

UM SIMULADOR DINÂMICO PARA MINI-ROBÔS MÓVEIS COM MODELAGEM DE COLISÕES

Marcelo M. Yamamoto*

yamamoto@dca.ufrn.br

Adelardo A. D. de Medeiros*

adelardo@dca.ufrn.br

Diogo P. F. Pedrosa*

diogo@dca.ufrn.br

Pablo J. Alsina*

pablo@dca.ufrn.br

*Dept^o de Eng^a de Computação e a Automação, Universidade Federal do Rio Grande do Norte
59072-970 Natal, RN, Brasil

ABSTRACT

This paper presents a real time simulator for a wheeled non-holonomic mobile robot with differential drive. It is presented the simulator behaviour, as well as robot's kinematics and dynamics and the detection and treating of collisions between robots and objects.

KEYWORDS: Simulator, mobile robots, robot soccer, dynamic model, collisions.

RESUMO

Este artigo apresenta um simulador em tempo-real para robôs móveis não-holonômicos com acionamento diferencial. São apresentados o funcionamento do simulador, bem como a modelagem cinemática e dinâmica do robô e a detecção e o tratamento das colisões entre os robôs e objetos.

PALAVRAS-CHAVE: Simulador, robôs móveis, futebol de robôs, modelo dinâmico, colisões.

1 INTRODUÇÃO

Muitos simuladores para robôs já foram desenvolvidos, tanto para robôs manipuladores (Cook et al., 1994), quanto para robôs móveis (Kimoto e Yuta, 1992; Kulkarni et al., 1991; Kulkarni e Pettus, 1991). Três pontos importantes devem ser observados em simuladores: modelagem do robô, modelagem das colisões e tempo de simulação. Muitos simuladores não incluem a modelagem dinâmica do robô, se limitando à modelagem cinemática. Quanto às colisões, em alguns casos (Kulkarni et al., 1991; Kulkarni e Pettus, 1991) elas são detectadas porém não tratadas; em outros casos (Kimoto e Yuta, 1992), os robôs têm sua forma simplificada para círculos. Outro problema freqüente é que o tempo de simulação não corresponde ao tempo real, o que dificulta o teste e o acoplamento com os controladores reais.

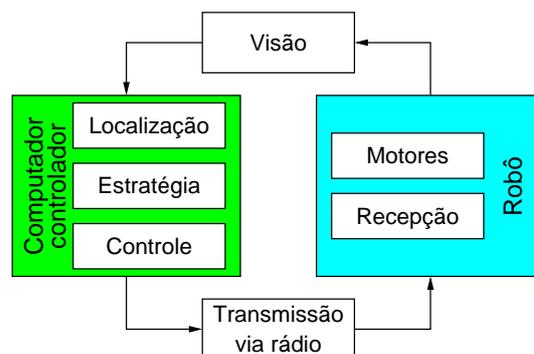


Figura 1: Futebol de robôs com frota real

A motivação principal para o desenvolvimento de um simulador que contorna estas limitações foi sua utilização em um ambiente de futebol de robôs. No sistema a ser simulado, cada time possui três robôs cúbicos com arestas de 7,5cm. Uma bola de golfe laranja é utilizada. Cada time possui uma câmera localizada sobre um campo com dimensões 130×150cm, ligada a um computador. A câmera capta a imagem do campo, o sistema de visão localiza os robôs, a estratégia gera os pontos de referência para cada um dos robôs, o controle transforma essa informação em tensões para os motores e essas tensões são transmitidas para os robôs via rádio, como mostra a Figura 1.

O simulador foi implementado para funcionar em um computador distinto daquele onde funciona o sistema de controle, como mostra a figura 2. A transmissão dos sinais de controle via rádio é substituída pela transmissão via uma rede Ethernet. Como o simulador funciona em tempo real, os sinais de controle são levados em conta a partir do instante em que são efetivamente recebidos pelo simulador.

