



PLANEJAMENTO DE VELOCIDADE DE UM ROBÔ MÓVEL UTILIZANDO O MÉTODO DE CAMPO POTENCIAL

Carlos Erlan Olival Lima¹, Aurélio Agostinho Adão², Francisco Marcelino Almeida³, Mário Bibiano da Silva Júnior¹

¹Aluno de Graduação em Engenharia Mecânica – IFPI. e-mail: erlanolival@gmail.com

²Mestre do Programa de Pós-Graduação em Engenharia Elétrica - IFPI. e-mail: aaadão@uol.com.br

³Mestrando do Programa de Pós-Graduação em Biotecnologia – IFPI. e-mail: franciscomarcelinoalmeida@gmail.com

¹Aluno de Graduação em Engenharia Mecânica – IFPI. e-mail: mario_bibiano@hotmail.com

Resumo: Este trabalho tem como objetivo planejar a velocidade e o caminho percorrido pelo robô de forma a alcançar o seu alvo no menor tempo possível. O planejamento da velocidade é determinado a partir das velocidades relativas entre o robô e o alvo assim como a partir das posições relativas entre ambos. A velocidade angular do robô e do alvo também é considerada nos cálculos. Este trabalho será utilizado para determinar a velocidade e o posicionamento de um robô da categoria sumô, o qual tem por finalidade empurrar o seu oponente para fora da arena onde se encontra, sendo que esta é uma superfície plana negra com as bordas brancas, sendo que o robô deve detectar estas bordas com a finalidade de evitar a sua saída da arena, portanto, durante a determinação da trajetória e da velocidade, será considerado que as bordas serão o obstáculo e que estarão em estado estacionário e o robô oponente será o alvo em movimento. As simulações a serem executadas serão utilizadas para comprovar a efetividade da proposta a ser analisada. Os softwares utilizados nas simulações foram o MATLAB e o ROBOTBASIC. A partir dessas simulações obteve-se que a velocidade que o robô deverá ter será de 32 cm/s para que possa encontrar o oponente em 1,7 segundos.

Palavras-chave: Planejamento da velocidade, planejamento do percurso, robô móvel

1. INTRODUÇÃO

Para determinar o planejamento da velocidade e da posição é utilizado o método de campo potencial. O método convencional de campo potencial não é adequado para determinar a trajetória de um robô em um ambiente dinâmico, ou seja, um ambiente onde o alvo e o obstáculo estão em movimento (L. Huang, 2009).

O método de campo potencial vem sendo bastante utilizado na robótica para a geração de caminhos, neste método é considerado que a meta a ser alcançada corresponde a uma força de atração e os obstáculos correspondem a uma força de repulsão, já com relação à direção a ser seguida pelo robô para desviar dos obstáculos e encontrar o seu alvo é considerada como sendo um vetor resultante da soma de todos os vetores de força envolvidos (GEDSON FARIA).

Atualmente há poucas pesquisas que procuram apresentar o traçado que o alvo a ser perseguido percorre. (BORENSTEIN; KOREN, 1989) determinou uma nova função potencial que foi definida como a soma das funções quadráticas das posições e velocidades relativas do robô com relação ao alvo e o obstáculo.

Neste artigo será estendida a aproximação para o método de campo potencial, de forma que esse método possa ser utilizado para planejar a velocidade e o percurso do robô móvel com o alvo e o obstáculo em movimento, de forma que este método possa ser aplicado a outros tipos de utilização do robô móvel, que podem ser mais avançados que o necessário para o movimento do robô a ser analisado neste trabalho.

Para simular a trajetória e a velocidade do robô foram utilizados os softwares MATLAB e ROBOTBASIC.

O objetivo deste trabalho é planejar a velocidade e o percurso de um robô autônomo da classe sumô para participação em competições de guerra de robôs. Para realizar o planejamento do percurso e da velocidade para esta classe foi considerado que a borda da arena irá equivaler ao obstáculo e estará em movimento estacionário, e o oponente será o alvo em movimento.

