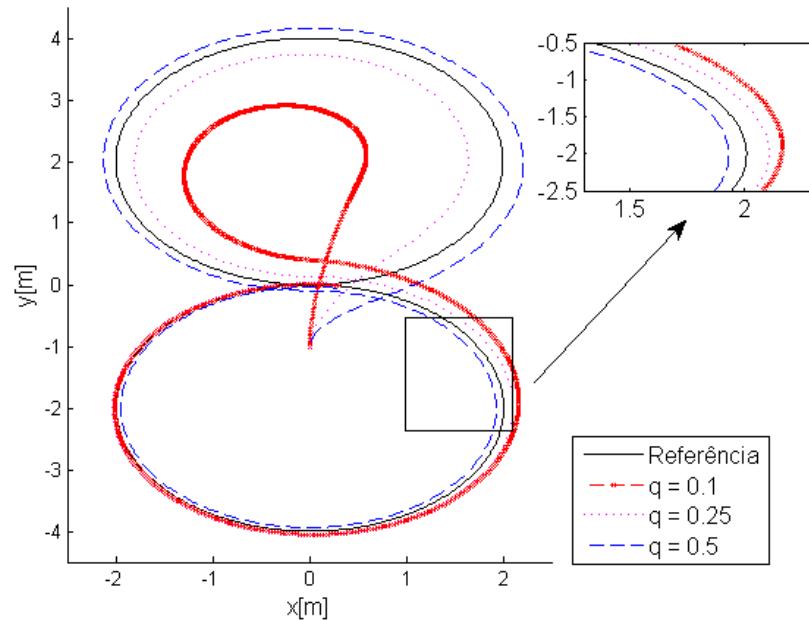
O comportamento dos parâmetros de controle do MPC linearizado também é bastante semelhante ao comportamento do controlador EPSAC, então à medida que se aumenta os valores de N ou da matriz de ganho Q existe uma melhoria na resposta do sistema.

Na Fig. 5.6 é apresentado o comportamento do MPC linearizado quanto variado os valores do termo da matriz de ponderação Q , lembrando que a variação da resposta com a variação dos termos de Q é empírica, quando alterado o valor de $q_{avaliado}$ de 0.1 para 0.25 ocorreu uma melhora na resposta, quando o valor de $q_{avaliado}$ é igual a 0.5 o mesmo não acontece, e tem-se um afastamento da trajetória de referência.

Na Fig.5.7 o comportamento do MPC linearizado em relação ao horizonte de controle N é apresentado, da mesma forma que o EPSAC quanto maior o valor de N melhor é a resposta do sistema. Inicialmente, os valores de N foram os mesmos para a avaliação do EPSAC. Porém, como a resposta do MPC linearizado com $N = 5$ não foi satisfatória, foi aumentado o valor de N sendo avaliado o comportamento do controlador para $N = 10$ e $N = 20$, que em teoria deveria melhorar a resposta do sistema, porém aumenta consideravelmente o esforço computacional envolvido nos cálculos,

Figura 5.6: Simulação do WMR utilizando MPC com a matriz de ponderação Q constante e o horizonte de controle N variando

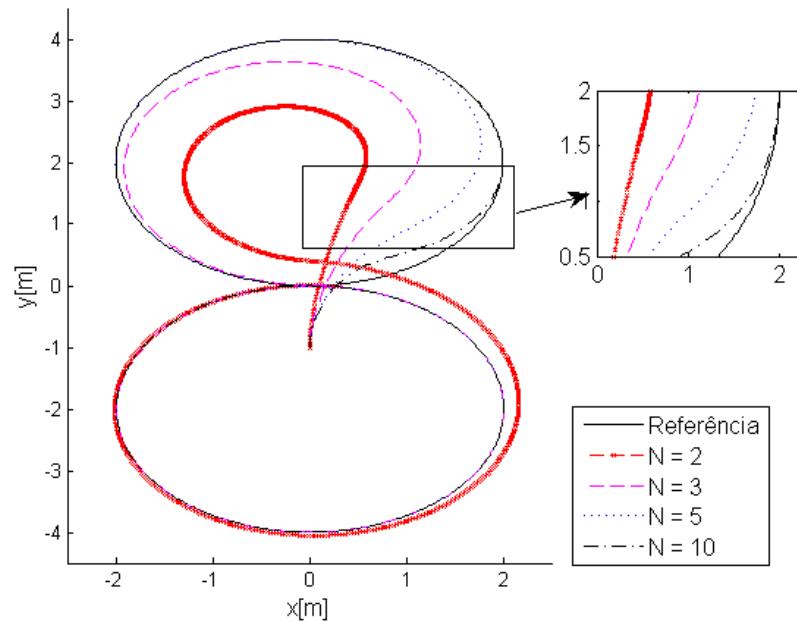
Trajetória do robô MPC com variação de Q



Fonte: Autor

Figura 5.7: Simulação do WMR utilizando MPC a matriz de ponderação Q variando e o horizonte de controle N constante

Trajetória do robô MPC com variação de N



Fonte: Autor

5.1.4 Comparação entre os controladores

O horizonte de controle N , tanto para o EPSAC quanto para o MPC linearizado foi igual para os dois sendo $N = 5$, o valor do ganho do controlador Klancar foi definido como $g = 5.A$

matriz de ponderação Q , também foi a mesma para os dois controladores, sendo:

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0,1 \end{bmatrix}$$

Os valores dos parâmetros de controle para cada controlador analisado foram escolhidos a partir da análise anteriormente apresentado, sendo para o EPSAC os valores de Q e N uma boa resposta uma vez que corrigiu o erro após aproximadamente $T = 12s$ e o tempo necessário para tanto foi de $T_{sim} = 0,505s$, o parâmetro g do Klancar foi o valor que teve uma resposta aproximada ao do EPSAC, porém apresenta um valor alto para as velocidades iniciais. Enquanto o MPC linearizado teve os mesmos valores de Q e N para facilitar a comparação entre os dois. Vale ressaltar que a melhoria do desempenho dos controladores EPSAC e MPC linearizado acarreta em um aumento do custo computacional.

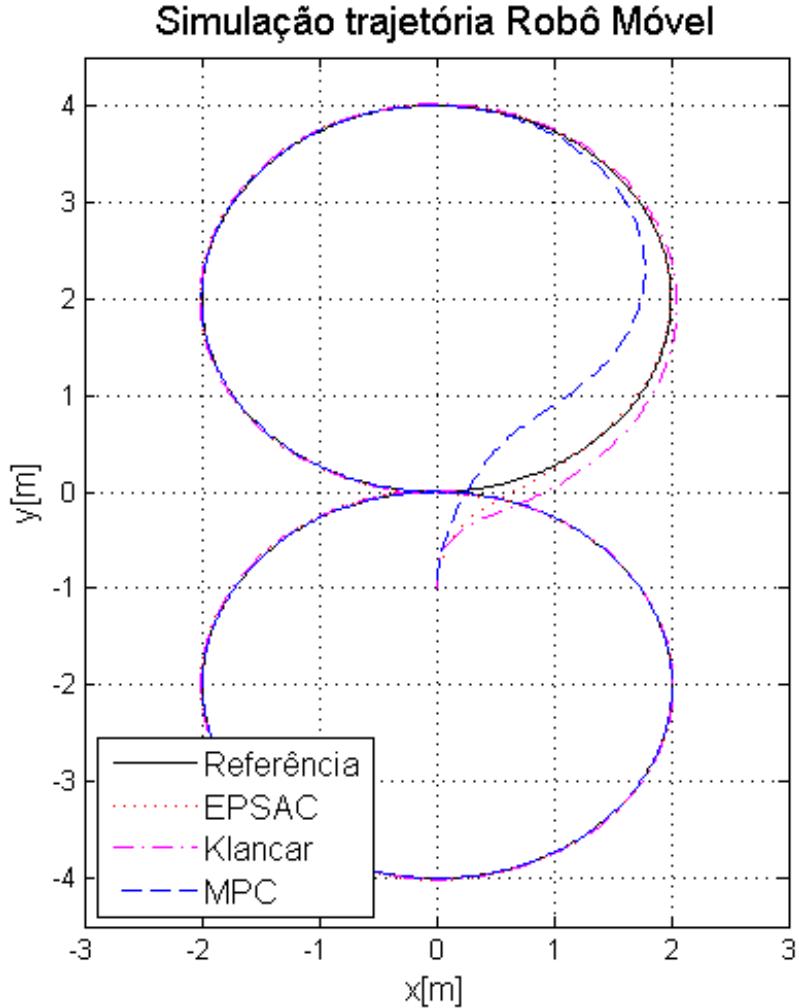
É simulado para uma situação na qual o robô tem um erro no estado inicial $x_0 = [0; -1; \pi/2]$, ou seja, a posição inicial do robô não coincide com a origem da trajetória. Pode-se perceber pela Fig. 5.8 que apesar do erro inicial todos os controladores conseguem fazer a correção e o robô segue a trajetória, porém a correção da trajetória quando utilizado o EPSAC é mais rápido do que para o Klancar (KLANCAR *et al.*, 2005) e o controlador MPC linearizado (KÜHNE *et al.*, 2004) apresenta, para a situação e utilizados, a pior resposta dentre os três controladores.