No sistema real, uma imagem é captada pela câmera a cada 30ms. Para manter um comportamento equivalente,

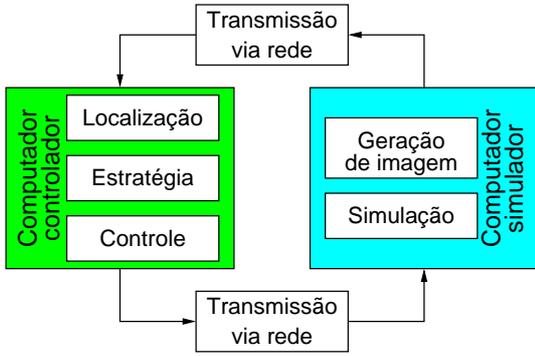


Figura 2: Futebol de robôs com simulador

a cada 30ms de simulação é gerada uma imagem sintética do campo, com os robôs e a bola nas respectivas posições. Esta imagem é então enviada (de forma comprimida) para o sistema de controle, que realiza então o seu processamento como se a imagem fosse originária da câmera. Desta forma, a substituição da frota de robôs pelo simulador não é percebida pelo sistema de controle.

2 MODELAGEM MATEMÁTICA DO ROBÔ

Esta seção descreve o modelo matemático de um robô cúbico com aresta de 7,5cm, com duas rodas acionadas diferencialmente por dois motores de corrente contínua com acoplamento direto (Guerra et al., 2002), conforme esquema da figura 3. As principais grandezas envolvidas na modelagem estão descritas na tabela 1, onde os índices *d* e *e* dizem respeito ao lado direito e esquerdo, respectivamente.

2.1 Modelagem Cinemática

A relação entre velocidades no espaço de atuadores (ω_r) e no referencial do robô (ω) é:

$$\mathbf{v} = {}^v\mathbf{T}_\omega \cdot \omega_r \rightarrow \omega_r = \omega \mathbf{T}_v \cdot \mathbf{v} \quad (1)$$

onde:

$$\mathbf{v} = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad \omega_r = \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} \quad {}^v\mathbf{T}_\omega = \begin{bmatrix} r_d/2 & r_e/2 \\ r_d/b & -r_e/b \end{bmatrix}$$

O robô modelado é sujeito a restrições não holonômicas, isto é, restrições nas possíveis velocidades que pode assumir, devido ao fato que o atrito impede que as rodas deslizem

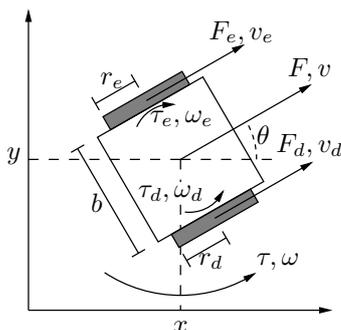


Figura 3: Diagrama esquemático do robô

x, y	posição do centro de massa do robô
θ	orientação do robô
b	distância entre as duas rodas
r_d, r_e	raios das rodas
ω_d, ω_e	velocidades angulares das rodas
ω	velocidade angular do robô
v_d, v_e	velocidades lineares na borda das rodas
v	velocidade linear do robô
τ_d, τ_e	torques nas rodas
τ	torque resultante no robô
f_d, f_e	forças na borda das rodas
f	força resultante no robô
e_d, e_e	tensões de armadura nos motores
i_d, i_e	correntes de armadura nos motores
R_d, R_e	resistência de armadura nos motores
K_d, K_e	constantes contra-eletromotrizes nos motores
J_d, J_e	momento de inércia dos rotores dos motores
β_d, β_e	coeficiente de atrito nos motores
m	massa do robô
J	momento de inércia do robô
β_l	coeficiente de atrito viscoso linear no robô
β_θ	coeficiente de atrito viscoso angular no robô

Tabela 1: Grandezas dinâmicas para a modelagem do robô

lateralmente. Assim, o robô só pode se movimentar na direção em que está orientado. Esta restrição é expressa matematicamente como:

$$\frac{dy}{dx} = \frac{\sin \theta}{\cos \theta} \Rightarrow \dot{y} \cos \theta - \dot{x} \sin \theta = 0 \quad (2)$$