Apesar de este artigo utilizar este método para robôs da classe sumô, a sua utilização pode ser estendida a outros tipos de robôs, como os seguidores de linha e da classe de futebol, além de poder



ser implementado na indústria, quando se há ambientes desestruturados ou semi-estruturados, dos quais temos informações limitadas quanto à disposição estática ou dinâmica dos objetos.

Existem diversas técnicas para estabelecer a trajetória que um robô deve descrever, sendo que dentre as principais temos o VFH (Virtual Field Histogram) e o DWA (Dynamic Window Approach). A vantagem do método apresentado nesse artigo é que o mesmo pode ser estendido para ambientes onde tanto os obstáculos quanto o alvo estão em movimento, embora o conteúdo deste artigo irá ater-se apenas a ambientes onde o alvo está em movimento, devido ao objetivo proposto no artigo.

2. REVISÃO DA LITERATURA

2.1 Modelamento do Sistema

Na figura 1 é apresentada a trajetória do robô, assim como os vetores relativos à velocidade e as posições relativas entre o robô e o alvo e entre o robô e o obstáculo. Para simplificar o modelamento será considerado nos cálculos apenas um obstáculo e um alvo a ser atingido. Para descrever o sistema são utilizadas as seguintes notações:

XOY: coordenadas globais no plano de trabalho

$p_{tar} \in \mathbb{R}^2$: posição do alvo

$v_{tar} \in \mathbb{R}^2$: velocidade do alvo

$p \in \mathbb{R}^2$: posição do robô

$v \in \mathbb{R}^2$: velocidade do robô

$p_{obs} \in \mathbb{R}^2$: posição do obstáculo

$v_{obs} \in \mathbb{R}^2$: velocidade do obstáculo

$r_{obs} \in \mathbb{R}$: raio do obstáculo

$p_{rt} = p_{tar} - p$: Posição relativa do robô ao alvo

$p_{ro} = p_{obs} - p$: posição relativa do robô ao obstáculo

$p_{ot} = p_{tar} - p_{obs}$: posição relativa do obstáculo ao alvo

Ψ : ângulo of p_{rt}

θ_{ro} : ângulo of p_{ro}

θ_{ot} : ângulo of p_{ot}

θ_{tar} : ângulo of v_{tar}

θ_{obs} : ângulo of v_{obs}

θ : ângulo of v

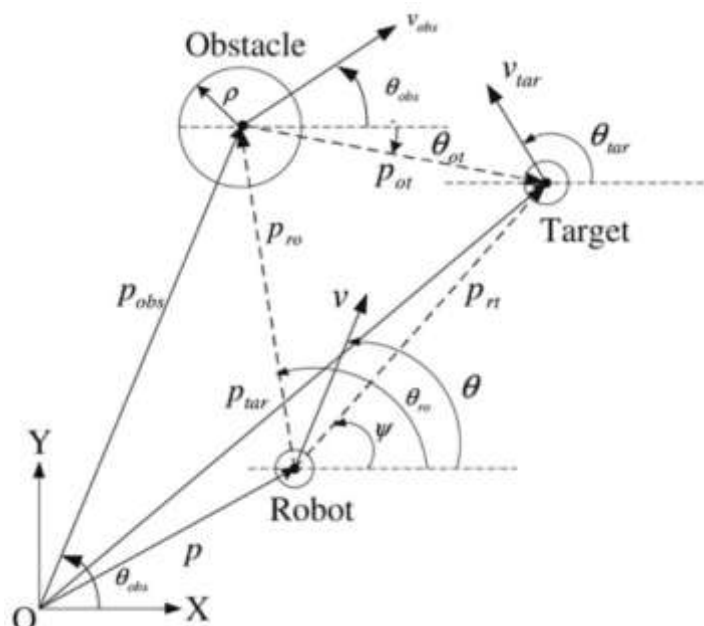


Figura 1- Trajetória Percorrida pelo Robô Móvel, L. HUANG

Proposta 1: As posições \mathbf{p} , \mathbf{p}_{tar} e \mathbf{p}_{obs} e velocidades \mathbf{v} , \mathbf{v}_{tar} e \mathbf{v}_{obs} são todas conhecidas.

Proposta 2: O robô, o alvo e os obstáculos são tratados como pontos de massa.

