



Centro Universitário da FEI
Projeto de pesquisa



Projeto de Iniciação Científica

Sistema de controle para o futebol de robôs

Relatório Final

Bolsista: Ronaldo Kazuyoshi Satomi
e-mail: rksatomi@yahoo.com.br

Orientador: Prof. Flávio Tonidandel
Departamento: Ciência da Computação- FEI
e-mail: flaviot@fei.edu.br

05/04/2006

Resumo

O presente projeto visa desenvolver uma parte do projeto de competições robóticas do Centro Universitário da FEI iniciado em 2003. Um sistema de controle para o time futebol de robôs da categoria Mirostot.

O sistema de controle é baseado na cinemática diferencial do robô “Differential Driver”, utilizando um sistema clássico de controle proporcional derivativo (PD), além do Filtro de Kalman. O controle para o time de robôs irá garantir que os mesmos, possam realizar o percurso determinado pela estratégia com o mínimo de erro possível.

Esse estudo utilizou-se de um simulador, para testes e avaliações de seu funcionamento, através de análises gráficas de desempenho.

I. Objetivos

O presente projeto visa desenvolver e implementar um sistema de controle para o time de futebol de robôs da FEI categoria MIROSOT, utilizando dois aspectos simples da cinemática: a velocidade linear e angular. Através de um controle servo-visual utilizando um controlador clássico PD [OGATA, 1997], que reduz os erros acumulados no sistema. Além disso, a utilização do Filtro de Kalman [BITENCOURT, 2003], [GUPTA, MESSOM e DEMIDENKO, 2003] deverá criar maior estabilidade com o aumento da velocidade.

Além disso, será realizado um levantamento de gráficos para análise de erro.

II. Revisão Bibliográfica

II.1. Introdução

Os robôs móveis autônomos são desenvolvidos de forma a terem capacidade de mobilidade e ações independentes, sem a necessidade de ser supervisionado por um operador humano. Esse tipo de sistema está tornando-se gradualmente mais presente dentro das indústrias, no intuito de aumentar a produtividade. Portanto, a necessidade de criar sistemas mais robustos capazes de circular por ambientes com obstáculos, interagir com outros robôs

com grande precisão e eficiência. Em alguns países, há comercialização de robôs que executam pequenas tarefas diárias como: cortar grama, varrer, transporte de objetos dentre outros.

Com o objetivo de difundir e desenvolver mais pesquisas sobre o assunto foi criada a FIRA (Federation of International Robot-soccer Association) [FIRA, 2006], pelo Prof. Jong-Hwan Kim na Korea em 1997, da qual realizam competições mundiais de futebol de robôs e congressos anualmente, ou seja, criando uma nova geração de tecnologia especializada, de modo a desenvolver sistemas de robôs autônomos que cooperam entre si. Isto contribui para o aprimoramento da tecnologia, possibilitando, que estudantes de diferentes regiões e países, possam pesquisar sobre assuntos como controle, inteligência artificial, comunicação, processamento de imagens, robótica, entre outros. A FIRA [FIRA, 2006] define várias categorias de jogos entre robôs, que se diferenciam pelo tamanho e capacidade computacional.

A categoria MIROSOT (Micro Robot World Cup Soccer Tournament), a qual será utilizada como referência para o desenvolvimento desse estudo, possui as seguintes configurações:

- campo é formado por dimensões 150 x 90 cm.
- uma câmera de vídeo CCD.
- um computador.
- sistema de transmissão de dados
- 3 robôs com dimensões máximas de 7,5 x 7,5 x 7,5 cm e sistema de comunicação sem fio.

O sistema funciona da seguinte forma:

A câmera superior capta as imagens, de forma a determinar a posição e direção dos robôs e bola, tudo em tempo real. Essas informações são utilizadas por um sistema de estratégia, definindo a melhor ação e a trajetória que cada um dos robôs deve realizar. O sistema de controle define os comandos a serem enviados a cada robô, para realizarem as ações determinadas pela estratégia, e a enviado-as através de uma comunicação por rádio, como mostra a figura 1.

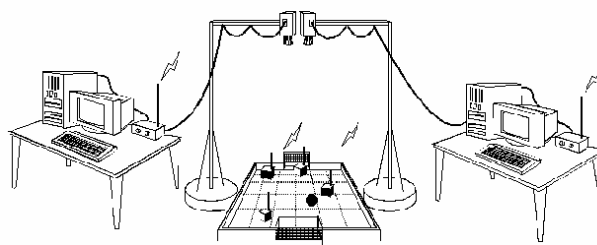


Figura 1 - Modelo de funcionamento

Fonte: FIRA, 2006.

II. 2. Sistema de Controle

Os sistemas de controle podem ser divididos em duas principais categorias, como mostrados por **OGATA [1997]**:

- Controle Clássico.
- Controle Moderno.

Onde o controle clássico, aplica-se a sistemas apenas com uma entrada e uma saída, porém, o controle moderno pode utilizar sistemas com várias entradas e saídas. Atualmente estão sendo desenvolvidos sistemas determinísticos e estocásticos como o controle adaptativo e controle com aprendizado.

II.2.1. Cinemática do Robô

O modelamento matemático é a equação que representa o sistema, podendo ele ser cinemático, dinâmico ou envolver ambos. Para tanto, pretende-se implementar um sistema de controle, que leve em consideração o modelo cinemático, do tipo *Differential Driver*, como os modelos apresentados pôr **GUPTA, MESSOM e DEMIDENKO [2003 e 2004]**, **VIERA ET AL. [2004]**, ao modelo do robô mostrado na figura 2.

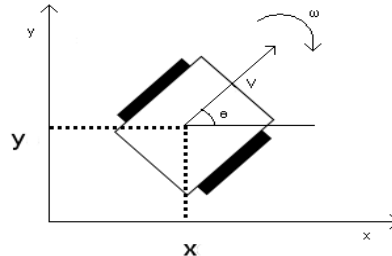


Figura 2 – Cinemática Differential Driver.

Fonte: Viera et al., 2004.

Onde:

ω \Rightarrow Velocidade angular.

Θ \Rightarrow Direção de rotação.

V \Rightarrow Velocidade linear.

x, y \Rightarrow Coordenadas de posição.