É considerado que robô possui uma faixa de operação tanto para as velocidades lineares e angulares, os limites superiores e inferiores das velocidades funcionam como restrições para os controladores MPC: o EPSAC e o MPC linearizado, (KÜHNE *et al.*, 2004). Esses valores são levados em consideração para a determinação dos respectivos controladores, porém o Klancar (KLANCAR *et al.*, 2005) não apresenta como é tratado as restrições na determinação dos valores de sinais de controle. Desta forma, na simulação é percebido que existem valores de velocidades acima das velocidades máximas desejada.

Na figura 5.9 é observado o pico de velocidade obtido pelo controlador Klancar, enquanto os outros dois variam as velocidades máximas, dentro da faixa pré-estabelecida. Após, aproximadamente, $T = 25s$ todos os três controladores analisados estabilizam a velocidade linear em torno de $0,3m/s$ que é o valor da velocidade de referência, por essa razão é o intervalo apresentado no gráfico. A velocidade angular estabiliza em torno de $0,15rad/s$ até a metade das amostras, após a segunda metade a velocidade estabiliza próximo a $-0,15rad/s$. Sendo próximo aos valores das velocidades angulares de referência de cada trecho, que são exatamente $0,15rad/s$ e $-0,15rad/s$ para o primeiro e o segundo trecho respectivamente.

Após $T = 10s$ pode-se observar pela Fig. 5.10 que os valores dos erros de posição x e y e de orientação θ do controlador EPSAC converge para aproximadamente zero, na figura também é

Figura 5.8: Saída do controlador para o robô simulado - EPSAC, Klancar e MPC linearizado



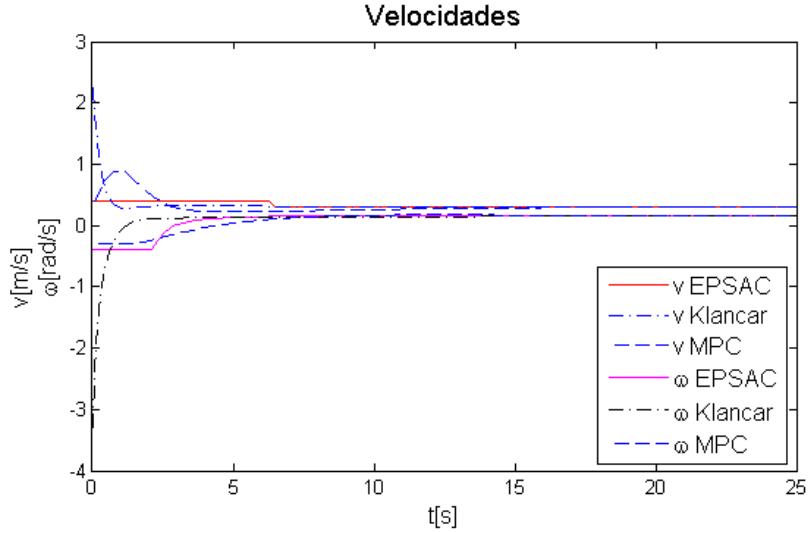
Fonte: Autor

apresentado a variação dos mesmos erros controlador Klancar, pode-se observar a convergência quase que ao mesmo tempo do controlador proposto. E observado que os erros de posição e orientação referente ao controlador MPC linearizado convergem para zero após os outros dois controladores analisados.

Para facilitar a visualização, o comportamento da variação dos erros de posição x e y e de orientação θ é apresentado separadamente o detalhe nos instantes iniciais de $t = 0$ a $t = 40s$ na Fig. 5.11, uma vez, que após aproximadamente 20s todos os erros tendem a zero.

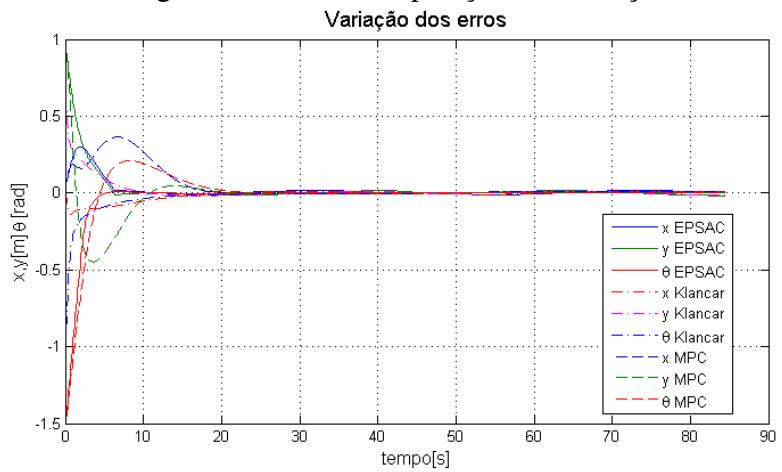
Na tabela tab. 5.1 são apresentados os valores para os índices de desempenho supracitado para cada controlador, vale ressaltar que para cada índice é calculado para cada saída x , y e θ . Pode-se observar como era esperado, que os valores para o EPSAC e Klancar são bastante próximos tento um desempenho semelhante, porém o Klancar, não lida com as restrições diferentemente do EPSAC, para os valores do horizonte de predição N e a matriz de ponderação Q

Figura 5.9: Valores das velocidades lineares e angulares para cada controlador analisado



Fonte: Autor

Figura 5.10: Erros de posição e orientação

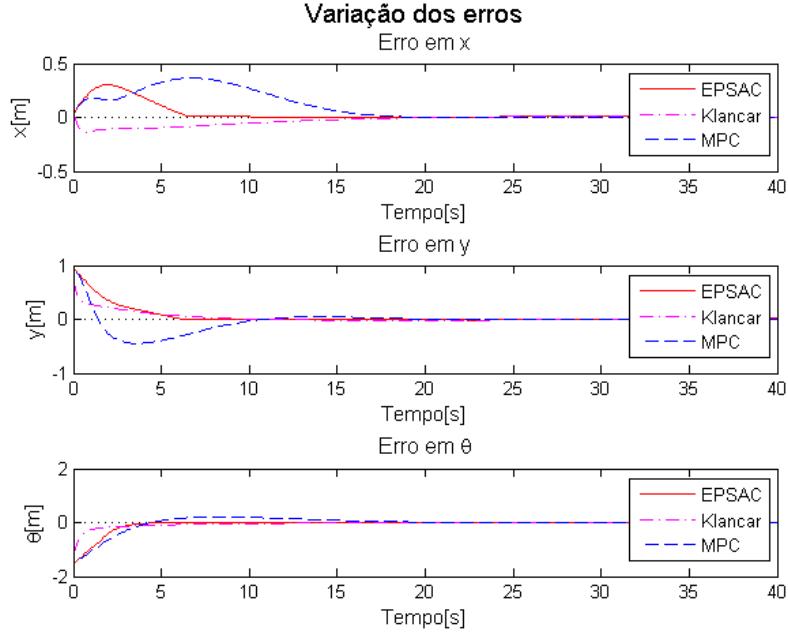


Fonte: Autor

dados o MPC linearizado teve o pior rendimento, aumentando o valor de N , ou algum dos valores da diagonal principal da matriz Q , ou tudo simultaneamente melhoraria a resposta do controlador MPC linearizado, porém o processamento dos cálculos requereria mais do computador, então teria como principal limitante a capacidade de processamento do equipamento utilizado.

Percebe-se pelos valores da tabela que o comportamento do ISE e do IAE são equivalentes, com o controlador Klancar com os valores menores, os valores do EPSAC intermediário, mas próximos ao dos Klancar e os valores para o controlador MPC linearizado bem maiores do que para os dois primeiros. Enquanto para os valores de ITAE e ITSE é observado que uma vez tirado o peso dos erros iniciais, o Klancar teve a melhor resposta, porém o MPC teve uma resposta melhor do que o EPSAC, para os erros de x e y . Para θ o comportamento foi semelhante

Figura 5.11: Erros de posição e orientação - detalhe



Fonte: Autor

ao do ISE e IAE, com a melhor resposta do controlador Klancar, seguido do EPSAC e por último o MPC linearizado.

Tabela 5.1: Índices de desempenho para cada controlador analisado

	ISE			IAE		
	x	y	θ	x	y	θ
EPSAC	2,7424	9,687	21,370	18,112	24,999	26,7745
MPC	9,3722	13,0743	29,9253	35,9114	33,6918	46,9221
Klancar	2,5898	9,7821	21,2422	16,1493	24,1074	26,7365
ITAE			ITSE			
	x	y	θ	x	y	θ
	3448,79	3502,351	2349,998	102,0024	143,353	186,074
MPC	2730,459	1675,99	2445,888	666,7735	397,3005	549,7377
Klancar	2609,599	2735,37	2551,152	82,2633	131,8667	184,6701

5.2 Aplicação no robô real

Os controladores estudados foram aplicados no Robô Móvel da National Instruments modelo NI LabVIEW Robotics Starter Kit, com características apresentadas na sessão 2.4. Uma característica do robô utilizado é que o mesmo utiliza o software Labview, desta forma, os códigos do Matlab foram reescritos para a implementação no robô. O Labview possui requisitos de sistema superiores ao Matlab e os cálculos levam mais tempo para serem realizados.