Proposta 3: A máxima velocidade linear e angular do robô são $\mathbf{v}_{máx}$ e $\mathbf{w}_{máx}$

Proposta 4: A velocidade do alvo é limitada por $\|\mathbf{v}_{tar}\| \leq \mathbf{v}_{máx}$

Fazendo $\mathbf{p}_{rt} = [x_{rt} \ y_{rt}]^T$ e $\mathbf{p}_{ro} = [x_{ro} \ y_{ro}]^T$, o movimento relativo entre o robô e o alvo é descrito por:

$$\mathbf{p}_{rt} = x_{rt} \ y_{rt}^T \quad (1)$$

Onde:

$$x_{rt} = v_{tar} \cos \theta_{tar} - |v| \cos \theta \quad (2)$$

$$y_{rt} = v_{tar} \sin \theta_{tar} - |v| \sin \theta \quad (3)$$

E o movimento relativo entre o robô e o obstáculo é descrita por:

$$\mathbf{p}_{ro} = x_{ro} \ y_{ro}^T \quad (4)$$

Onde:

$$x_{ro} = v_{obs} \cos \theta_{obs} - |v| \cos \theta \quad (5)$$

$$y_{ro} = v_{obs} \sin \theta_{obs} - |v| \sin \theta \quad (6)$$

O método de campo potencial usa como teoria o potencial de atração e repulsão entre os corpos, sendo que entre o robô e o alvo temos um potencial de atração e entre o robô e o obstáculo temos um potencial de repulsão. O potencial de atração (U_{att}) e o potencial de repulsão (U_{rep}) são definidos como:

$$U_{att} \triangleq 1/2 \varepsilon_1 p_{rt}^T p_{rt} \quad (7)$$

$$U_{rep} \triangleq 1/2 \varepsilon_2 p^{-1} - p_0^{-1} \quad (8)$$

$$U = U_{att} + U_{rep} \quad (9)$$

Onde p denota a mínima distância entre o robô e o obstáculo, $p_0 > 0$ é a distância de influência do obstáculo e ε_1 e ε_2 devem ser maiores que zero e correspondem aos fatores de escala para potencial atrativo e de repulsão.

2.2 Planejamento da Velocidade do Robô quando $p > p_0$

Como o planejamento de velocidade está sendo feito para um robô da categoria sumo, foi considerado que a influência do obstáculo é baixa, portanto o potencial de atração é utilizado para planejar a velocidade.

De acordo com L.Huang (2009) teremos que a velocidade linear vetorial do robô nesta situação será dada por:

$$\mathbf{v} = v_{tar} + \varepsilon_1 p_{rt} \quad (10)$$

Isso equivale às seguintes relações trigonométricas:

$$v \sin \theta - \psi = v_{tar} \sin(\theta_{tar} - \psi) \quad (11)$$

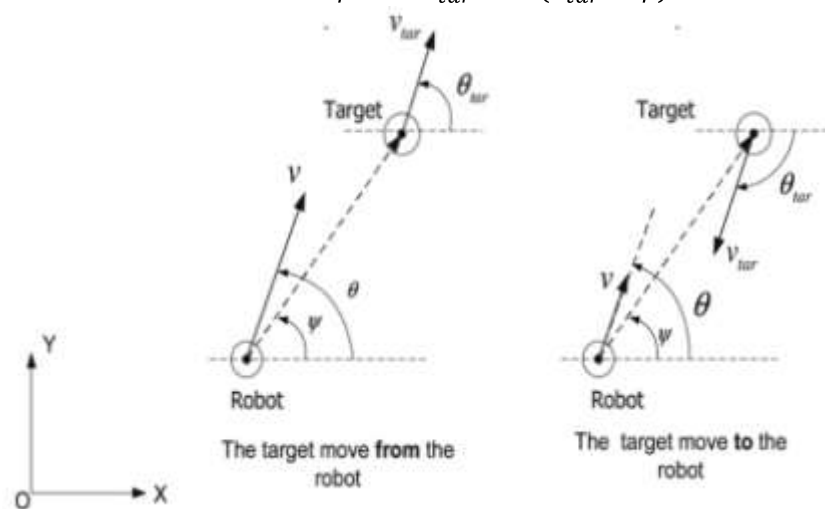


Figura 2 – Representação do Posicionamento do Robô em Relação ao Alvo quando $p > p_0$, (L.HUANG, 2009)



Na figura 2 são apresentados os ângulos formados entre o robô e o eixo horizontal e entre o eixo horizontal e a trajetória projetada pelo alvo, além do ângulo entre o alvo e o eixo horizontal e os vetores velocidade de cada elemento participante.