Com a variação da velocidade das duas rodas independentemente, podemos obter um movimento linear (equação. I) ou angular (equação II).

$$V = \frac{v_d + v_e}{2} \quad \text{(Equação I)}$$

$$\omega = \frac{v_d - v_e}{2} \quad \text{(Equação II)}$$

Mas pode ser representado da seguinte forma demonstrada por **LUCA, ORIOLO, VENDITTELLI [2001]** na (equação III, IV e V):

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 \\ \sin(\alpha) & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} v \\ \omega \end{bmatrix} \quad \text{(Equação III)}$$

Onde o vetor “q” representa o vetor de posicionamento linear e angular (equação IV), enquanto o vetor “V” representa as velocidades lineares e angular equação (V).

$$q = [x \quad y \quad \alpha]^T \quad (\text{Equação IV})$$

$$V = [v \quad \omega]^T \quad (\text{Equação V})$$

Sendo o objetivo seguir uma trajetória determinada pela estratégia, pode-se determinar as equações diferenciais de, “ $v(t)$ ”, “ $\omega(t)$ ” e “ $\theta(t)$ ” a velocidade linear, a velocidade angular, e o ângulo (orientação) respectivamente em função do tempo, de acordo com a trajetória definida, como segue as equações (VI, VII, VIII), mostradas por **LUCA, ORIOLLO e VENDITTELLI [2001]**.

$$v_d(t) = \pm \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} \quad (\text{Equação VI})$$

A equação VI calcula a velocidade desejada em função do tempo, determinando se o robô irá andar para frente ou para trás, pelo sinal positivo ou negativo respectivamente.

$$\omega_d(t) = \frac{\ddot{y}(t)\dot{x}(t) - \ddot{x}(t)\dot{y}(t)}{\dot{x}^2(t) + \dot{y}^2(t)} \quad (\text{Equação VII})$$

A equação VII calcula a velocidade angular desejada, com as diferenciais de segunda ordem.

$$\theta_d(t) = \left(A \tan^2(\dot{y}(t), \dot{x}(t)) + \kappa\pi \right), \quad \kappa = 0,1; \quad (\text{Equação VIII})$$

A função arco-tangente ao quadrado da equação VIII, calcula o valor do ângulo, os sinais de ambos os argumentos são usados para determinar o quadrante do resultado, a função retorna o resultado em radianos, sendo $k = 1$ quando o robô estiver andando para trás.

II.2.2 Movimento do robô

O robô deve percorrer uma trajetória determinada pela estratégia partindo de uma coordenada inicial (x_i, y_i) , até a coordenada final (x_f, y_f) . O sistema de controle, portanto, atribuirá velocidades para cada roda, de modo que o robô possa realizar o percurso e alcançar o objetivo estipulado como mostra a figura 2.

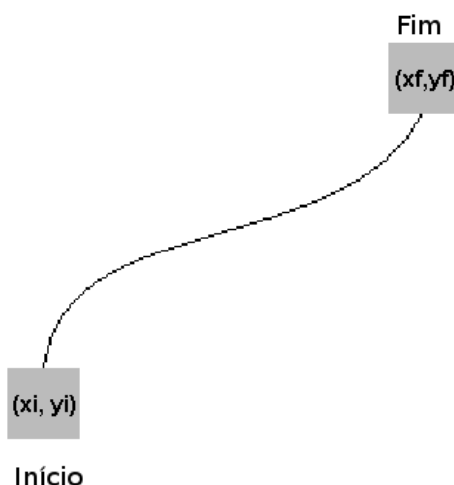


Figura 2 - Seguindo a trajetória.

Entretanto em uma situação real, surgem inúmeros problemas que deverão ser contornados, como obstáculos em movimento, o próprio atrito das rodas com o piso, escorregamento das rodas (patinar), além do dinamismo que exige a competição como a interceptação da bola, empurrar a bola para o gol adversário podendo gerar instabilidade no robô, que por sua vez podem tirá-lo da trajetória.

A solução encontrada foi à utilização de um controle servo-visual (sistema de malha fechada), ou controle de retroação [OGATA, 1997], que permite realizar a trajetória e absorver os erros inerentes das limitações do robô e das situações de jogos, que consiste em comparar o valor da grandeza de saída com o valor da grandeza desejada, de modo a obter o valor do desvio (erro), e gerar um novo valor atenuado do erro, como mostra a figura 3.

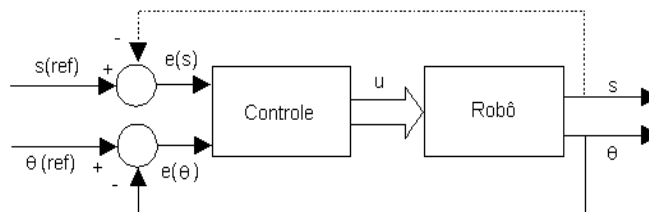


Figura 3 - Diagrama de bloco do sistema de controle.

Fonte: Vieira et al. 2004

Onde:

- $s(\text{ref})$ é a distância de referência, e $\theta(\text{ref})$ é a direção de referência.
- $e(s)$ são os erros em relação distância de referência.
- $e(\theta)$ são os erros em relação direção de referência.
- u é a ação do controle para robô, devido à referência e o erro gerado.

O sistema implementado utilizou-se do modelo apresentado por **GUPTA, MESSOM e DEMIKENKO [2004]**, que é um controlador de posição PD.

Onde P é o ganho proporcional, ou seja, quanto maior o erro, mais rápida a ação do sistema para correção, como mostra a figura 4:

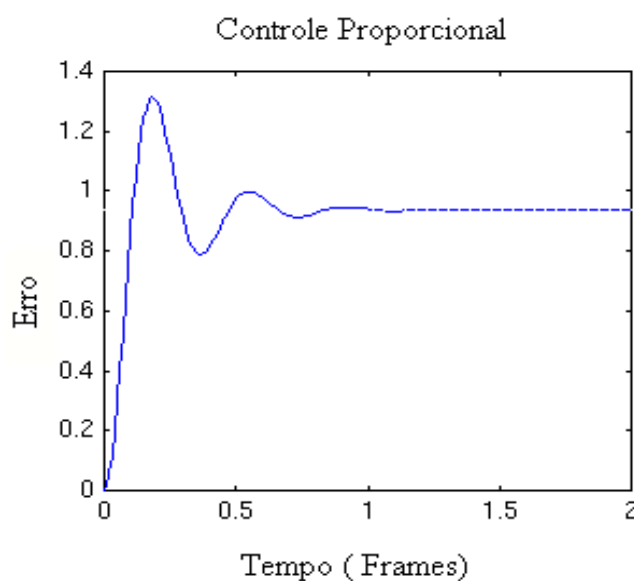


Figura 4 – Ação do coeficiente Proporcional

Fonte: PID Tutorial, 2006.

Porém, somente com o ganho proporcional o sistema torna-se instável (oscilações), então a combinação do ganho derivativo D, ou seja, PD faz com que a instabilidade seja amortecida, como mostra a figura 5:

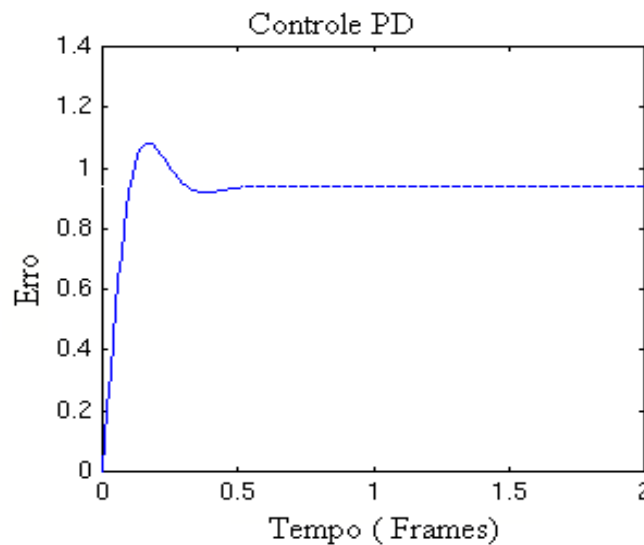


Figura 5 – Controlador PD
Fonte: PID Tutorial, 2006.