A partir do Labview é embarcado o algoritmo no robô, assim todos os cálculos são realizados na memória interna do mesmo. O cabo utilizado é conexão entre o robô e um computador para que os dados sejam coletados enquanto é realizado o experimento, seja possível acompanhar em tempo real os valores alcançado e caso ocorra algum imprevisto parar o robô.

Como era esperado pelos resultados preliminares da simulação em Matlab os controladores que utilizam estratégia MPC, no caso o MPC linearizado e o EPSAC, possuem na implementação no robô real um tempo de iteração superiores ao do controlador Klancar. Para o MPC linearizado o tempo de cada iteração foi de $0,250s$ contra $0,030s$ para o Klancar. Desta forma, o tempo de amostragem do MPC foi de $T_{sMPC} = 0,3s$ enquanto os do Klancar o tempo de $T_{sPI} = 0,1s$ (o mesmo valor utilizado durante a simulação). Enquanto o tempo de cada iteração do EPSAC variou entre $0,085 \sim 0,097s$ como é muito próximo do valor inicialmente assumido para o tempo de amostragem ($T_s = 0,1s$) este foi aumentado para $T_{sEPSAC} = 0,2s$.

Por causa da alteração nos tempo de amostragem T_s para cada controlador, a trajetória de referência foi refeita considerando menos pontos. Essa mudança foi realizada para que o robô utilizando qualquer um dos três controladores percorresse a trajetória no mesmo tempo. A proporção utilizada foi a razão entre o tempo de amostragem do controlador T_s do *cont* e o tempo utilizado na simulação, assim $Pontos Utilizados = T_s / 0,1$, desta forma para o EPSAC foi considerado a metade dos pontos da trajetória de referência, e para o MPC um terço.

Os testes foram realizados em um espaço onde o piso é de calçamento irregular, por se tratar de uma área aberta existe a presença de folhas de árvores e outras sujeiras. As condições do piso e a marcação das posições iniciais e da origem da trajetória são apresentadas na Fig. 5.12. Os valores dos parâmetros de cada controlador foram determinados através de diversos testes e foram escolhidos os valores que conseguiram realizar o controle da trajetória sem que houvesse uma necessidade de aumentar o valor do tempo de amostragem.

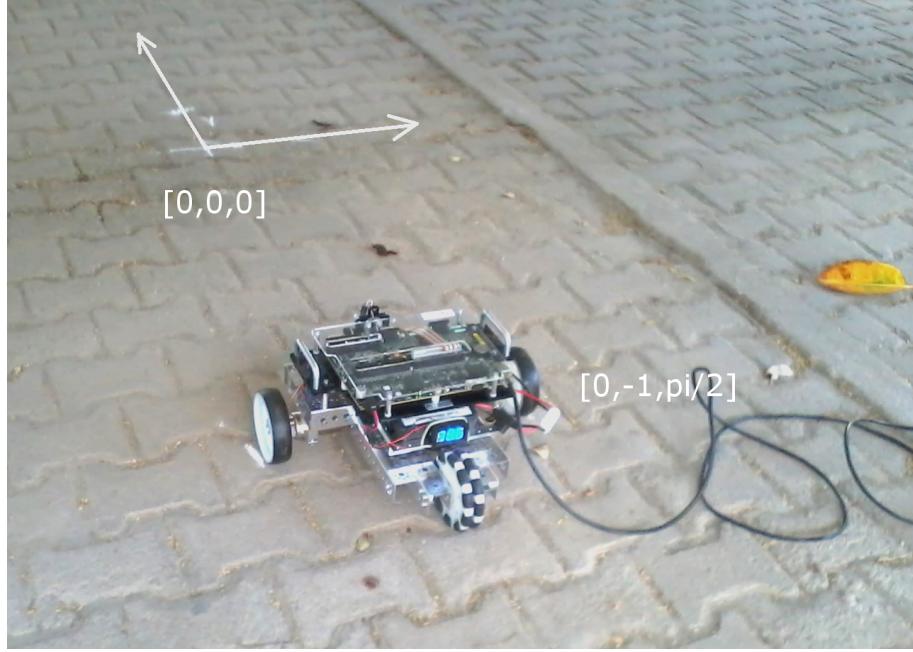
Os dados foram colhidos pelo Labview e exportados em forma de planilhas para ser trabalhado no Matlab.

5.2.1 Controlador Klancar

O tempo de amostragem para o controlador Klancar foi o mesmo tempo utilizado na simulação e portanto a trajetória de referência foi utilizada com todos os pontos inicialmente determinados. A trajetória realizada pelo robô é apresentada na Fig.5.13, pode-se observar que o controlador consegue fazer com que o robô siga a trajetória mesmo com o erro de estado inicial. O parâmetro do controlador utilizado foi $g = 5$.

Os valores de entrada do controlador referente as velocidades lineares são apresentados nas Fig 5.14, pode se observar que a velocidade linear após 10s tende ao valor da referência

Figura 5.12: Robô no local de teste



Fonte: Autor

velocidade de referência. Vale ressaltar que a velocidade linear de referência é constante durante toda a trajetória.

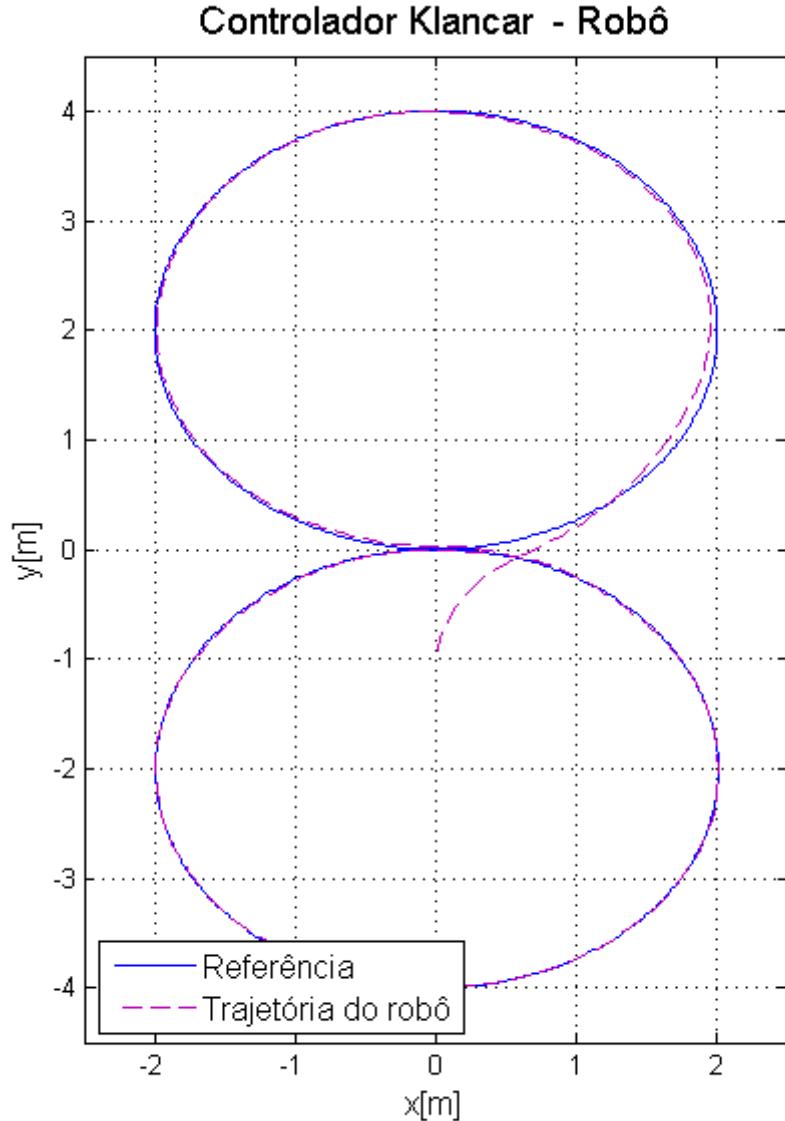
A Fig.5.15 apresenta a variação da segunda entrada do controlador: a velocidade angular. Vale ressaltar que para este controlador não há restrições. Observa-se que após os $T = 10s$ a velocidade angular tende a referência, porém na metade da trajetória ocorre uma mudança no sentido da trajetória provocando uma perturbação. Provocando na velocidade angular um valor de velocidade muito maior do que a referência e logo tende a referência.

A Fig. 5.16, apresenta uma comparação entre os resultados reais e o obtidos quando implementado o controlador Klancar. Pode se observar que as respostas são bastante semelhantes, tendo as variações esperadas quando implementado um controlador na prática. Vale ressaltar que não ouve variação no tempo de amostragem entre a simulação e a implementação.