A relação entre velocidades no referencial do robô e velocidades em espaço de configuração é dada por:

$$\dot{\mathbf{q}} = {}^a\mathbf{T}_v \cdot \mathbf{v} \quad (3)$$

onde:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad {}^a\mathbf{T}_v = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}$$

2.2 Modelagem Dinâmica

A partir das restrições não-holonômicas, a seguinte relação entre esforços em espaço de atuadores τ e esforços no referencial do robô \mathbf{f}_θ é obtida:

$$\mathbf{f}_\theta = \omega \mathbf{T}_v^T \cdot \tau \Rightarrow \tau = {}^v\mathbf{T}_\omega^T \cdot \mathbf{f}_\theta \quad (4)$$

onde:

$$\mathbf{f}_\theta = \begin{bmatrix} f_\theta \\ \tau_\theta \end{bmatrix} \quad \tau = \begin{bmatrix} \tau_d \\ \tau_e \end{bmatrix} \quad \omega \mathbf{T}_v^T = \begin{bmatrix} 1/r_d & 1/r_e \\ b/2r_d & -b/2r_e \end{bmatrix}$$

A dinâmica dos atuadores (motores) é dada pelas equações elétricas (5) abaixo. A modelagem dos atritos leva em conta apenas o atrito viscoso linear e despreza os efeitos não-lineares (atrito seco e de Coulomb).

$$\mathbf{i} = -\rho \mathbf{K}_r \omega_r + \rho \mathbf{e} \quad \tau = \mathbf{K}_r \cdot \mathbf{i} - \mathbf{J}_r \dot{\omega}_r - \beta_r \omega_r \quad (5)$$

x_b, y_b	posição do centro de massa da bola
v_b	velocidade linear da bola
θ_b	ângulo de movimentação da bola
β_b	coeficiente de atrito viscoso linear da bola
r_b	raio da bola
m_b	massa da bola

Tabela 2: Grandezas dinâmicas para a modelagem da bola

onde:

$$\mathbf{i} = \begin{bmatrix} i_d \\ i_e \end{bmatrix} \quad \boldsymbol{\omega}_r = \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} \quad \mathbf{e} = \begin{bmatrix} e_d \\ e_e \end{bmatrix}$$

$$\mathbf{K}_r = \begin{bmatrix} K_d & 0 \\ 0 & K_e \end{bmatrix} \quad \boldsymbol{\beta}_r = \begin{bmatrix} \beta_d & 0 \\ 0 & \beta_e \end{bmatrix}$$

$$\boldsymbol{\rho} = \begin{bmatrix} 1/R_d & 0 \\ 0 & 1/R_e \end{bmatrix} \quad \mathbf{J}_r = \begin{bmatrix} J_d & 0 \\ 0 & J_e \end{bmatrix}$$

A equação de movimento do robô pode ser obtida a partir das leis de Newton e Euler:

$$\mathbf{f}_\theta = \mathbf{J}_R \cdot \dot{\mathbf{v}} + \boldsymbol{\beta}_R \cdot \mathbf{v} \quad (6)$$

onde:

$$\mathbf{J}_R = \begin{bmatrix} m & 0 \\ 0 & J \end{bmatrix} \quad \boldsymbol{\beta}_R = \begin{bmatrix} \beta_l & 0 \\ 0 & \beta_\theta \end{bmatrix}$$

Substituindo as equações de dinâmica de atuadores (5) na equação de movimento do robô (6), obtém-se a equação 7 que expressa, no referencial do robô, o comportamento dinâmico do robô.

$$\mathbf{K} \cdot \mathbf{e} = \mathbf{M} \cdot \dot{\mathbf{v}} - \mathbf{B} \cdot \mathbf{v} \quad (7)$$

onde:

$$\mathbf{K} = \boldsymbol{\omega} \mathbf{T}_v^T \cdot \boldsymbol{\rho} \cdot \mathbf{K}_r \quad \mathbf{M} = \mathbf{J}_R + \boldsymbol{\omega} \mathbf{T}_v^T \cdot \mathbf{J}_r \boldsymbol{\omega} \mathbf{T}_v$$

$$\mathbf{B} = \boldsymbol{\beta}_R + \boldsymbol{\omega} \mathbf{T}_v^T [\boldsymbol{\rho} \cdot \mathbf{K}_r^2 + \boldsymbol{\beta}_r] \cdot \boldsymbol{\omega} \mathbf{T}_v$$