Portanto, com a utilização do controle PD pode-se reduzir os erros inerentes do sistema, como demonstraram **GUPTA, MESSOM e DEMIKENKO [2004]**, através da equação IX.

$$V(t) = k_p * e + k_d * \left[\frac{de(t)}{dt} \right] \quad (\text{Equação IX})$$

As equações X e XI representam o modelamento matemático com seus respectivos erros e ganhos mostrado por **GUPTA, MESSOM e DEMIKENKO [2004]**:

$$vr(t) = kd_p . Se + kd_d . ve + ka_p . \theta \quad (\text{Equação X})$$

$$vl(t) = kd_p . Se + kd_d . ve - ka_p . \theta \quad (\text{Equação XI})$$

Através da figura 6, pode-se observar como a determinar os coeficientes de erros dados pelas equações X e XI.

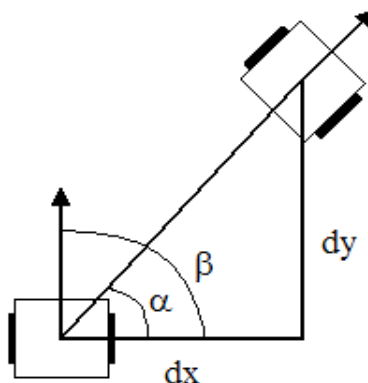


Figura 6 - Controle de Posição
Fonte: Gupta, Messom e Demidenko (2003,2004).

Onde:

$$\theta = \alpha - \beta \text{ (ângulo de erro)}$$

α é o ângulo de referência.

β é o ângulo da direção do robô

$dx = (x_f - x_i)$ (distância entre o robô o objetivo na coordenada x)

$dy = (y_f - y_i)$ (distância entre o robô o objetivo na coordenada y)

$Se = \sqrt{(dx^2 + dy^2)}$ (distância entre o ponto inicial ao final)

$ve =$ (velocidade final – velocidade real)

$d = 0$ (quando o robô gira em torno do eixo)

$d = 1$ (quando o robô realiza um movimento linear ou curvilíneo)

K_{dp} = Coeficiente Proporcional para a distância.

K_{dd} = Coeficiente Derivativo para a velocidade.

K_{ap} = Coeficiente Proporcional para o ângulo.

II.2.3. Filtro de Kalman

O Filtro de Kalman (FK) [BITENCOURT, 2003], [GUPTA, MESSOM e DEMIKENKO, 2003], é um sistema de estimativa linear dinâmico recursivo estocástico, que é capaz de estimar estados do passado, presente e futuro, que evolui de um estado para outro no decorrer do tempo, em decorrência do seu estado atual, eliminando possíveis ruídos. A equação utilizada para este foi apresentada por GUPTA, MESSOM e DEMIKENKO [2003].

Os cálculos de velocidades do robô, descritos nas equações X e XI, dependem da posição atual, mas devido a sua inércia e o tempo de processamento dos computadores, isso pode gerar atrasos, pois, o cálculo é realizado, baseado que o robô está na posição (x_0, y_0) naquele instante, mas na realidade o mesmo está na posição (x_{0+1}, y_{0+1}) , como mostra a figura 7.

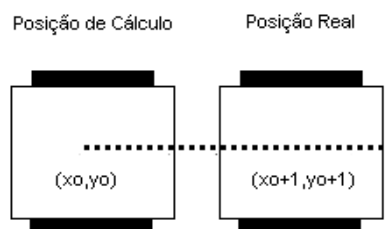


Figura 7 – Posição real do robô

Nesse caso, a utilização do Filtro de Kalman [BITENCOURT, 2003], [GUPTA, MESSOM e DEMIKENKO, 2003], torna-se importante para atenuar o erro existente, em caso de haver atraso no envio de comandos para os robôs, que utilizam as equações (XII e XIII) [GUPTA, MESSOM e DEMIKENKO, 2003]:

$$\dot{\hat{x}}_n = \hat{\dot{x}}_n + \frac{h}{T} (y_n - \hat{x}_n) \quad \text{(Equação XII)}$$

$$\hat{x}_n = \hat{x}_n + g_n (y_n - \hat{x}_n) \quad \text{(Equação XIII)}$$

Onde:

\hat{x}_n : Estimativa com FK da velocidade (linear /angular) em um tempo discreto n.

\hat{x}_{n-1} : Estimativa da velocidade (linear /angular) em tempo discreto n-1.

\bar{x}_n : Estimativa com FK da (posição /direção) em tempo discreto n.

\bar{x}_{n-1} : Estimativa da (posição /direção) em tempo discreto n-1.

T: Período entre Frames.

y_n : Leitura do sensor (câmera) no tempo n.

h e g: parâmetros do FK.

III. Materiais e Métodos

III.1. Recursos humanos e materiais

Todos os equipamentos e softwares encontram-se disponíveis no laboratório de Inteligência Artificial e Robótica (sala D005), que incluem: computadores, Microsoft Visual Studio 2003.NET, simulador Robot Soccer 1.5v, e robôs Mirobot.

III.2.2. Métodos

A implementação do sistema de controle foi desenvolvida em C++, testadas no simulador Robot Soccer 1.5v. Para a realização dos testes, foram criados os seguintes procedimentos:

- Criar programas específicos para determinar os coeficientes de ganho.
- Levantar os gráficos de desempenho dos ganhos.
- Criar um programa que gere uma trajetória em forma de “oito”.
- Criar o programa de controle.
- Levantar gráficos para avaliação de desempenho.

III.2.2.1. Coeficientes ganho

Para determinar o Kap (ganho proporcional em relação ao ângulo), é necessário lembrar que os robôs são simétricos, portanto, tem-se que ação do robô varie entre + 90° e 270°, pois o mesmo, pode andar para frente ou para trás de acordo com a necessidade do jogo.

O ensaio foi realizado de modo a encontrar o melhor fator de escala, para reduzir o erro, com a utilização da tabela 1 determinada por **GUPTA, MESSOM e DEMIKENKO [2004]**:

Ângulo (erro)	Ganho PD
71 - 90	0.2857 * fator de escala
51-70	0.3429 * fator de escala
31-50	0.4 * fator de escala
11-30	0.2857 * fator de escala
0-10	0.1 * fator de escala

Tabela 1 – Cálculo do ganho PD em função do ângulo (erro)

Posicionado-o em uma condição que o robô esteja a 90° de seu objetivo, ele tende a girar em torno de seu próprio eixo, para diminuir o erro existente de direção em que está para o objetivo, a partir disso, pode-se verificar qual o melhor fator de escala através de tentativa e erro que apresente o melhor desempenho, como apresentados por **GUPTA, MESSOM e DEMIKENKO [2004]**.