Considerando que os vetores v , v_{tar} e p_{rt} , apresentados na figura 2 formam um triângulo, teremos a partir da lei dos cossenos, que:

$$v = v_{tar}^2 + 2\varepsilon_1 p_{rt} v_{tar} \cos(\theta_{tar} - \psi + \varepsilon_1^2 |p_{rt}|^2)^{1/2} \quad (12)$$

Nesta equação, se deve assumir que v_{tar} e $\|p_{rt}\|$ são diferentes de zero.

A direção do robô pode ser determinada a partir das equações 11 e 12, sendo que o ângulo que o robô faz com o eixo x é determinado por:

$$\theta = \psi + \sin^{-1}((v_{tar} \sin \theta_{tar} - \psi) / (|v|)) \quad (13)$$

Com a velocidade do alvo sendo incluída no planejamento de velocidade, se pode determinar mais precisamente a direção que o robô deve seguir.

2.3 Máxima Velocidade Angular do Robô

Para tornar a velocidade angular aceitável, a direção do robô, que é indicada por θ deve ser planejada para ser a menor possível. A análise da direção para esta situação é feita a partir da expansão da equação 13.

$$\theta = \psi + \sin^{-1}((\sin \theta_{tar} - \psi) / (1 + 2\varepsilon_1 \kappa \cos \theta_{tar} - \psi + \varepsilon_1^2 \kappa^2)^{1/2}) \quad (14)$$

Sendo que,

$$\kappa \triangleq |p_{rt}| / |v_{tar}|$$

De acordo com [1], após várias simplificações temos que a velocidade angular é representada por:

$$w = (\theta_{tar} + \varepsilon_1^2 \kappa^2 \psi + \kappa \cos \psi - \theta_{tar} \theta_{tar} + \psi + \sin \psi - \theta_{tar} \kappa) / (\varepsilon_1 \kappa + 2\kappa \cos \psi - \theta_{tar} + \frac{1}{\varepsilon_1}) \quad (15)$$

3. SIMULAÇÃO

3.1 Simulação com RobotBasic

Antes de repassar os cálculos para o microcontrolador, foi feita uma simulação no software RobotBasic, software que pode ser utilizado para simular os movimentos de um robô utilizando a linguagem BASIC. Neste software, a localização do robô é representada pelas coordenadas R_x , R_y , como pode ser observado na Figura 3. Nesta figura temos que o alvo está localizado na posição T_x , T_y e as diferenças entre as coordenadas do robô e do alvo são representadas por dX e dY .

Para calcular a distância do robô em relação ao alvo é utilizada a função **POLARR (dx, dy)** e para determinar a direção que o robô deve girar é utilizada a função **POLARA (dx, dy)**, a qual retorna o ângulo em relação ao eixo horizontal, que é formado pela linha entre o centro do robô e o centro do alvo, como mostrado na figura 3.

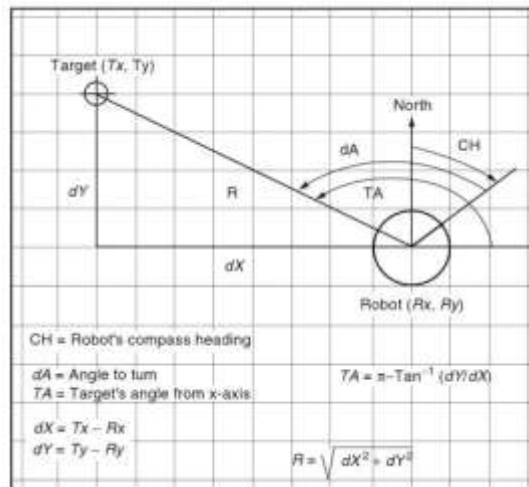


Figura 3 – Representação das Coordenadas do Robô no Software ROBOTBASIC, (BLANKENSHIP, 2008)

O valor retornado pela função **POLARA ()** é dado em radianos, então caso se deseje converter o valor de radianos para graus, deve-se utilizar o seguinte comando:

$$\text{Angle_In_Degrees} = \text{Angle_In_Radians} * \pi () / 180$$

Então, para se determinar o ângulo desejado, coloca-se a função **POLARA (dx, dy)** onde se deve colocar o valor do ângulo em radianos. No software tem-se que a posição leste representa 0°, a posição norte representa 90°, a posição oeste representa 180° e a posição sul representa 270°.