5.2.2 EPSAC

Para a determinação do valor do controlador EPSAC é realizado a minimização da função custo a cada iteração. O custo computacional para a minimização é superior ao custo do controlador Klancar. O tempo necessário para a realização dos cálculos no Labview chegou ao valor de $0,095s$, desta forma o tempo de amostragem foi corrigido para o valor de $T_{aEPSAC} = 0,200s$. A trajetória foi reduzida a metade dos pontos, ou seja, a cada dois pontos foi retirado apenas um. A trajetória realizada pelo robô é apresentada na Fig.5.17. Os parâmetros utilizados foram

Figura 5.13: Trajetória do robô com o controlador Klancar



Fonte: Autor

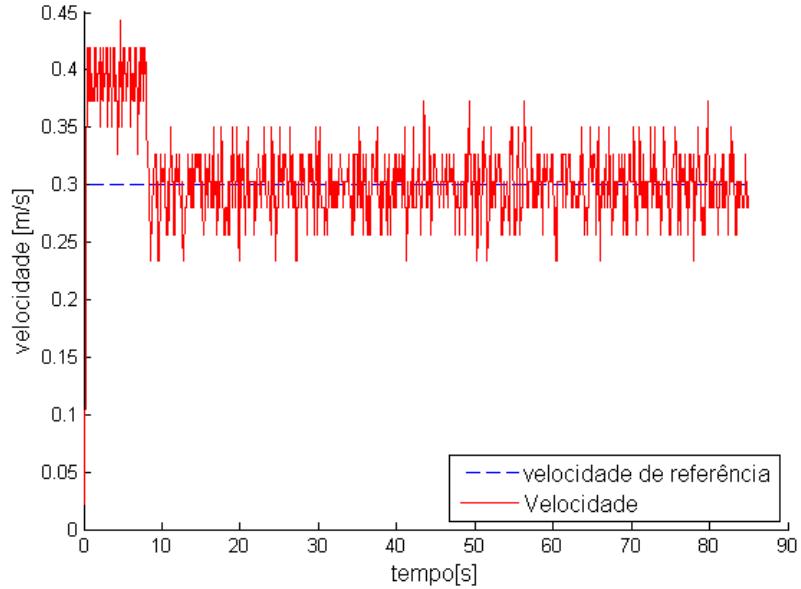
para o horizonte de controle $N = 10$ e a diagonal da matriz de ponderação $diag(Q) = 1, 1e0, 1$.

A velocidade linear é para o controlador EPSAC durante todo o percurso é apresentada na Fig. 5.18, pode-se observar que após aproximadamente 8s a velocidade v_{EPSAC} acompanha o valor da v_{ref} .

Para facilitar a visualização a Fig.5.19 apresenta os instantes iniciais do ensaio. Sendo que no primeiros 8s a velocidade que alcança o valor de aproximadamente $v_{EPSAC} = 0,4m/s$, o valor mais alto permitido pelas restrições colocadas no controlador. Após os oito segundos iniciais a velocidade estabiliza em torno da velocidade de referência, $v_{EPSAC} = v_{ref} = 0,3m/s$.

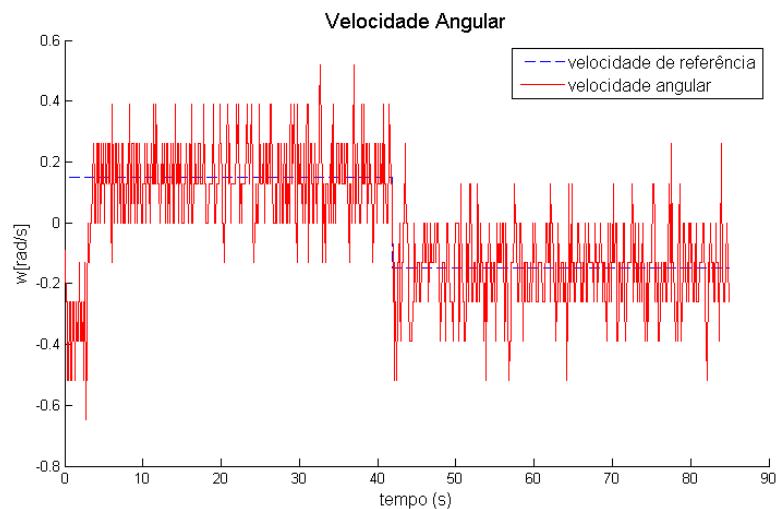
A Fig. 5.20 apresenta a velocidade angular para o controlador EPSAC, após aproximada-

Figura 5.14: Velocidade linear - Klancar
Velocidade Linear



Fonte: Autor

Figura 5.15: Velocidade angular - Klancar
Velocidade Angular

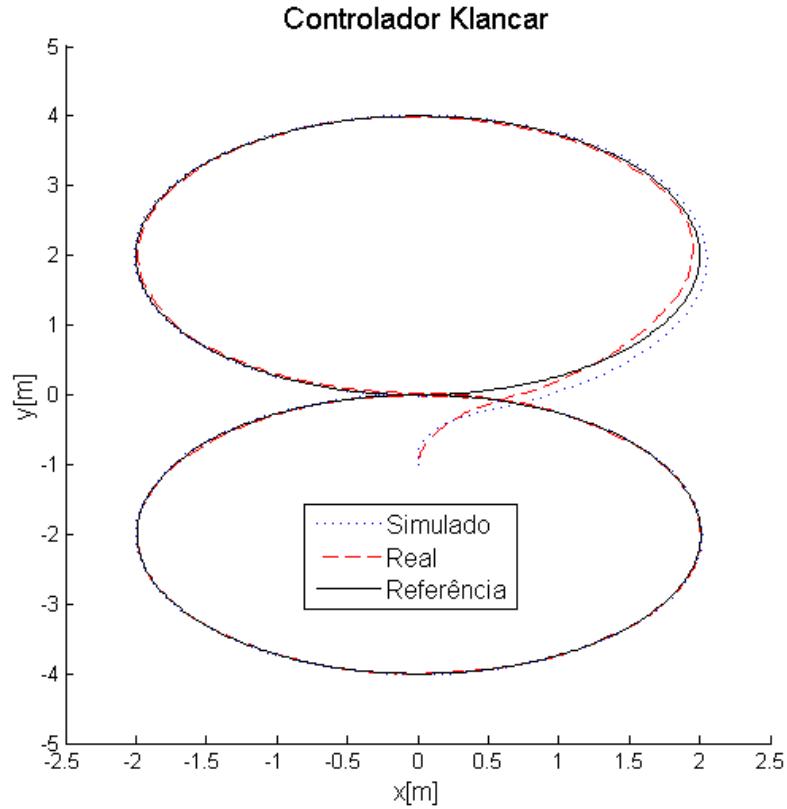


Fonte: Autor

mente 10s a velocidade tende ao valor da velocidade de referência. A variação do sinal que ocorre na metade do tempo da velocidade de referência é devido a mudança de sentido na trajetória do robô.

O robô apresenta durante toda a sua trajetória um erro em x, que não ocorria na simulação um dos motivos é que com a redução de pontos da trajetória aparece um erro na própria geração da trajetória uma vez que estamos aproximando uma circunferência por pequenos segmentos de reta, e no caso esse segmento é duas vezes maior do que o da simulação. O comportamento do

Figura 5.16: Comparação entre os resultados da simulação e da implementação no robô para o controlador Klancar



Fonte: Autor

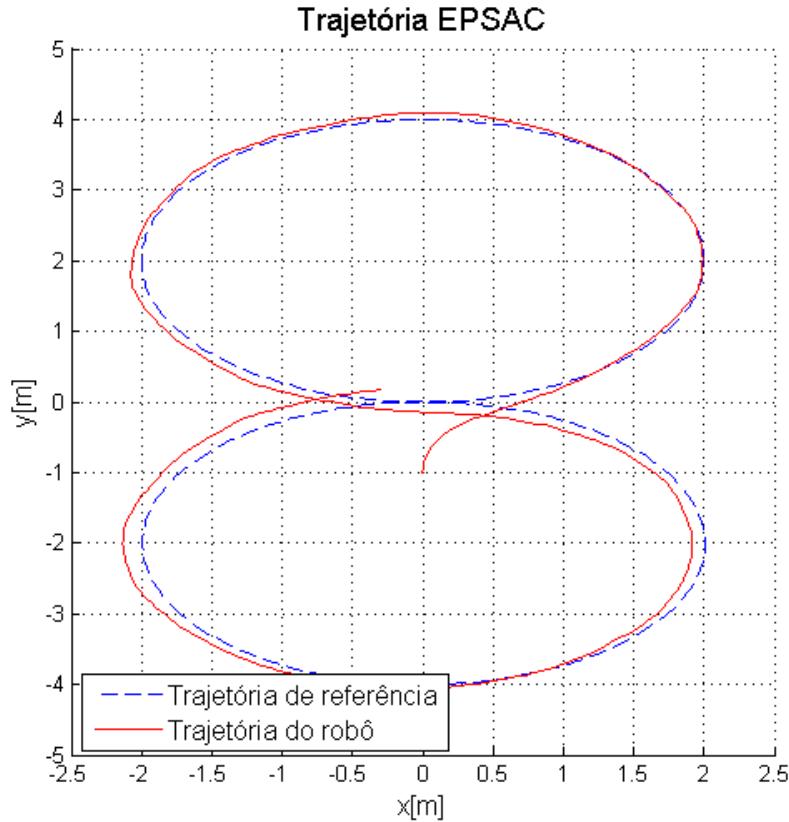
erro é apresentado na Fig. 5.21.