Para a determinação do Kdp e do Kdd (ganho proporcional em relação à distância e ganho derivativo em relação a velocidade respectivamente), posiciona-se o robô em linha reta ao seu objetivo, e verificam-se, também por tentativa e erro como mostrado por **GUPTA, MESSOM e DEMIKENKO [2004]**, os coeficientes que demonstrem o melhor desempenho.

Os ganhos de h e g do Filtro de Kalman [GUPTA, MESSOM e DEMIKENKO, 2003], são determinados realizando testes, através da trajetória em forma de “oito” onde se verifica qual o melhor coeficiente que estima a velocidade e posição respectivamente.

III.2.2.2. Implementação

Após os levantamentos dos coeficientes de ganho, foi elaborado o seguinte pseudocódigo que representa a forma de funcionamento do sistema.

(pseudocódigo do sistema de controle)

Início.

Receber as coordenadas (x, y) do objetivo e do robô.

Calcular a distância e o ângulo do robô em relação ao objetivo (erros).

Se o erro anterior $>$ erro atual.

{

 Calcula a distância e ângulo (erros) pelo FK.

 Erro atual = Erro FK.

}

Definir ganho PD.

Se direção $(85^\circ >= \text{ângulo} >= 275^\circ)$ // Desloca-se para frente

{

 Calcular as velocidades do robô, em relação aos erros (positiva).

}

Se direção $(265^\circ >= \text{ângulo} >= 95^\circ)$ // Desloca-se para trás

{

 Calcular as velocidades do robô, em relação aos erros (negativa).

}

Caso contrario

{

 Girar.

}

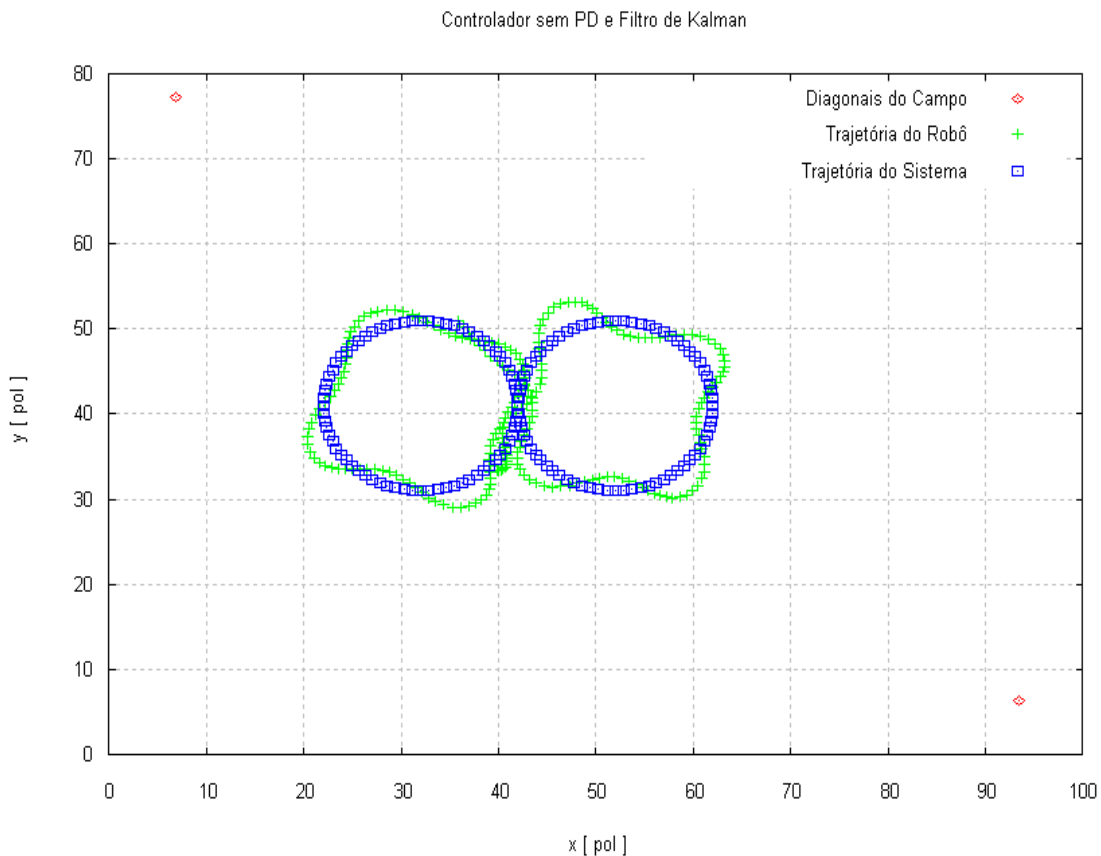
Enviar os comandos para o robô.

Fim.

IV. Resultados experimentais

Na figura 8, é gerado o gráfico de trajetória em “oito” e a que o robô realizou sem a ação do controle PD e Filtro de Kalman.

Figura 8 – Sem o controlador PD e FK.



Como pode ser visualizado na figura 8, o robô não consegue realizar a trajetória, sendo assim, criar um sistema de controle de autocorreção torna-se necessário.

IV.1. Determinado os coeficientes K_p e K_d .

As figuras 9 e 10 serão utilizadas para determinar o melhor fator de escala para o ganho K_p e K_d respectivamente, enquanto figura 11 será utilizada para determinar o coeficiente para K_{dd} .

