

### 3 Simulador Desenvolvido

O simulador de veículos robóticos desenvolvido para esta dissertação tem mais de 9.690 linhas de códigos e foi todo escrito em C++ (Win32/API), com a utilização da interface gráfica OpenGL. Denominado VirtualBotz 3D, é uma generalização 3D de trabalhos feitos em 2D por Barral [4] e Santos [5]. Houve um cuidado particular no realismo do sistema, em especial na modelagem da dinâmica, para permitir características não-lineares em terrenos acidentados que revelem ter alto grau de importância para o comportamento de robôs reais, como o Veículo para Inspeção Visual Interna de dutos (VIVI, Figura 16) desenvolvido pelo Laboratório de Robótica da PUC-Rio para inspeção de tubulações.

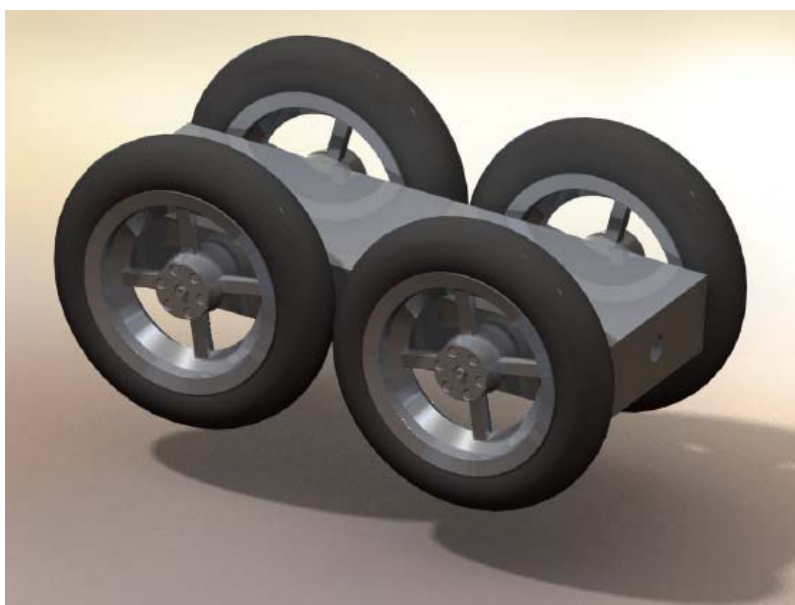


Figura 16 – Veículo para Inspeção Visual Interna de dutos (VIVI)

O simulador desenvolvido permite a visualização 3D em tempo real da interação entre robôs móveis e os mais diversos tipos de terrenos, além de

- importar perfis de terrenos em forma de matriz de elevação e mapa de altura, inclusive com a possibilidade de construir perfis de terrenos a partir de funções;
- importar texturas para o terreno e alterar escalas a partir do script do simulador;
- importar diferentes partes do robô, como chassi e roda, por meio de arquivos no formato wavefront (.OBJ), com características visuais de materiais;
- implementar diferentes tipos de algoritmos de controle;
- interagir com o veículo robótico no ambiente virtual em tempo real com a utilização de um joystick analógico;
- alterar o ponto de vista utilizando o teclado ou o joystick;
- utilizar diferentes câmeras, permitindo a visualização de diferentes ângulos, com ajustes também pelo teclado ou joystick, dependendo do tipo de câmera;
- gravar dados do sistema como velocidades, acelerações, torques, forças, etc., em arquivos texto, para posterior análise do algoritmo de controle ou comportamento dinâmico do sistema;
- gravar a simulação em arquivo de vídeo em diferentes tipos e formatos;
- poder implementar diferentes tipos de motores de corrente contínua usados na locomoção do sistema, mediante a alteração dos parâmetros em um arquivo de script em formato texto;
- visualizar a trajetória do veículo robótico na simulação por meio de vetores ou linha contínua, com a possibilidade de comparar trajetórias ao final de uma simulação;
- poder, após a simulação, posicionar o veículo robótico em um ponto específico da sua trajetória;
- se o computador tiver mais de um processador, poder reservar um processador exclusivo para executar os cálculos dinâmicos;
- poder utilizar um passo temporal de integração da ordem de  $10^{-7}$ s;
- poder ajustar a frequência para o controle.

Os arquivos de códigos gerados em linguagem C++ estão divididos em pastas, para uma percepção clara e imediata de sua organização (**Anexo A**), e separados em classe de negócio e de interface. O simulador pode ser transportado para qualquer outra plataforma sem nenhuma dificuldade. As pastas estão divididas em

- *\_library* – contendo os arquivos com definição de tipos, variáveis globais e pastas de classes de negócio e interface, dentre outras
- *\_library/Botz* – classes de negócio
- *\_library/Graph* – classes gráficas para leitura de arquivos de imagens e outros arquivos do gênero
- *\_library/Math* – classes para matrizes e vetores
- *\_library/OpenGL* – classes de interface
- *\_library/Tools* – ferramentas para manipulação de matrizes (*arrays*) de ponteiros e geração de valores aleatórios

No **Anexo B** foram gerados os diagramas em UML (*Unified Modeling Language*), para que se tenha uma visão clara da organização das classes envolvidas no sistema, assim como seus relacionamentos. É bom salientar que o padrão de linguagem gráfica UML [46] é uma ferramenta que permite a visualização em gráficos padronizados, em forma de diagrama, e que não tem a capacidade de informar quais passos devem ser tomados e muito menos como projetar o sistema. Por trás de todas as suas notações gráficas, o UML acaba gerando a semântica do sistema, ferramenta de extrema importância para os desenvolvedores.