O modelo descrito acima pode ser representado no espaço de estados pela equação 8 a seguir:

$$\left\{ \begin{array}{l} \begin{bmatrix} \dot{v} \\ \dot{\omega} \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -\mathbf{M}^{-1} \mathbf{B} & \vdots & \mathbf{0} \\ \dots & \dots & \dots \\ \cos(\theta) & 0 & \vdots & \mathbf{0} \\ \sin(\theta) & 0 & \vdots & \mathbf{0} \\ 0 & 1 & & \end{bmatrix} \begin{bmatrix} v \\ \omega \\ x \\ y \\ \theta \end{bmatrix} \\ + \begin{bmatrix} \mathbf{M}^{-1} \mathbf{K} \\ \dots & \dots \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} e_d \\ e_e \end{bmatrix} \\ \mathbf{y} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ x \\ y \\ \theta \end{bmatrix} \end{array} \right. \quad (8)$$

O comportamento dinâmico da bola também precisa ser modelado para ser simulado. A partir das grandezas dinâmicas definidas na tabela 2, pode-se chegar com relativa

facilidade à conclusão que o comportamento da bola é regido pelo modelo da equação 9.

$$\left\{ \begin{array}{l} \begin{bmatrix} \dot{v}_b \\ \dot{x}_b \\ \dot{y}_b \end{bmatrix} = \begin{bmatrix} \beta_b & 0 & 0 \\ \cos(\theta_b) & 0 & 0 \\ \sin(\theta_b) & 0 & 0 \end{bmatrix} \begin{bmatrix} v_b \\ x_b \\ y_b \end{bmatrix} \\ \mathbf{y}_b = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_b \\ x_b \\ y_b \end{bmatrix} \end{array} \right. \quad (9)$$

O simulador dinâmico se baseia essencialmente na integração numérica de várias instâncias da equação 8 (uma para cada mini-robô) e da equação 9. A integração numérica pode ser feita por qualquer dos métodos usuais, como o de Euler e o Runge-Kutta de quarta ordem (Garcia, 1997). Para garantir um maior realismo no sistema, resta tratar as colisões e incluir os seus efeitos na simulação.

3 TRATAMENTO DAS COLISÕES

Existem quatro tipos de colisões possíveis no ambiente do futebol de robôs: colisão bola-parede, colisão robô-parede, colisão robô-bola e colisão robô-robô, três das quais estão ilustradas na figura 4. O tratamento de uma colisão envolve a detecção de sua ocorrência (seção 3.1) e a inclusão dos seus efeitos na simulação (seção 3.2).

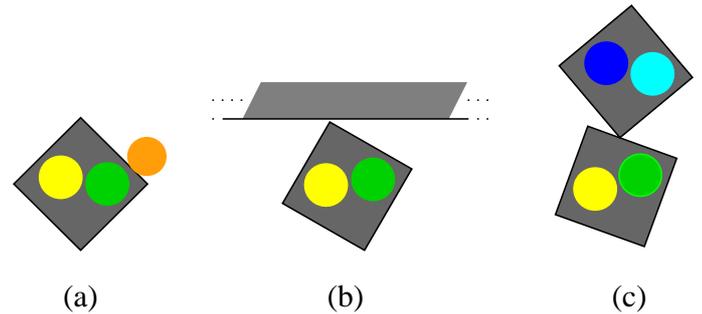


Figura 4: Exemplo das colisões no simulador de futebol de robôs: (a) colisão robô-bola, (b) robô-parede e (c) robô-robô

3.1 Detecção de Colisões

Nas colisões robô-parede, assume-se que os robôs batem apenas com os vértices; nas colisões robô-robô, adota-se que é sempre um dos vértices de um robô que bate no lado de outro. Presume-se também que a bola sempre bate em um lado do robô, nunca em um vértice. Essas simplificações são feitas pois os possíveis outros tipos de colisão (por exemplo, uma colisão perfeitamente frontal dentre dois robôs) têm uma probabilidade mínima de acontecer.