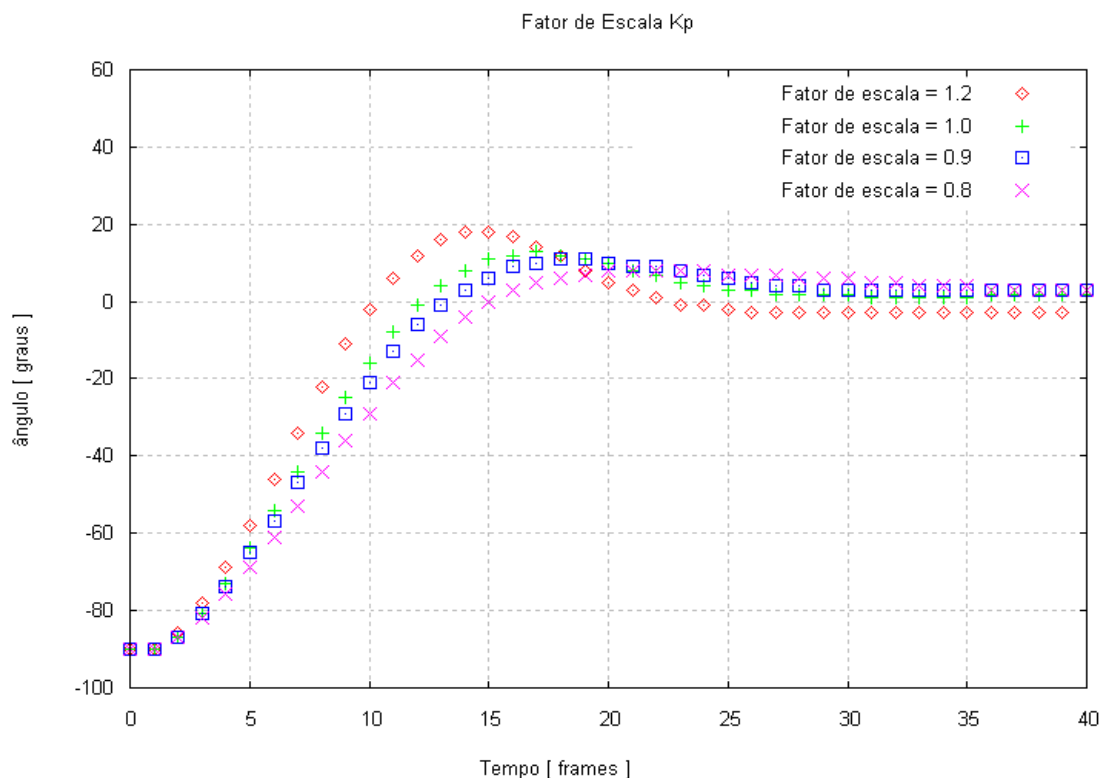


Figura 9 – Definição do Fator de Escala para K_p (ângulo de erro)

Através das curvas do gráfico da figura 9, verifica-se a curva que obtém o melhor desempenho, ou seja, a curva que possui o menor sobre sinal (overshoot), para determinar o melhor fator de escala que neste caso é 0.8.

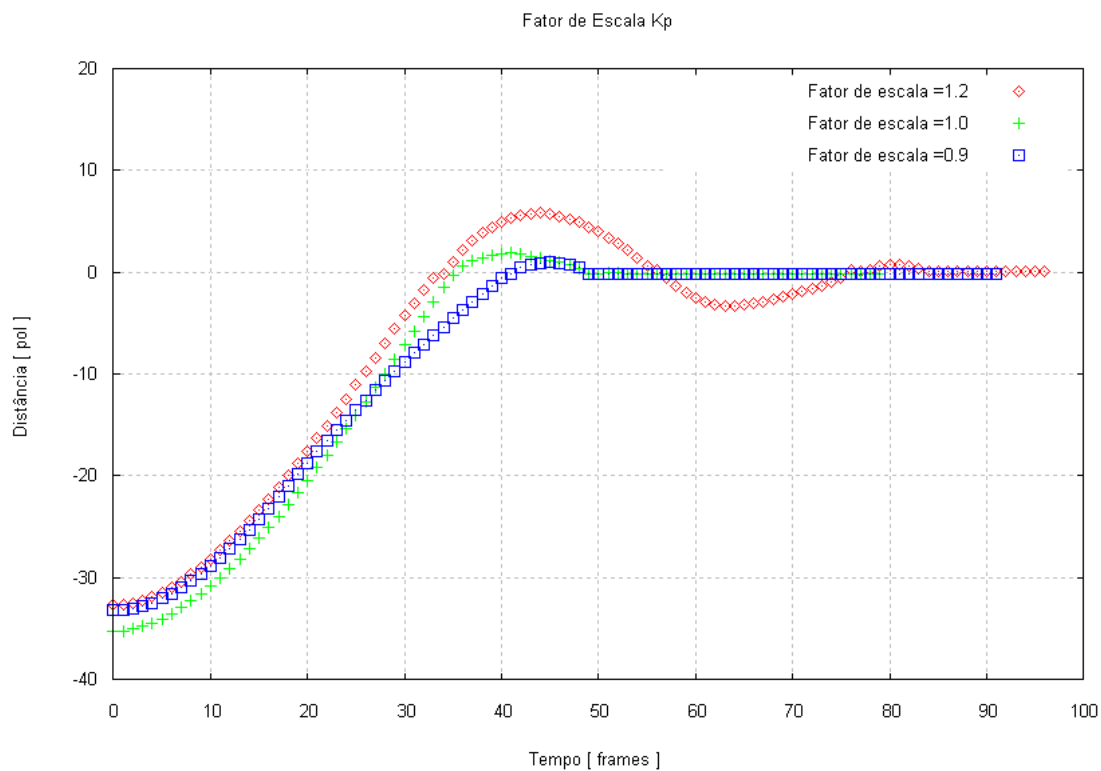


Figura 10 - Definição do Fator de Escala para Kdp (distância de erro)

Através das curvas do gráfico da figura 10, verifica-se a curva que obtém o melhor desempenho, ou seja, a curva que possui o menor sobre sinal (overshoot), para determinar o melhor fator de escala que neste caso é 0.9.

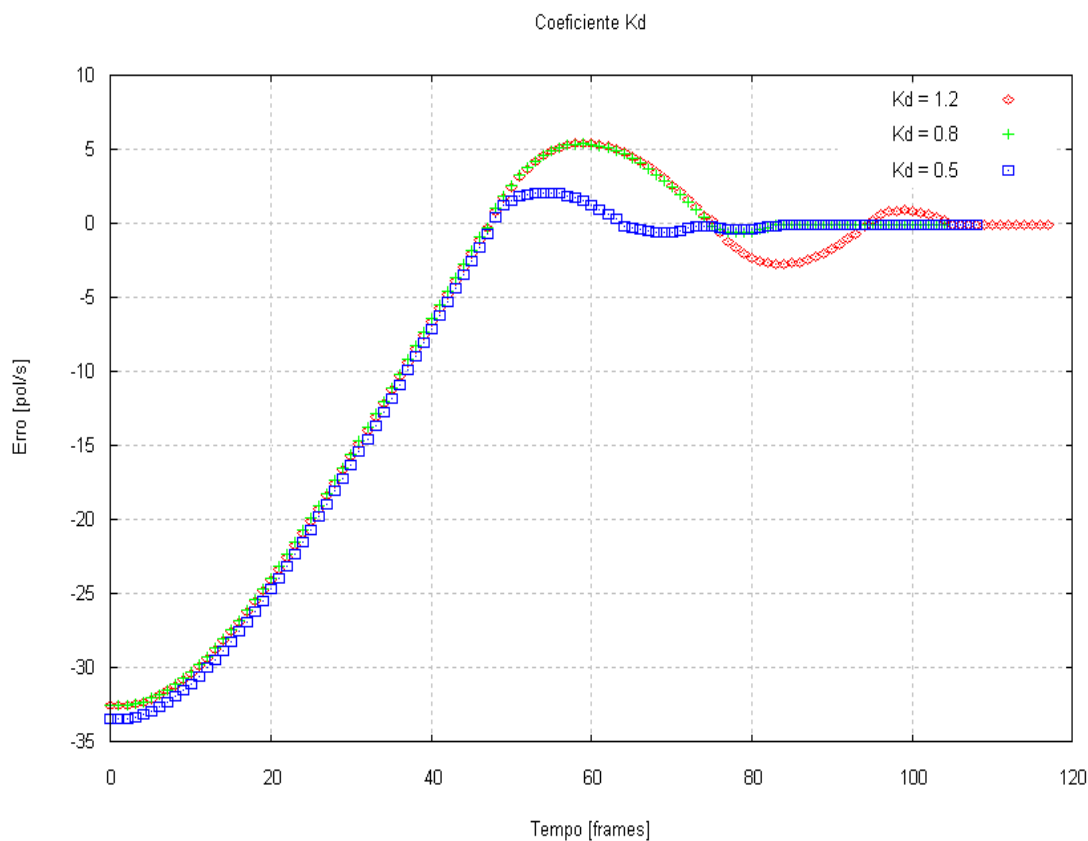


Figura 11 - Definição do coeficiente Kdd (velocidade de erro)

Através das curvas do gráfico da figura 11, verifica-se a curva que obtém o melhor desempenho, ou seja, a curva que possui o menor sobre sinal (overshoot), para determinar o melhor fator de escala que neste caso é 0.5.