Na Fig. 5.22 é apresentado os resultados real comparado com os resultados de simulação, tendo dois valores para o tempo de amostragem, sendo mantido $T_a = 0,1s$ o valor inicial das simulações e uma simulação com mesmo tempo de amostragem utilizado para a aplicação real $T_a = 0,2s$. Apesar da mudança do tempo de amostragem a simulação não apresenta uma mudança considerável, uma vez que o EPSAC é um controlador preditivo e o controlador considera as referências futuras. A resposta do real, teve um erro um pouco superior ao da simulação, mas esse erro é esperado quando implementado no robô e mostra que na prática que o resultado foi próximo ao esperado.

5.2.3 MPC linearizado

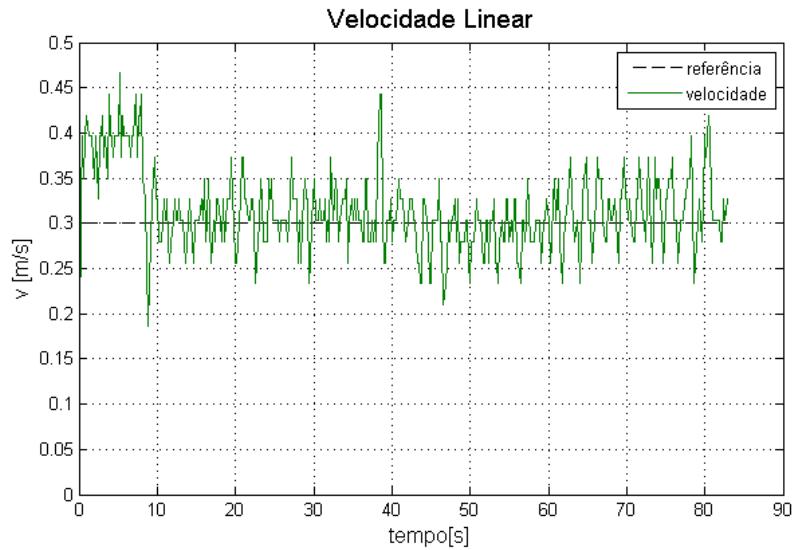
A trajetória realizada pelo controlador MPC é apresentada na Fig. 5.23, vale ressaltar que devido ao alto tempo que o Labview leva para determinar o controle em torno de $250ms$ a quantidade de pontos utilizada pela a trajetória de referência foi reduzida a um terço. Os parâmetros utilizados foram para o horizonte de controle $N = 5$ e a diagonal da matriz de ponderação

Figura 5.17: Trajetória do robô - EPSAC



Fonte: Autor

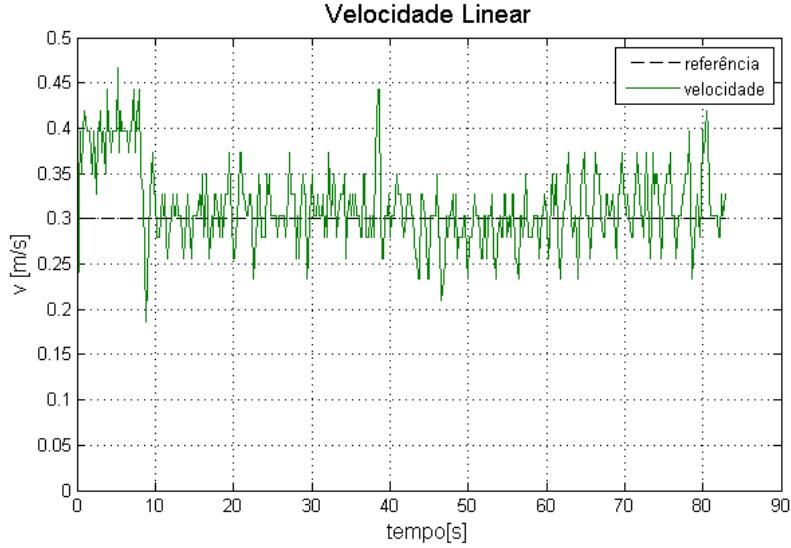
Figura 5.18: Velocidade linear para o robô utilizando controlador EPSAC



Fonte: Autor

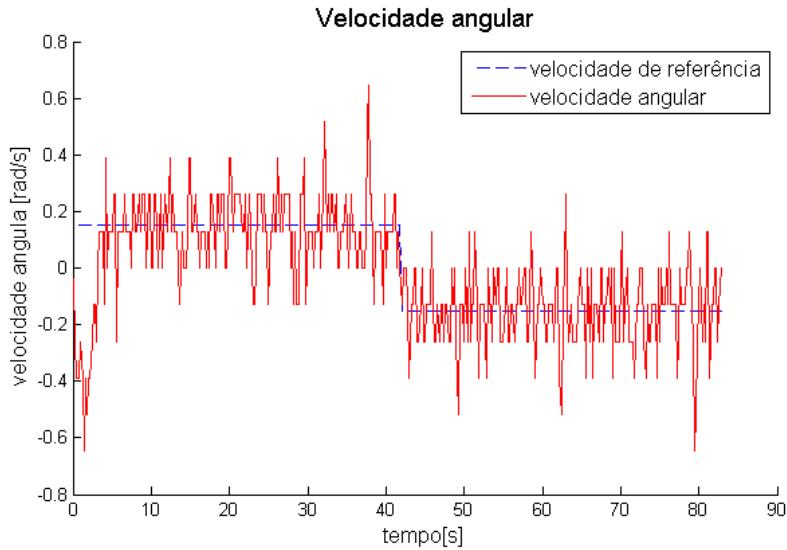
$diag(Q) = 1, 1, 0, 1$. O aumento do valor de N para 7 melhora o desempenho do controlador, mas o tempo para o cálculo de cada iteração chega próximo a 300ms.

Figura 5.19: Velocidade linear para o robô utilizando controlador EPSAC



Fonte: Autor

Figura 5.20: Velocidade angular para o robô utilizando controlador EPSAC

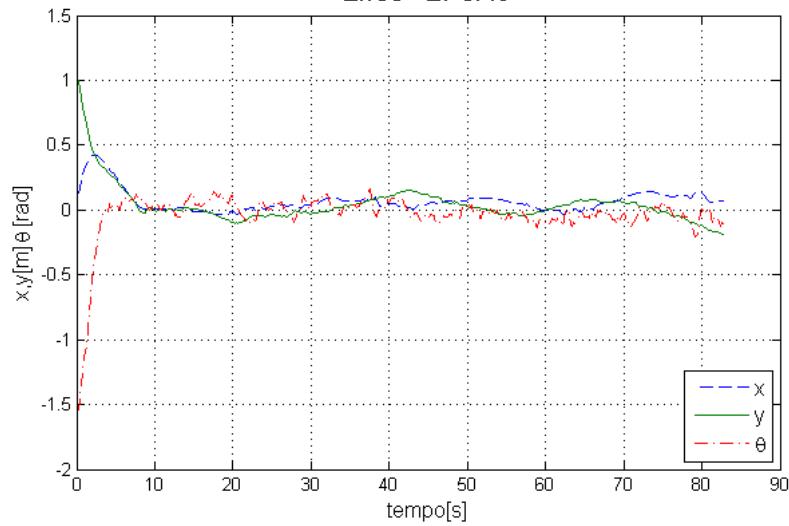


Fonte: Autor

O modelo linearizado é utilizado diretamente no cálculo dos controladores. Em alguns testes no robô foi utilizado o modelo não linear o mesmo dos dois controladores estudados, no caso do robô real ele é utilizado para converter os valores de saída dos sensores (velocidade de cada roda) para os valores de posição, ocorreu uma diminuição no tempo de cálculo, porém foi mantido o modelo linearizado uma vez que foi o apresentado no artigo de referência (KÜHNE *et al.*, 2004).