3.1.1 Colisão bola-parede

A colisão bola parede é a mais simples de ser detectada porque as paredes são estáticas e a bola é esférica. Para detectar se houve colisão ou não, basta calcularmos a distância entre o centro da bola e a parede: se essa distância for menor ou igual que o raio da bola, houve colisão.

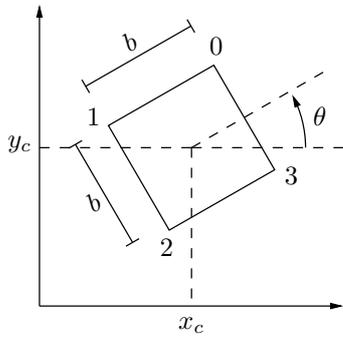


Figura 5: Cálculo dos vértices de um robô

3.1.2 Colisão robô-parede

Para verificar se ocorreu colisão de algum dos robôs com as paredes, devemos primeiro calcular a posição dos seus quatro vértices e em seguida verificar se algum deles ultrapassa uma parede em qualquer das suas componentes (x ou y). Se um robô possui lado b , seu centro está nas coordenadas (x_c, y_c) e está rotacionado de um ângulo θ , conforme a figura 5, as coordenadas (x_j, y_j) do seu j -ésimo vértice, $j = 0, 1, 2, 3$, são dadas por:

$$\begin{cases} x_j = x_c + \frac{b}{\sqrt{2}} \cdot \cos(\theta + 45^\circ + 90^\circ \cdot j) \\ y_j = y_c + \frac{b}{\sqrt{2}} \cdot \sin(\theta + 45^\circ + 90^\circ \cdot j) \end{cases}$$

3.1.3 Colisão robô-bola

A colisão robô-bola utiliza os mesmos princípios da colisão bola-parede. Porém aqui, devemos calcular a distância entre o centro da bola e cada uma das quatro semi-retas que caracterizam os lados do robô. Porém, para diminuir o tempo de processamento, antes de testar a distância da bola com cada uma das quatro semi-retas de cada um dos robôs, realizamos um teste de proximidade. É calculada a distância entre o centro da bola e o centro do robô: se essa distância for maior que a soma do raio da bola mais o raio da circunferência que circunscreve o robô, os outros testes não são necessários, pois não há possibilidade de colisão.

A semi-reta que passa pelo j -ésimo e pelo $(j + 1)$ -ésimo vértice do robô tem a seguinte equação paramétrica, onde o parâmetro adimensional λ varia de 0 a 1:

$$\begin{cases} x(\lambda) = x_j + (x_{j+1} - x_j) \cdot \lambda \\ y(\lambda) = y_j + (y_{j+1} - y_j) \cdot \lambda \end{cases} \quad (10)$$

Se a bola tem centro (x_b, y_b) , o ponto da reta mais próximo da bola ocorre para o seguinte valor de λ :

$$\lambda = \frac{(x_b - x_j) \cdot (x_{j+1} - x_j) + (y_b - y_j) \cdot (y_{j+1} - y_j)}{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \quad (11)$$

Para testar a colisão entre a bola e um dos lados de um robô, o valor de λ deve ser calculado pela equação 11 e eventualmente reduzido ao intervalo $[0,1]$. Em seguida o ponto $(x(\lambda), y(\lambda))$ é calculado pela equação 10 e verifica-se a distância do ponto ao centro da bola (x_b, y_b) : se a distância for menor ou igual que o raio da bola, houve colisão.