Com a implementação do controlador PD, obtém-se o seguinte gráfico da trajetória realizado pelo robô em forma de “oito”, como mostra a figura 12.

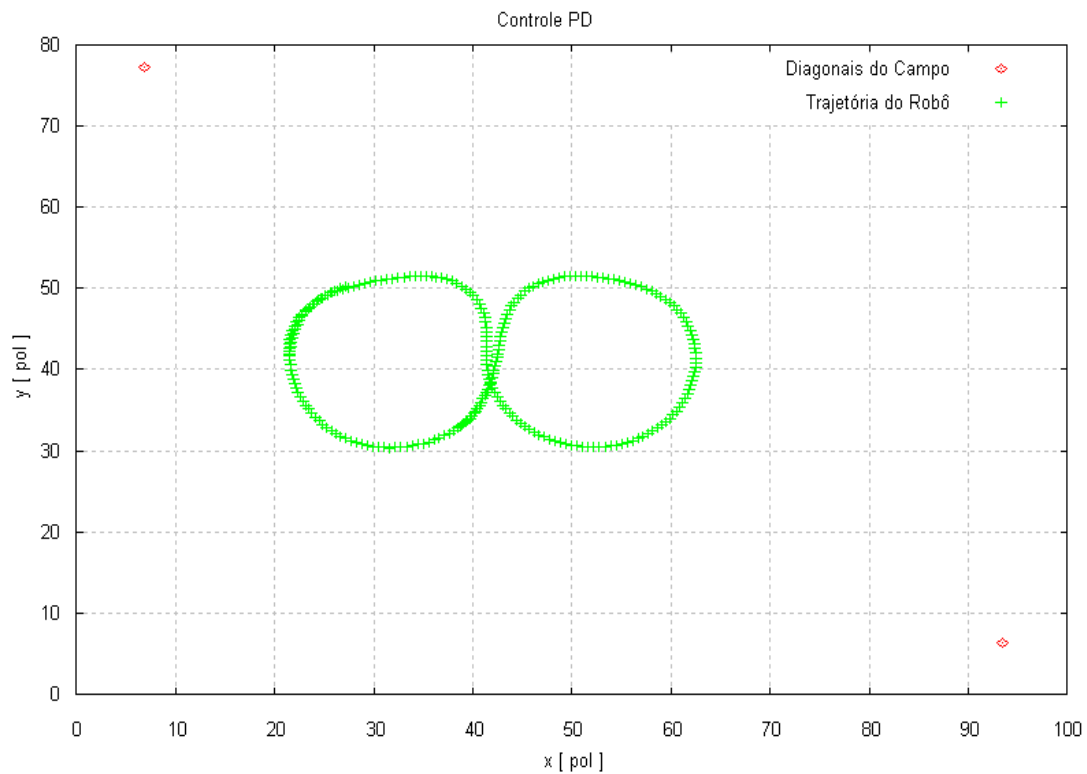


Figura 12 – Controlador PD

Com a implementação do controlador PD e comparando-no com a trajetória do sistema em forma de “oito”, obtém-se o seguinte gráfico como mostra a figura 13.

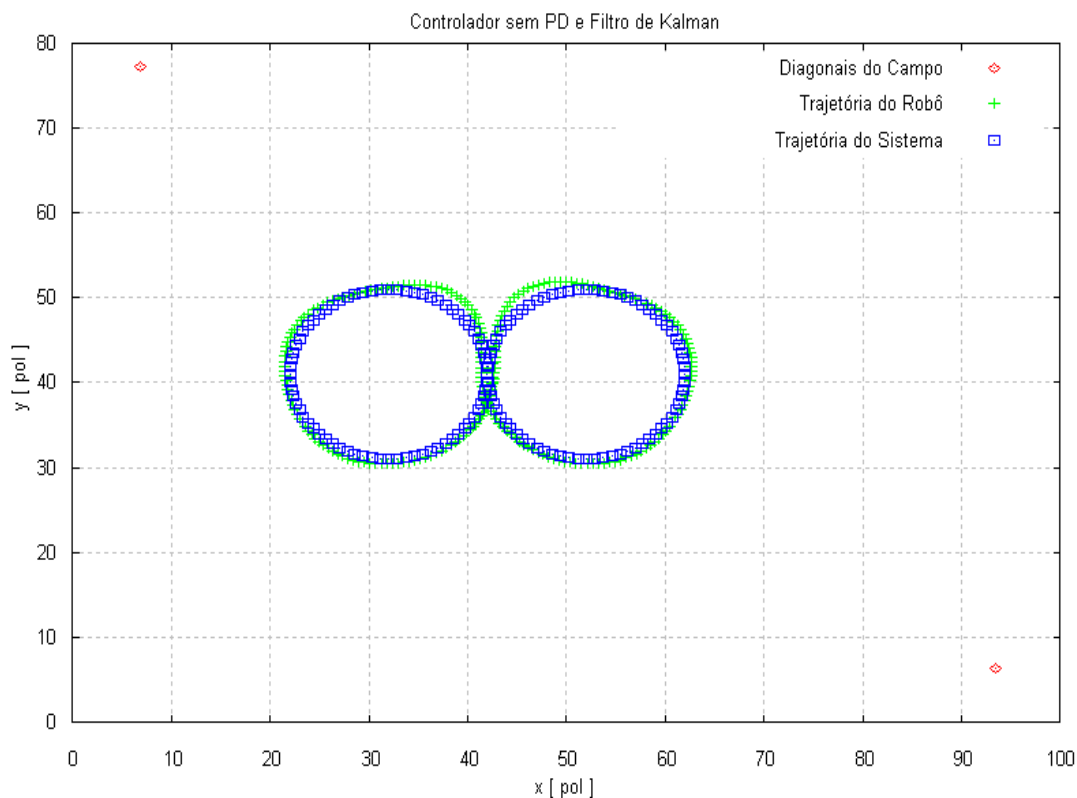


Figura 13 – Controlador PD comparado com a trajetória do sistema

IV.2. Determinado os coeficientes h e g.

Definindo o ganho de h e g através da figura 14.