Portanto é necessário acrescentar 90° aos ângulos obtidos, para que o programa realize os cálculos da direção do robô em relação à posição norte, que seria representada pelo eixo y em coordenadas cartesianas.

Sendo assim, o cálculo de **dA**, que representa o ângulo de giro do robô será calculado pelo seguinte comando:

$$dA = \text{POLARA}(dx, dy) * 180 / \pi () + 90 - CH$$

Após esses pequenos conceitos sobre os comandos a serem utilizados para simular o planejamento de posição no ROBOTBASIC é apresentada na figura 4 a linha de código utilizada para fazer a simulação e na figura 5 temos a imagem do robô gerada pela simulação situada em um ambiente que representa as condições reais de operação, com as mesmas proporções em relação ao ambiente real.

```
findbeacon:
  cnt = 0
  repeat
    cnt = cnt + 1
  goseek facebeacon
  if cnt < 10
    gosub forwardtillblocked
    if (cnt-1) and (rfeel() < 0) then cnt = 10
    turndir = 3
  gosub goaround
  else
    gosub unstick
    cnt = 0
  endif
until beaconfound
return
unstick:
if random(100) < 50 then return 100
for i = 0 to 10 + random(200)
  while not (xtasper() < 404)
    rforward 1
  wend
  status random (0) -3
next
return
facebeacon:
while not (beacon (red))
  status 1
wend
return
forwardtillblocked:
while not (rfeel() < 404) and not (xtasper() < 404)
```

Figura 4 – Parte do Código Gerado pelo Software ROBOTBASIC Simulando o Método de Campo Potencial

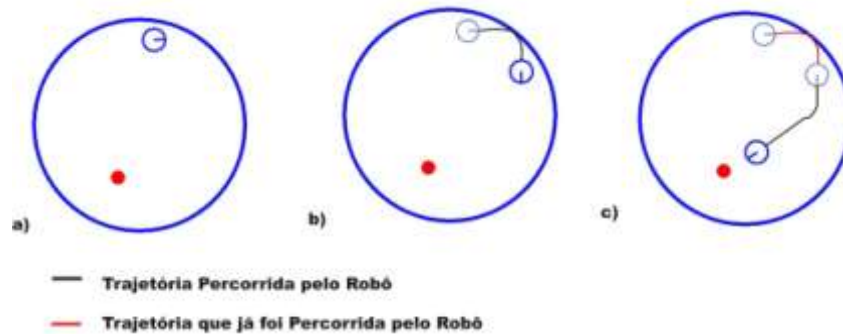


Figura 5 – Simulação Gerada pelo Software ROBOTBASIC Representando o Posicionamento e o Deslocamento do Robô na Arena: a) Posição Inicial do Robô antes de Iniciar o Movimento; b) Robô Tentando Localizar o Alvo; c) Deslocamento do Robô ao Encontrar o Alvo

A partir dessa simulação se pôde definir como seria programado o robô, de forma a cumprir o objetivo de encontrar o oponente o mais rápido possível e jogá-lo para fora da arena.

3.2 Simulações do Método de Campo Potencial a partir do Software MATLAB

Para estabelecer a velocidade que o robô deverá ter de forma que possa gerar um torque elevado e possa alcançar a sua meta no menor tempo possível, as equações apresentadas no presente artigo foram simuladas no software MATLAB.

A partir destas simulações gerou-se os gráficos apresentados na figura 6 e 7, sendo que o primeiro relacionada o tempo com as possíveis velocidades a serem estabelecidas para o robô, e o segundo mostra o percurso que o robô irá descrever até encontrar o alvo.