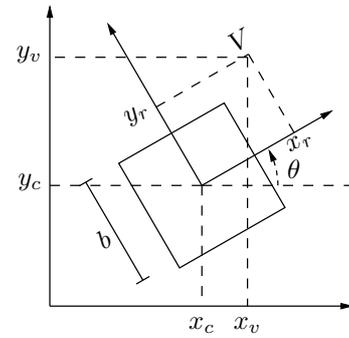


Figura 6: Pertinência de um ponto a um quadrado

3.1.4 Colisão robô-robô

O teste consiste em verificar se cada vértice de um robô se situa dentro da região delimitada pelas quatro semi-retas que caracterizam os lados do outro robô. Dado dois robôs, este teste deve ser feito do primeiro em relação ao segundo e vice-versa.

O teste sobre se um vértice $V = (x_v, y_v)$ está dentro do quadrado é facilitado calculando-se as coordenadas (x_r, y_r) deste ponto no sistema de coordenadas associado ao robô, conforme indica a figura 6. Assim, um ponto (x_v, y_v) está dentro do robô se:

$$-b/2 \leq x_r \leq b/2 \quad -b/2 \leq y_r \leq b/2$$

onde as grandezas x_r e y_r são dadas por:

$$\begin{aligned} x_r &= \sqrt{(x_v - x_c)^2 + (y_v - y_c)^2} \cdot \cos\left(\tan^{-1} \frac{y_v - y_c}{x_v - x_c} - \theta\right) \\ y_r &= \sqrt{(x_v - x_c)^2 + (y_v - y_c)^2} \cdot \sin\left(\tan^{-1} \frac{y_v - y_c}{x_v - x_c} - \theta\right) \end{aligned}$$

3.2 Modelagem das Colisões

A conservação da energia cinética e do momento linear permitem modelar o comportamento de duas esferas após um choque inelástico. Pode-se partir desta situação e modelar os quatro tipos de colisões necessários para o simulador.

No choque entre duas esferas da figura 7, o eixo y é definido pelos dois centros de massa e o eixo x é perpendicular a y e passa pelo ponto de contato. As duas esferas têm massas e velocidades m_1 e v_1 e m_2 e v_2 , respectivamente.

Após a colisão, as componentes das velocidades v_1 e v_2 na direção x permanecem inalteradas. As componentes na di-

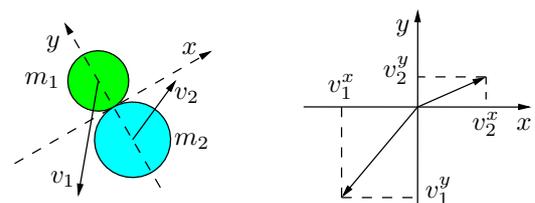


Figura 7: Colisão entre duas esferas



Figura 8: Colisão envolvendo a parede

reção y são alteradas de acordo com a equação 12:

$$\begin{aligned} v_1^y &\leftarrow \frac{m_1 - m_2}{m_1 + m_2} v_1^y + \frac{2m_2}{m_1 + m_2} v_2^y \\ v_2^y &\leftarrow \frac{m_2 - m_1}{m_1 + m_2} v_2^y + \frac{2m_1}{m_1 + m_2} v_1^y \end{aligned} \quad (12)$$

Com as componentes na direção x (inalterados) e y (novos), calculam-se as novas velocidades v_1 e v_2 dos objetos.

A bola se enquadra perfeitamente na modelagem da colisão entre duas esferas: o módulo da nova velocidade substituirá v_b e a orientação substituirá θ_b na equação 9 da simulação. Nas colisões envolvendo a parede e um objeto, a parede pode ser modelada como uma esfera de velocidade nula, massa infinita e posicionada de modo a preservar o mesmo ponto de colisão, conforme a figura 8

Um robô pode ser modelado como uma esfera cuja massa coincide com a do robô, cujo centro de massa está posicionado de forma a gerar o mesmo ponto de contato com o outro objeto e com velocidade igual (em módulo e orientação) à velocidade linear do ponto de colisão. A velocidade deste ponto é a soma vetorial das componentes geradas pelas velocidades linear (\vec{v}_{lin}) e de rotação (\vec{v}_{rot}) do robô, como mostra a figura 9. \vec{v}_{lin} tem a mesma orientação e módulo que a velocidade linear do robô \vec{v} . \vec{v}_{rot} é perpendicular ao vetor que vai do centro de massa do robô ao ponto de contato e tem módulo $|\vec{v}_{rot}| = |\omega| \cdot d$, onde ω é a velocidade angular do robô e d a distância entre o ponto de colisão e o centróide do robô.