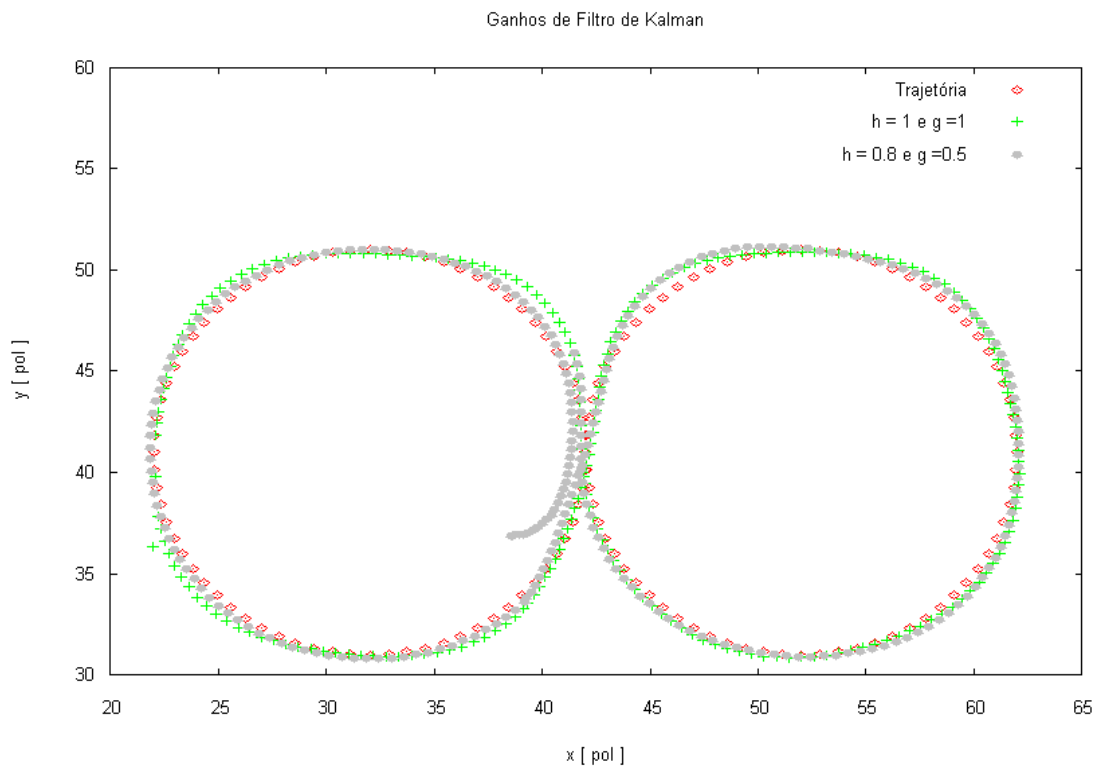


Figura 14 – Ganho do FK h e g.

Através do gráfico da figura 14, verifica-se o melhor coeficiente que obtém uma trajetória estimada para o robô, mais aproximada à do sistema, $h = 0.8$ e $g = 0.5$.

Com a implementação do controlador PD e FK, obtêm-se o seguinte gráfico da trajetória realizado pelo robô em forma de “oito”, como mostra a figura 12.

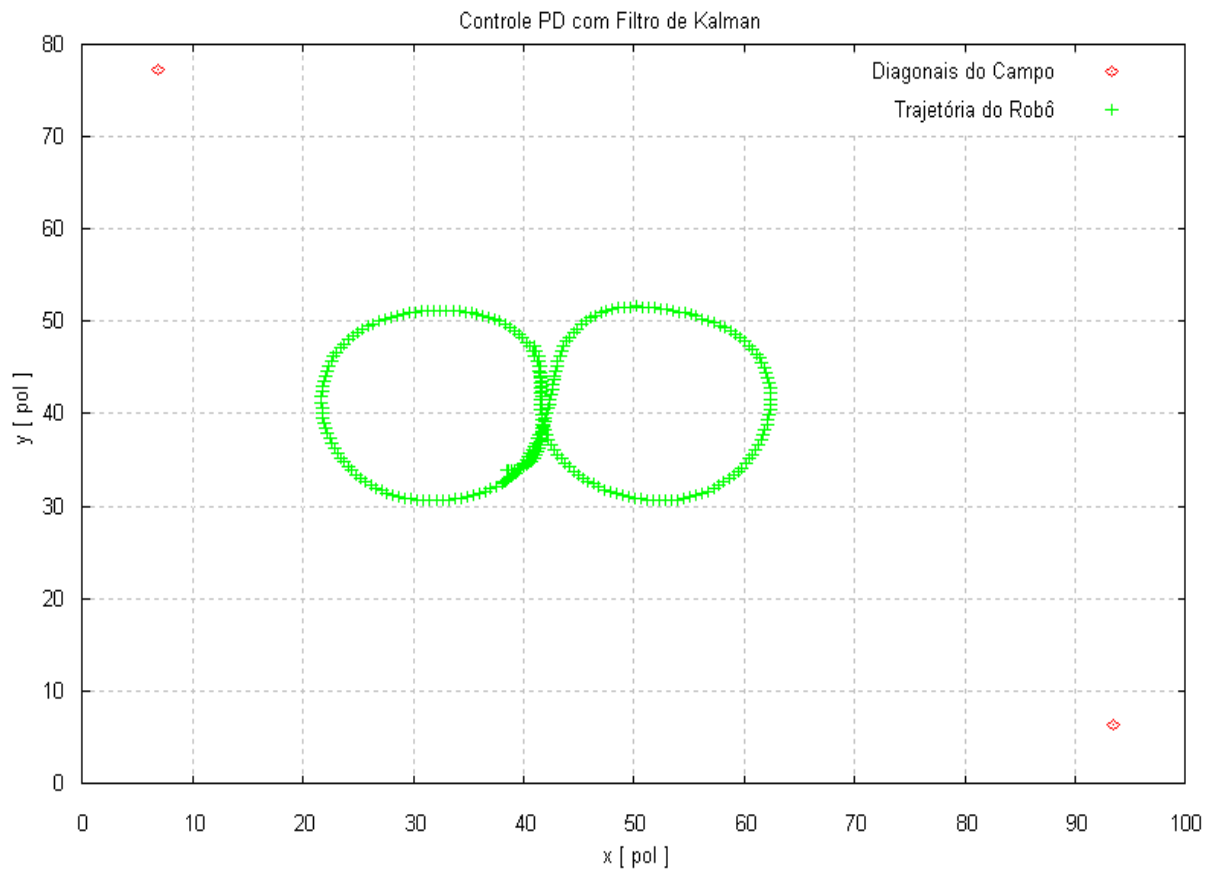


Figura 15 – Controlador PD com FK.

Com a implementação do controlador PD com FK, obtém-se o seguinte gráfico da trajetória realizada pelo robô em forma de “oito”, e comparando na com a do sistema como mostra a figura 16.