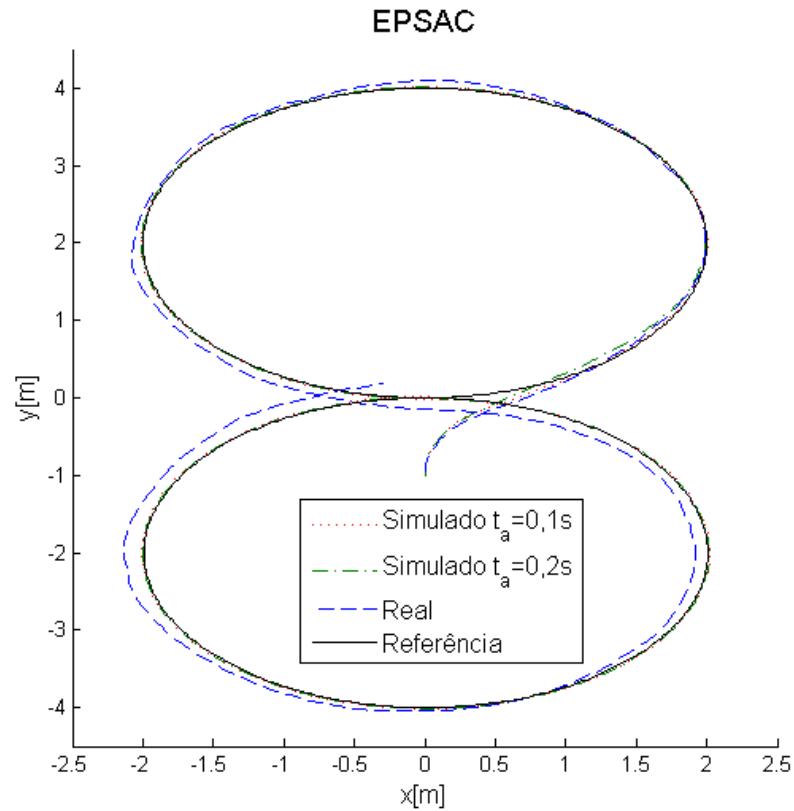
A Fig.5.24 apresenta os valores da velocidade linear, é observado que nos instantes iniciais a velocidade linear é em torno de $v_{MPC} = 0,4m/s$, porém a oscilação de $\pm 0,1m/s$ em torno da

Figura 5.21: Erro de posição e orientação para o controlador EPSAC
Erros - EPSAC



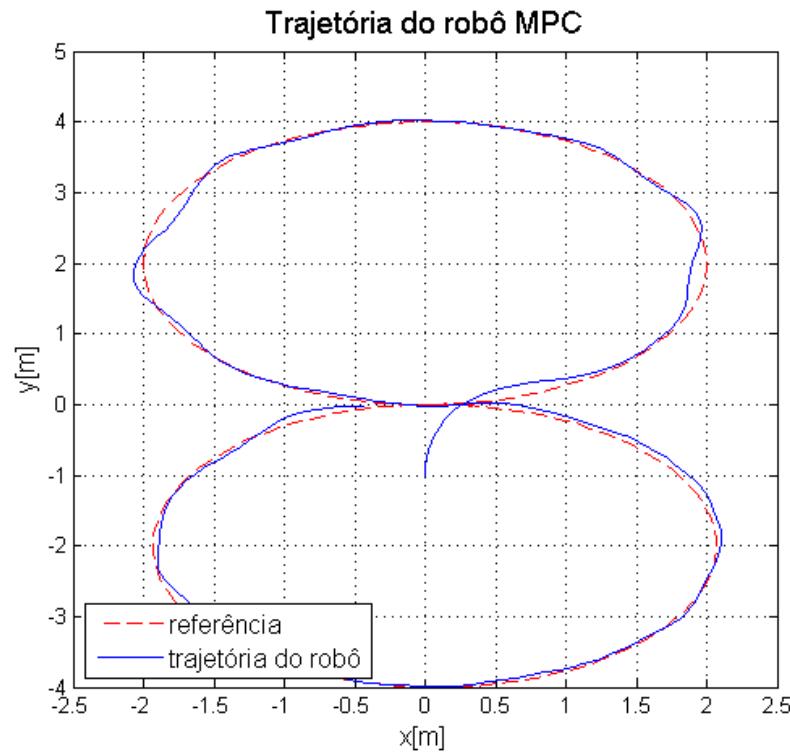
Fonte: Autor

Figura 5.22: Comparação entre os resultados das simulações e da implementação no robô para o controlador EPSAC



Fonte: Autor

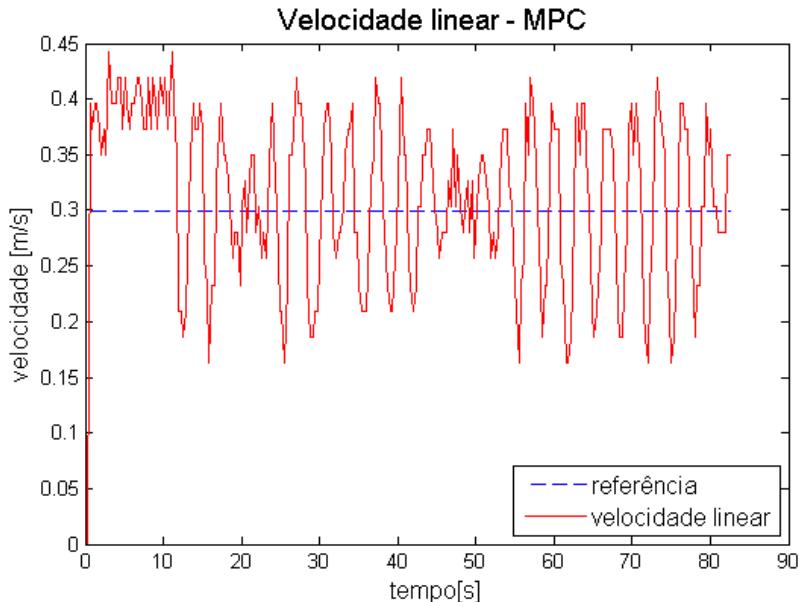
Figura 5.23: Trajetória do robô - MPC



Fonte: Autor

velocidade de referência, sendo a maior amplitude dos três controladores analisados.

Figura 5.24: Velocidade linear para o robô utilizando controlador MPC

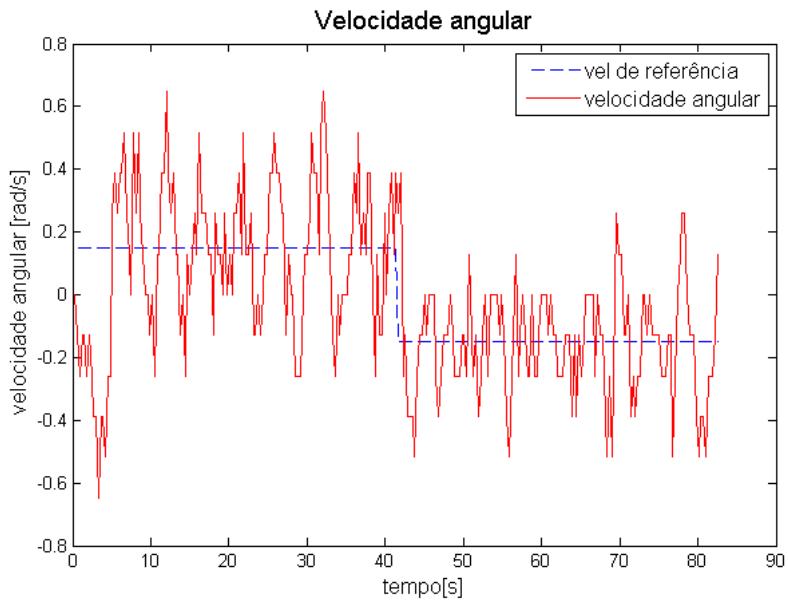


Fonte: Autor

Os valores da velocidade angular é apresentado na Fig. 5.25, observando que a variação em

torno da velocidade de referência é maior do que a dos outros dois controladores: o EPSAC e o Klancar.

Figura 5.25: Velocidade angular para o robô utilizando controlador MPC



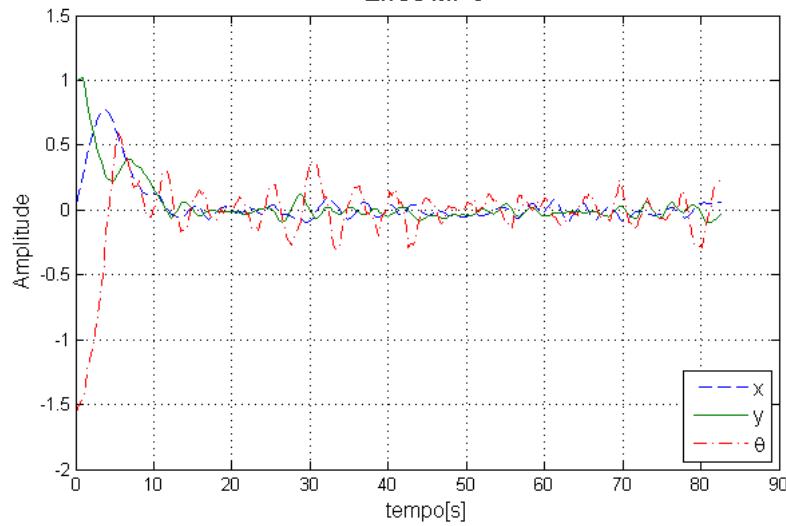
Fonte: Autor

Os erros de posição (x, y) e orientação θ são apresentados na Fig. 5.26. Pode-se observar que os valores dos erros obtidos quando utilizado o MPC é maior do que os valores dos outros dois controladores, essa piora no desempenho do controlador ocorre devido ao fato que este foi o controlador que trabalhou com a menor quantidade de pontos na trajetória devido ao custo computacional do mesmo.

A Fig. 5.27 apresenta as trajetórias simuladas para a amostragem de tempo de 0,1s e 0,3s e a resposta real do controlador. Os tempos de amostragem simulados foram escolhidos, pois $T_a = 0,1s$ foi o tempo utilizado inicialmente na simulação e também era o tempo desejado para realizar a implementação real, como não foi possível devido ao custo computacional o tempo utilizado foi de $T_a = 0,3s$ desta forma foi simulado com esse valor para ser possível a comparação das respostas.