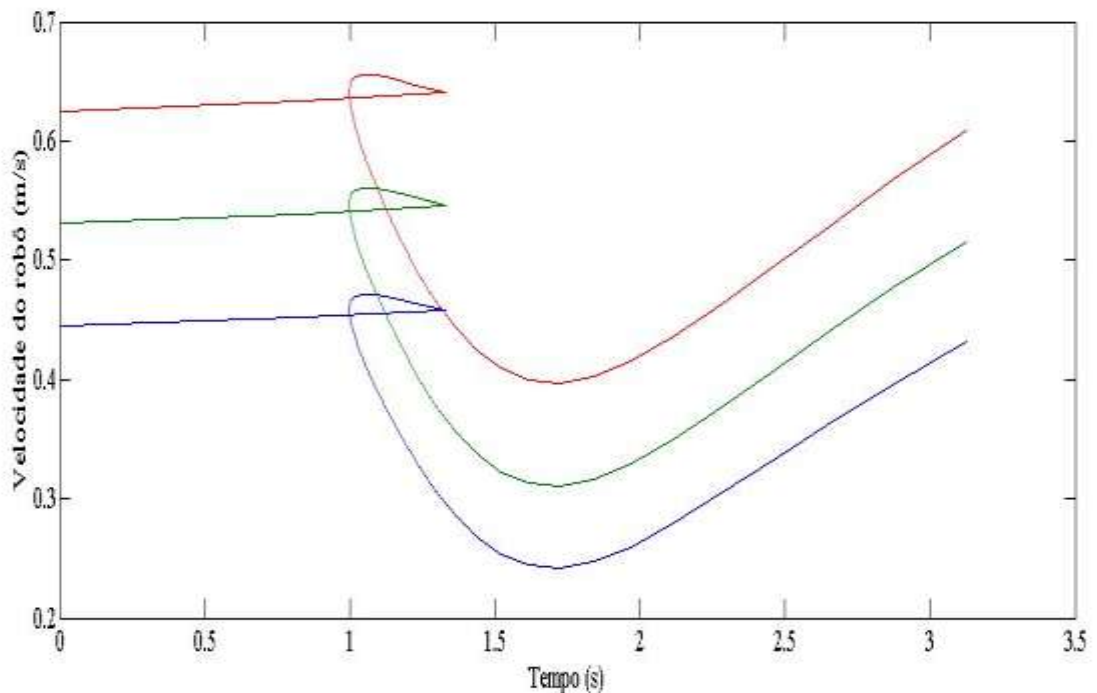


Figura 6 – Velocidade Projetada para o robô em relação ao Tempo: (a)Linha Azul – Simulação obtida quando a constante ϵ_1 for igual a 0.3; (b)Linha Verde – Simulação obtida quando a constante ϵ_1 for igual a 0.4; (c) Linha Vermelha - Simulação obtida quando a constante ϵ_1 for igual a 0.5;

Através da figura 6 percebe-se que quanto maior for o valor estabelecido para a constante ε_1 maior será o valor correspondente a velocidade do robô. Nesta simulação foi considerado que o alvo tem uma variação de velocidade entre [0: 0.3] e a velocidade estimada para o robô no início da simulação variou no intervalo [0: 0.5], ambas as velocidades foram estabelecidas em m/s.

De acordo com as equações apresentadas neste artigo, temos que a velocidade do robô deve ser sempre maior que a velocidade do alvo, mas não deve ser muito elevada, pois como o objetivo do robô é empurrar o oponente para fora da arena, então é mais importante que o mesmo tenha um elevado torque, e como o torque é inversamente proporcional a velocidade angular e esta é diretamente proporcional a velocidade linear, então quanto maior for o torque, menor deverá ser a velocidade linear.

Portanto, na figura 6 foi escolhido o gráfico cuja constante ε_1 foi estabelecida com valor de 0.4, pois nesse caso tem-se que a mínima velocidade necessária para que o robô encontre o alvo será de aproximadamente 0,32 m/s, o que garante uma velocidade acima da velocidade estabelecida como a máxima para o oponente, além de garantir um torque bem elevado, além de que o tempo para que o robô alcance o seu oponente será no máximo 1,7 segundos.