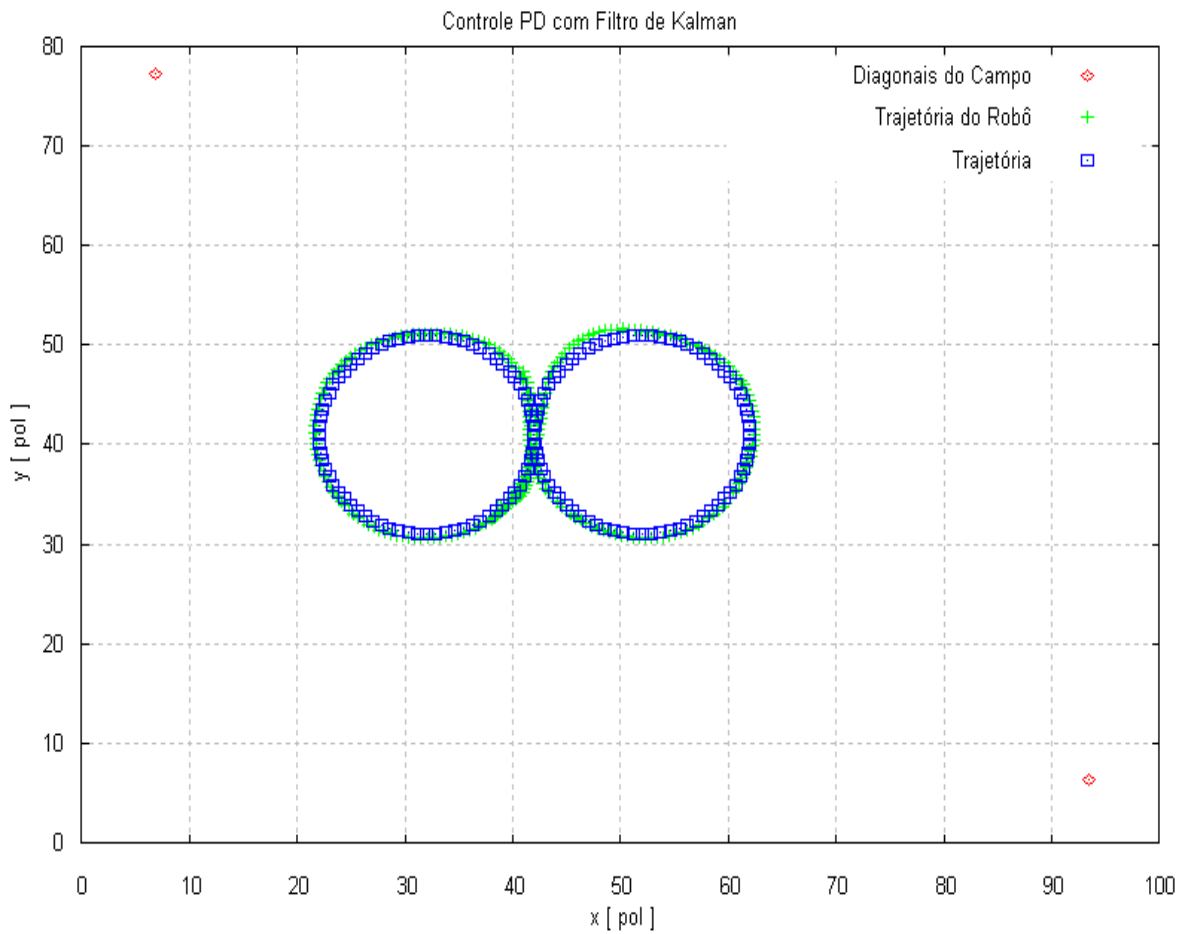


Figura 16 – Controlador PD com FK comparado com a trajetória do sistema.

V. Próximas Etapas

A implementação do sistema de controle no robô real e comparar os resultados obtidos. Criar uma interface gráfica para o sistema, de modo que se possa gerar uma trajetória qualquer através de uso do mouse, e que o robô possa realizar esse percurso e apresentar os gráficos de desempenho. Além dos ajustes de ganho sejam realizados automaticamente.

VI. Conclusões

Os testes foram realizados em simulação, eliminando possíveis problemas relacionados à visão computacional, transmissão de dados, o próprio robô e etc. Porém, o simulador Robot Soccer 1.5v leva em consideração efeitos físicos como atrito, inércia e outros, tornando-o o mais próximo das condições reais.

Com a implementação do controle PD, como demonstrado na figura 13, pode-se verificar a grande melhora obtida no controle do robô, apesar dos testes para definição de coeficientes serem de tentativa e erro, não foi difícil à regulação dos mesmos, como **GUPTA, MESSOM e DEMIKENKO [2004]** demonstraram em seu artigo. Porém com o uso do FK, pode melhorar ainda mais o controle, como exibido na figura 16. A definição dos parâmetros de ganho FK [**BITENCOURT, 2003**], [**GUPTA, MESSOM e DEMIKENKO, 2003**] tornou-se mais trabalhoso, pois a posição do robô é muito precisa em simulação, mas, no entanto, quando o robô atinge velocidades médias altas o mesmo mostra-se mais eficaz, estimando a posição e a velocidade do robô com grande precisão.

A implementação do modelo cinemático, também provou ser um sistema simples e eficiente, obtendo bons desempenhos em conjunto com controlador PD [**GUPTA, MESSOM e DEMIKENKO, 2004**] e Filtro de Kalman [**BITENCOURT, 2003**], [**GUPTA, MESSOM e DEMIKENKO, 2003**].



Centro Universitário da FEI Projeto de pesquisa



V. Referências Bibliográficas

Bitencourt Jr., Hudson. Métodos de Estimação Recursiva Baseada no Filtro de Kalman Aplicados a Sistemas Não Lineares, 2003. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Minas Gerais, Belo Horizonte, 2003. Disponível em : < <http://www.cpdee.ufmg.br/~hudson/pesquisa.htm>>. Acesso em 21 mar.

De Luca, A, Oriolo, G., Vendittelli, M., "Control of Wheeled Mobile Robots: An Experimental Overview, Eds., Springer-Verlag, 2001.

Federation of International Robot-soccer Association (FIRA), 1998

Disponível em: < <http://www.fira.net>>. Acesso em: 16 mar 2006.

Ogata. K, Engenharia de Controle Moderno, 3º Eds., LTC, 1997.

PID Tutorial, 1997.

Disponível: <<http://www.engin.umich.edu/group/ctm/PID/PID.html>> . Acesso em: 16 de mar 2006.

Sen Gupta, G., Messom, C, Demidenko, S., "Vision Assisted Measurement for Optimization of Robot Motion and Position Control Functions", IEEE Instrumentation and Measurement Technology Conference, Como, Italy, May 2004, pp 297- 302, ISBN 0-7803-8248-1.

Sen Gupta, G., Messom, C, & Demidenko, S., "Global Vision Based Optimization of Control Functions and Kalman Filtering for Robotic Applications", Proceedings of IEEE International Conference on Computational Intelligence, Robotics and Autonomous Systems, (CIRAS 2003), ISSN 0219-6131.

Vieira, F. C. et al. **Position and Orientation Control of a Two-Wheeled Differentially Driven Nonholonomic Mobile Robot**. II CINCO - International Conference on Informatics in Control, Automation and Robotics. Setúbal, Portugal, 08/2004.