Como o controlador MPC linearizado é um controlador preditivo o aumento do tempo de amostragem não interfere na resposta, uma vez que o controlador considera as referências futuras. Não há uma alteração considerável entre as respostas da simulação, o controlador quando aplicado no robô real, tende a realizar uma trajetória que oscila em relação a trajetória de referência, uma vez que são necessários três amostras (se comparado com o simulado) para o controlador tomar uma decisão desta forma quando a velocidade é inferior o controlador aumenta e quando é coletado novamente os dados o robô está com uma velocidade mais alta do

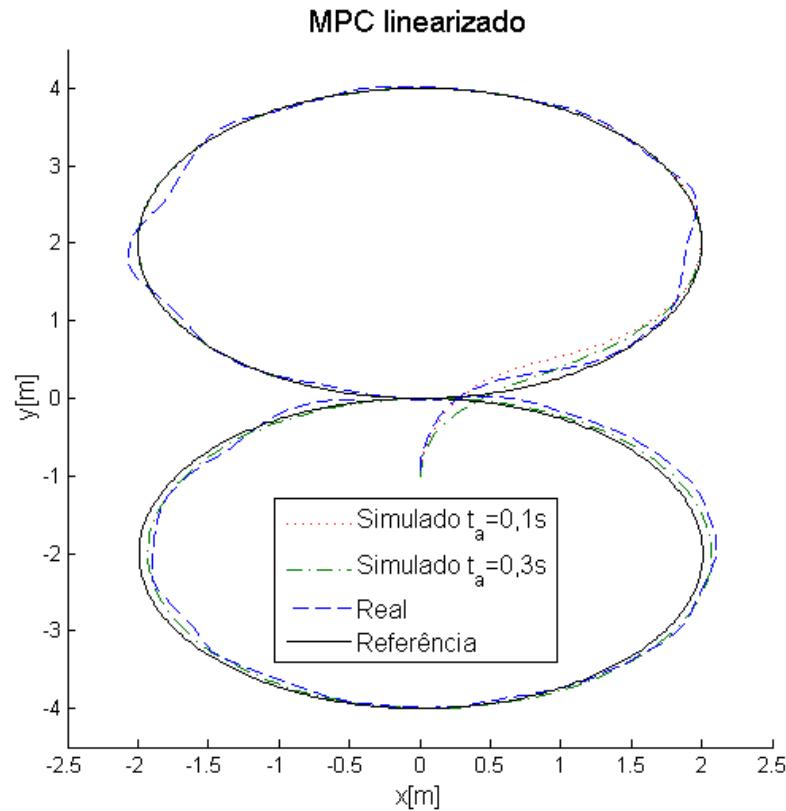
Figura 5.26: Erro de posição e orientação para o controlador MPC
Erros MPC



Fonte: Autor

que o necessário para chegar a referência desta forma é reduzido e o ciclo se repete.

Figura 5.27: Comparação entre os resultados das simulações e da implementação no robô para o controlador MPC



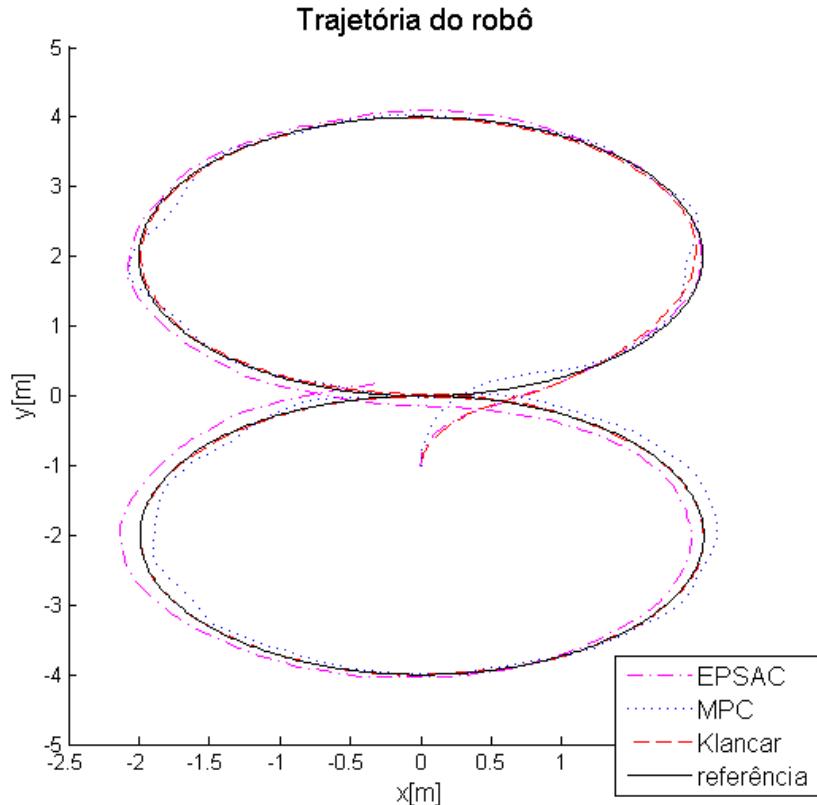
Fonte: Autor

5.2.4 Comparação entre os controladores

A Fig. 5.28 apresenta a resposta da implementação dos controladores no robô. Os tempos de amostragem devido ao custo computacional foi diferente para cada um dos três controladores, sendo para o controlador Klancar $T_{aKlancar} = 0,1s$, enquanto para o controlador EPSAC $T_{aEPSAC} = 0,2s$ e para o MPC $T_{aMPC} = 0,3s$. Para manter as velocidades de referência foi realizado uma correção na trajetória, na qual para o EPSAC é considerado apenas metade dos pontos, as velocidades referentes aos índices (1,3,5 ...) e para o MPC é utilizado apenas um terço dos pontos, referentes aos índices (1,4,7,...).

Pode-se observar que o controlador Klancar teve nessa situação a melhor resposta, porém como foi apresentado na Seção 5.2.1 não existe restrições para as entradas do controlador. O controlador EPSAC desenhou a trajetória porém, em especial para a segunda metade da trajetória, com valores de $x_{EPSAC} < x_{ref}$, enquanto a trajetória quando implementado o controlador MPC ficou oscilando em torno da trajetória de referência.

Figura 5.28: Comparação entre os resultados da implementação no robô dos controladores Klancar, EPSAC e MPC



Fonte: Autor

6 Conclusão

As leis de controle analisadas conseguiram realizar o controle de trajetória mesmo em uma situação na qual o robô tenha um erro de estado inicial, fazendo com que os erros na posição x e y e orientação θ do robô tende a zero. Quando foram implementadas no robô real teve que ser realizado alguns ajustes devido ao tempo que era levado para realizar os cálculos.

O controlador preditivo do tipo EPSAC, utilizado em diversas aplicações, se mostrou uma boa alternativa para o controle de trajetória de robôs móveis com rodas, tendo a capacidade de corrigir erro de estado inicial fazendo com que após algumas os erros de posição x , e e orientação θ tendam zero. Quando implementado no robô real, com os parâmetros utilizados, o EPSAC conseguiu controlar a trajetória do robô real, mesmo com o erro de estado inicial e presença de incertezas, tais como: o chão irregular, a presença de obstáculos. Além de erros no modelo uma vez que foi considerado que o robô tem apenas rolamento não ocorrendo deslizamento durante a movimentação.

É possível melhorar a resposta do controlador ajustando os parâmetros de controle, em especial, o horizonte de controle N , levando em consideração que essa melhora ocorre com um aumento no tempo de cálculo do controlador. Desta forma, a maior limitação desse controlador não é no algoritmo e sim na capacidade de processamento do *Hardware* envolvido. Necessitando de um processador mais potente e de um sistema com uma grande capacidade de armazenamento.

O Controlador Klancar é composto apenas de uma matriz de ganhos com um único parâmetro de ajuste, para a determinação do controlador é realizado cálculos envolvendo multiplicação da matriz de ganho pelo vetor dos erros. É o que possui o menor tempo de amostragem. Porém, para o erro dado não é possível melhorar a resposta do mesmo. Além disso ele originalmente não possui tratamento para restrições, desta forma pode impossibilitar o uso do controlador dependendo do robô utilizado.

O controlador MPC linearizado é entre três analisados o com maior tempo para o cálculo de cada interação, desta forma teve o maior tempo de amostragem e menos pontos na trajetória ocorrendo a partir desse fato um erro. Além, disso no início ele passa 0.3 segundos (ou 3 amostras) sem ação. As respostas poderiam ser melhoradas alterando os valores dos parâmetros

de ajuste do contro, em especial o valor do horizonte de controle N . Porém, isso elevaria o tempo de cálculo.

O modelo do robô é um modelo MIMO com duas entradas e três saídas, os controladores Klancar e EPSAC utilizam um modelo não linear do robô esse modelo é obtido através das equações da cinemática de primeira ordem. Enquanto, o MPC linearizado utiliza o modelo linearizado em torno da referência, porém esse controlador apresenta uma redução de desempenho quanto mais distantes o robô encontrar-se da trajetória de referência.