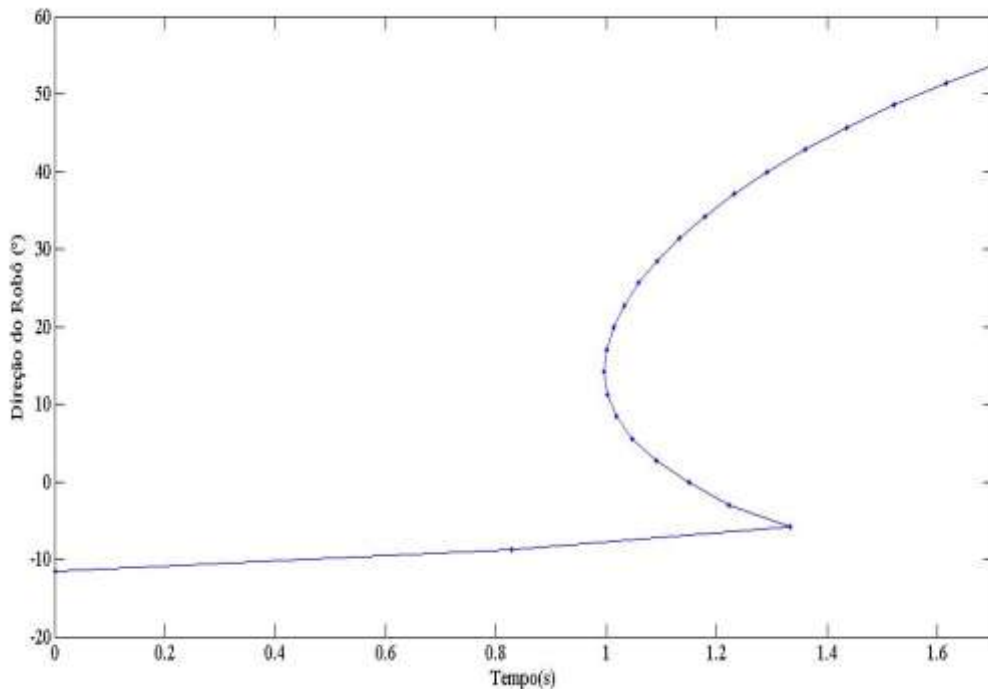


Figura 7 – Trajetória Descrita pelo robô até Encontrar o Alvo

Na figura 7 temos a descrição da trajetória seguida pelo robô até encontrar o seu oponente, como os respectivos ângulos formados pelo robô em relação ao eixo x.

4. APLICAÇÃO

Como já foi comentado anteriormente, este método será aplicado em um robô da categoria sumo, de forma que se possa aprimorar a percepção do mesmo em relação ao ambiente e o oponente. A figura 8 apresenta o protótipo onde será incrementado este método.



Figura 8 – Protótipo de um Robô Móvel para Participação na Guerra de Robôs na Categoria Sumô Autônomo

5. CONCLUSÃO

Através das simulações realizadas nota-se que o método é bastante efetivo, pois mesmo com uma velocidade relativa pequena, o robô consegue encontrar o seu alvo em apenas 1,7 segundos, sendo que se considera que o alvo está em movimento e tentando fugir de seu perseguidor, que no caso, é o robô.

Espera-se utilizar este método não apenas em robôs do tipo sumo, mas também em robôs seguidores de linha.

REFERÊNCIAS

BLANKENSHIP, John; MISHAL, Samuel. **ROBOT PROGRAMMER'S BONANZA**. Ed. McGraw Hill, 2008.

Gedson Faria, Priscila da Silva Martins, Mauro Conti Pereira. **Time de Robôs controlado por Campos Potenciais**. Anais do XXVI Congresso da SBC, pág. 443-448.

J. Borenstein, Y. Koren, **Real-time obstacle avoidance for fast mobile robots**, IEEE Transactions on Systems, Man, and Cybernetics 20 (4) (1989) 1179–1187.

L. Huang, **Velocity planning for a mobile robot to track a moving target — a potential field approach**, Robotics and Autonomous Systems 57 (2009) 55–63.