A figura 10 apresenta um exemplo de colisão entre dois robôs, onde se destacam os vetores das velocidades compostas antes da colisão e as esferas equivalentes. Uma vez calculada a nova velocidade da esfera, ela deve ser decomposta nas suas componentes \vec{v}_{lin} e \vec{v}_{rot} , a partir das quais se determinam os novos valores de v e ω a serem considerados na equação 8 do robô correspondente.

4 O SISTEMA DE SIMULAÇÃO

O sistema de simulação envolve duas componentes: o simulador propriamente dito e o visualizador. O simulador roda em um processo dedicado (uma *thread*) de alta priori-

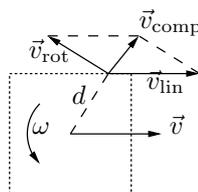


Figura 9: Composição de velocidades em um ponto

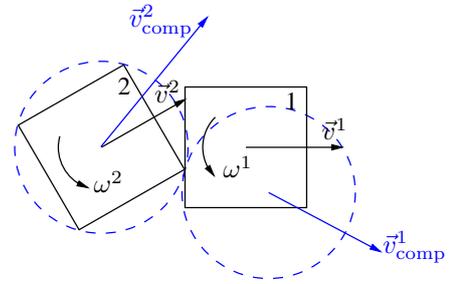


Figura 10: Exemplo da colisão entre dois robôs e a disposição dos vetores velocidade composta e o eixo de colisão

dade. Os dados simulados ficam disponíveis em uma memória compartilhada, onde podem ser lidos pelo visualizador, como mostra a figura 11. O visualizador funciona em outro processo e exibe os dados para o usuário. Esta configuração evita que eventuais atrasos ou um grande consumo de tempo do visualizador comprometam as características de tempo real do simulador.

4.1 O simulador

O funcionamento do simulador segue o algoritmo simplificado a seguir:

```

Repetir
| Esperar até que:
| | Chegar sinal de controle
| | ou
| | Completar 30ms do envio da última imagem
| fim_Esperar
| Ler tempo atual
| Simular até o tempo atual
| Se 30ms desde a última imagem enviada:
| | Gerar e enviar imagem
| fim_Se
fim_Repetir

```

A simulação é feita utilizando um método de integração numérica. O algoritmo a seguir ilustra a utilização do método de Euler, onde x representa o vetor de estados nas equações 8 e 9 e Δt é o passo de integração:

```

t, t_atual, delta_t
Ler t_atual
Enquanto t < t_atual:
| Calcular dx (equações 8 ou 9)
| x <- x + dx*delta_t
| Testar colisões:
| | Bola com cada parede

```

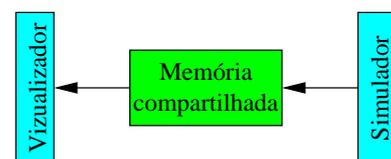


Figura 11: Sistema completo de simulação



Figura 12: Imagem do visualizador

```

| | Para cada robô:
| | | Robô com cada parede
| | | Robô com bola
| | | Robô com cada robô
| | Se colisão
| | | Alterar x
| | fim_Se
| t <- t + delta_t
fim_Enquanto

```

Para diminuir o tempo de geração e de envio da imagem artificial, ao invés de transmitir a cada passo a imagem do campo inteiro ($640 \times 480 \times 3$ bytes), o simulador no início transmite uma imagem do campo vazio, capturada antes do jogo. O simulador gera e envia a cada passo apenas as alterações em relação a esta imagem, ou seja, as imagens dos seis robôs e da bola, o que dá aproximadamente 2% da imagem total. A transmissão destas imagens em uma rede *Fast Ethernet* (100 Mb/s) leva em torno de 1,5 ms. O computador do sistema de visão simplesmente sobrepõe as imagens dos robôs e da bola na imagem já existente do campo. O tempo para este processamento combinado com o tempo de transmissão é da mesma ordem de grandeza que o tempo que a placa de aquisição de imagens leva para disponibilizar uma imagem real para o computador.