O fator limitante dos controladores preditivos avaliados foi custo computacional, indicado pelo tempo necessário para a realização dos cálculos, quando aplicado no robô esse custo foi elevado pelas limitações do *Hardware* quanto pelo programa envolvido para realizações do cálculo. Uma vez, que o robô utilizado é habilitado apenas para o *Labview* que tem requisitos de sistemas maiores do que o *Matlab* utilizado nas simulações. Desta forma, para um robô que utilize uma plataforma diferente os parâmetros de controle podem ser mais elevados permitindo uma melhoria na resposta.

Vale ressaltar que o robô tem apenas encoders em cada roda, sendo uma proposta de trabalho futuro a utilização de outros sensores para a determinação de posição e destas forma ser determinado o controle. Outra proposta de trabalho futuro é a redução do custo computacional dos controladores preditivos. Além disso, não foi realizado o estudo de robustez para cada controlador sendo uma outra sugestão de trabalho.

Referências Bibliográficas

- AIBO, S. **Sony Aibo — Sony Aibo Image Gallery**. 2013. Disponível em: <<http://www.sony-aibo.co.uk/sony-aibo-image-gallery/>>.
- AL-ARAJI, A.; ABBOD, M.; AL-RAWESHIDY, H. Design of an adaptive nonlinear pid controller for nonholonomic mobile robot based on posture identifier. In: **Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on**. [S.l.: s.n.], 2011. p. 337–342.
- ANDRADE, F. V. de; TORRICO, B. C. Controle preditivo gpc multivariável de trajetória de um robô móvel. In: **VII Seminário Nacional de Controle e Automação**. [S.l.: s.n.], 2011.
- BATLLE, J.; BARJAU, A. Holonomy in mobile robots. **Robotics and Autonomous Systems**, v. 57, n. 4, p. 433 – 440, 2009. ISSN 0921-8890. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0921889008000833>>.
- BEKEY, G. **Autonomous Robots: From Biological Inspiration to Implementation and Control**. [S.l.]: MIT Press, 2005.
- BEMPORAD, A.; MORARI, M. Robust model predictive control: A survey. In: **Robustness in identification and control**. [S.l.]: Springer, 1999. p. 207–226.
- BRITO, J. O. *et al.* Pi multivariable control applied to temperature and humidity neonate incubators. In: **INDUSCON - IX Conferência Internacional de Aplicações Industriais, 2010, São Paulo**. [S.l.: s.n.], 2010.
- BUHMANN, J. *et al.* The mobile robot rhino. **AI Magazine**, v. 16, n. 1, 1995.
- CAMACHO, E.; BORDONS, C. **Model Predictive Control**. [S.l.: s.n.], 1998.
- COOK, G. **Mobile Robots: Navigation, Control and Remote Sensing**. [S.l.]: Wiley, 2011.
- CRAIG, J. J. **Introduction To Robotics: Mechanics And Control, 3/E**. [S.l.]: Pearson Education, 2008. ISBN 9788131718360.
- FIERRO, R.; LEWIS, F. Control of a nonholonomic mobile robot using neural networks. In: **Intelligent Control, 1995., Proceedings of the 1995 IEEE International Symposium on**. [S.l.: s.n.], 1995. p. 415–421. ISSN 2158-9860.
- FIGUEIREDO, L. C.; JOTA, F. G. Introdução ao controle de sistemas não holônicos. **Sba: Controle Automático**. Sociedade Brasileira de Automática, scielo, v. 15, p. 243 – 268, 09 2004. ISSN 0103-1759. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592004000300002&nrm=iso&_t>.
- GARCIA, E. *et al.* The evolution of robotics research. **Robotics Automation Magazine, IEEE**, v. 14, n. 1, p. 90–103, 2007. ISSN 1070-9932.

- HEDJAR, R.; ALSULAIMAN, M.; MUTEB, K. One-step-ahead predictive control of steering mobile robot. In: **Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on**. [S.l.: s.n.], 2009. p. 638–641.
- HODREA, R. *et al.* Nonlinear epsac predictive control of blood glucose level in diabetic patients for prandial disturbance rejection. In: **Automation, Quality and Testing, Robotics, 2008. AQTR 2008. IEEE International Conference on**. [S.l.: s.n.], 2008. v. 2, p. 180–185.
- HOLKAR, K.; WAGHMARE, L. An overview of model predictive control. **International Journal of Control & Automation**, v. 3, n. 4, 2010.
- INSTRUMENTS, N. **NI LabVIEW Robotics Starter Kit**. [S.l.: s.n.], 2012. 14 p.
- JANG, J. O.; CHUNG, H. T. Neuro-fuzzy network control for a mobile robot. In: **American Control Conference, 2009. ACC '09**. [S.l.: s.n.], 2009. p. 2928–2933. ISSN 0743-1619.
- KANAYAMA, Y. *et al.* A stable tracking control method for an autonomous mobile robot. In: **Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on**. [S.l.: s.n.], 1990. p. 384–389 vol.1.
- KEYSER, R. D. **Model Based Predictive Control**. [S.l.]: EOLSS Publishers, 2003. (UNESCO Encyclopaedia of Life Support Systems (EoLSS)).
- KLANCAR, G.; MATKO, D.; BLAZIC, S. Mobile robot control on a reference path. In: **Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation**. [S.l.: s.n.], 2005. p. 1343 –1348. ISSN 2158-9860.
- KÜHNE, F.; JR, J. M. G. d. S.; LAGES, W. F. Model predictive control of a mobile robot using linearization. In: **4th Mechatronics and Robotics, 2004, Aachen**. [S.l.: s.n.], 2004.
- LUCA, A. D.; ORIOLO, G.; VENDITTELLI, M. Control of wheeled mobile robots: An experimental overview. In: **Ramsete**. [S.l.]: Springer, 2001. p. 181–226.
- MOTOR, H. **Honda Worldwide — ASIMO — Evolution of ASIMO (2011) : Key Specifications**. 2013. Disponível em: <<http://world.honda.com/ASIMO/technology/2011/specification/index.html>>.
- NAGATANI, K. *et al.* Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. **Journal of Field Robotics**, v. 30, n. 1, p. 44–63, 2013. ISSN 1556-4967. Disponível em: <<http://dx.doi.org/10.1002/rob.21439>>.
- NASA, T. G. **MSL - News — NASA**. 2013. Disponível em: <http://www.nasa.gov/mission_pages/msl/news/index.html>.
- ORIOLO, G.; LUCA, A. D.; VENDITTELLI, M. Wmr control via dynamic feedback linearization: design, implementation, and experimental validation. **Control Systems Technology, IEEE Transactions on**, IEEE, v. 10, n. 6, p. 835–852, 2002.
- PIERI, E. R. de. Class Material, **Curso de Robótica Móvel**. 2002.
- POP, C.-I. *et al.* An alternative epsac based control strategy for multivariable time delay processes. In: **Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on**. [S.l.: s.n.], 2012. p. 150–155.

- PÉREZ, J. A. M. *et al.* Estrategias de control en la práctica de anestesia. **Revista Iberoamericana de Automática e Informática Industrial** {RIAI}, v. 8, n. 3, p. 241 – 249, 2011. ISSN 1697-7912. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1697791211000124>>.
- SECCHI, H.; MUT, V. Path following: Discrete stable control of mobile robots. In: **Mechatronics and Automation, 2007. ICMA 2007. International Conference on**. [S.l.: s.n.], 2007. p. 2144–2149.
- SIEGWART, R.; NOURBAKHSH, I. **Introduction to Autonomous Mobile Robots**. [S.l.]: Bradford Book, 2004. 321 p.

APÊNDICE A – Resposta do controlador no Labview

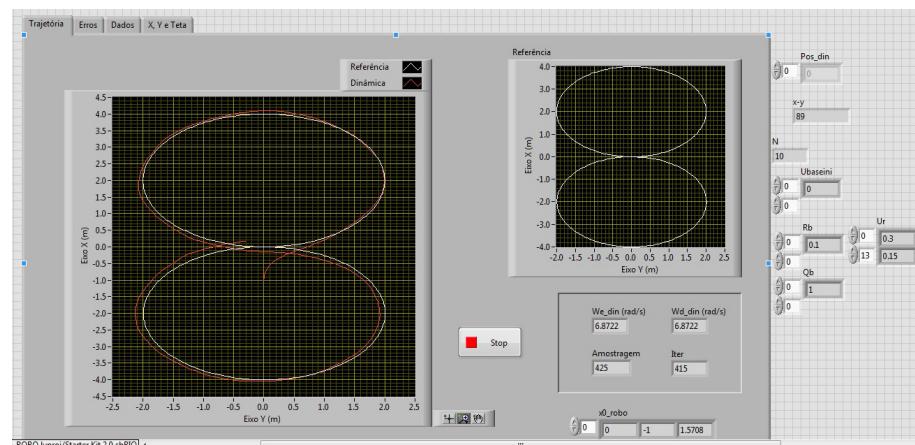


Figura A.1: Tela do Labview para acompanhamento do deslocamento do robô