4.2 O visualizador

O visualizador lê as informações diretamente da memória compartilhada, podendo funcionar na mesma máquina, em uma máquina remota comunicando via *socket* ou até não ser executado. A interface gráfica para o usuário foi feita utilizando a biblioteca gráfica OpenGL (Woo et al., 1999). A taxa de exibição de imagens (2 imagens por segundo) é bem menor que a taxa de simulação (30 simulações por segundo). Na figura 12 é mostrada uma imagem da interface com o usuário do visualizador.

5 RESULTADOS E CONCLUSÕES

O algoritmo de simulação, incluindo o tratamento de colisões, é bastante rápido, pois envolve apenas equações facilmente computáveis. Sua complexidade é de $\mathcal{O}(m \times n^2)$, onde $n = 6$ é o número de robôs e $m = 33\text{ms}/\Delta t$ é o número de passos de integração. Em computadores PC relativamente recentes, rodando sistema operacional Linux e utilizando um passo de integração de 1 ms, o cálculo da

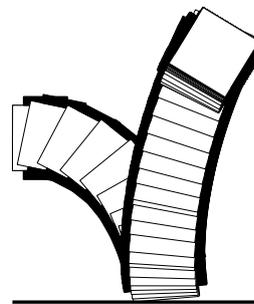


Figura 13: Comportamento do robô simulado em uma colisão contra a parede

nova situação do sistema é feita em tempo bastante inferior a 1 ms, o que é muito satisfatório tendo em vista que só é necessário o cálculo de uma nova situação a cada 30 ms.

O comportamento dos robôs simulados se parece bastante com o que se esperaria de um robô real. Para exemplificar, a figura 13 apresenta o comportamento do robô numa situação de colisão com a parede inferior do campo de futebol em um instante em que as tensões aplicadas nos motores são nulas. Percebe-se que o simulador de colisões reproduz, de maneira satisfatória, uma colisão real de um veículo com rodas contra uma parede.

REFERÊNCIAS

- Cook, G. E., Biegl, C., Springfield, J. F. e Fernandez, K. R. (1994). An intelligent robotics simulator, *IEEE Industry Applications Society Annual Meeting*, Vol. 3, Denver, CO, USA, pp. 1793–1800.
- Garcia, C. (1997). *Modelagem e Simulação*, Editora da Universidade de São Paulo, Av. Prof. Luciano Gualberto, Travessa J, 374, 05508, São Paulo, SP, Brasil. ISBN 85-314-0402-9.
- Guerra, P. N., Yamamoto, M. M. e Alsina, P. J. (2002). Um controlador adaptativo para um robô móvel com rodas: Análise e simulação, *CBA - Congresso Brasileiro de Automática*, Natal, RN, Brazil.
- Kimoto, K. e Yuta, S. (1992). A simulator for programming the behavior of an autonomous sensor-based mobile robot, *IROS - IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2, pp. 1431–1438.
- Kulkarni, J. A., Byrd, J. S. e Pettus, R. O. (1991). X-window based graphical mobile robot simulator for path planing and sensor fusion experiments, *IEEE Simulation Symposium*, New Orleans, LA, USA, pp. 185–192.
- Kulkarni, J. A. e Pettus, R. O. (1991). Distributing a mobile robot simulator using remote procedure calls, *IEEE Southeastern Symposium on System Theory*, Columbia, SC, USA, pp. 500–504.
- Woo, M., Neider, J., Davis, T. e Shreiner, D. (1999). *OpenGL Programming Guide*, 3 edn, Addison-Wesley, EUA. ISBN 0-20-160458-2.