### 3.1.

#### Descrição Geral

Na modelagem do simulador, foi utilizado o conceito de modularização da Programação Orientada a Objetos (**Anexo B**). Assim que o programa é executado, a janela principal exibe uma imagem do terreno e do veículo robótico, que será chamada de “mundo virtual”, ou simplesmente “mundo”. O veículo robótico se

movimentará comandado apenas por um joystick analógico conectado ao computador. Assim como o veículo robótico, o mundo virtual também pode ser movimentado, ou melhor, reposicionado, utilizando o próprio joystick, caso tenha dois manetes distintos, ou então, o mouse. Ambos os manetes de reposicionamento do mundo permitem a sua rotação, aproximação ou afastamento, e o reposicionamento do observador (**Anexo C**).

Para a simulação, utilizam-se dois arquivos-texto contendo toda configuração da simulação. No primeiro arquivo, o principal (**Anexo E**), apenas se faz referência ao segundo arquivo de simulação/configuração, propriamente dito, onde se encontram as definições do terreno e do veículo robótico (**Anexo G e I**, respectivamente). Dessa forma, é possível guardar várias configurações a serem executadas assim que necessário, apenas comentando ou não a linha contendo o arquivo corrente de simulação/configuração. Assim como os arquivos contendo os códigos do simulador, o código executável e seus acessórios também estão organizados em pastas bastante intuitivas:

- *Build* – pasta contendo os arquivos executáveis do simulador, tanto para a versão de teste, chamada de beta, quanto para a versão final, chamada de release. Nesta pasta também estão os arquivos scripts para configuração da simulação.
- *Build/Heightmaps* – pasta com imagens para mapas de alturas do terreno.
- *Build/Obj\_files* – pasta com todos os arquivos de modelo para terreno do tipo malha regular, *mesh* (.msh), para rodas e chassi do tipo wavefront (.OBJ), incluindo os arquivos de materiais (.MTL) para os modelos wavefront.
- *Build/Report* – pasta com todas as saídas de dados geradas em arquivos específicos durante uma simulação.
- *Build/Scripts* – pasta com todos os scripts de simulações.
- *Build/Tests* – pasta com arquivos com coordenadas *X*, *Y* e *Z* para comparar trajetórias do veículo robótico.

- *Build/Textures* – pasta com todas as texturas a serem utilizadas pelo simulador.
- *Build/Videos* – pasta com todos os arquivos de vídeo gerados para cada simulação, se necessário.

### 3.2.

#### Modelagem do Terreno

Os modelos mais utilizados na representação de relevos (terrenos acidentados) são curvas de nível, grade regular retangular conhecida também por *digital elevation model* (DEM), rede triangular irregular (*triangular irregular network* – TIN), conhecida por triangulação, e o diagrama de Voronoi.

Para o simulador VirtualBotz 3D foi utilizado o modelo grade regular retangular (Figura 17). Os modelos de terrenos (**Anexo F**) podem ser criados a partir do conjunto de amostras do terreno ou por qualquer modelador geométrico 3D que possibilite a armazenagem dos dados em forma de matriz de elevação.

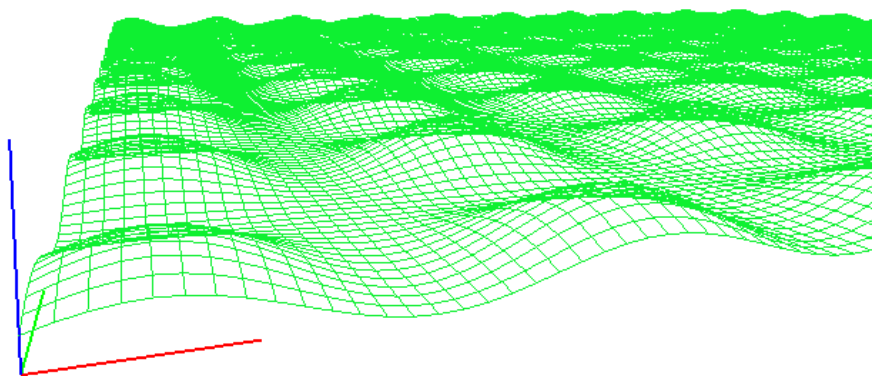


Figura 17 – Modelo de grade regular retangular utilizado pelo VirtualBotz 3D

O terreno é especificado por meio de uma matriz de duas dimensões denominada  $\underline{M}$ , onde cada elemento em  $\underline{M}$  corresponde à coordenada  $z$ , ou altura do terreno, em um sistema de coordenadas  $(x, y)$  em um plano horizontal. Os

extremos do domínio do terreno no VirtualBotz 3D (Figura 18) têm as coordenadas  $(x_{min}, x_{max})$  e  $(y_{min}, y_{max})$ .

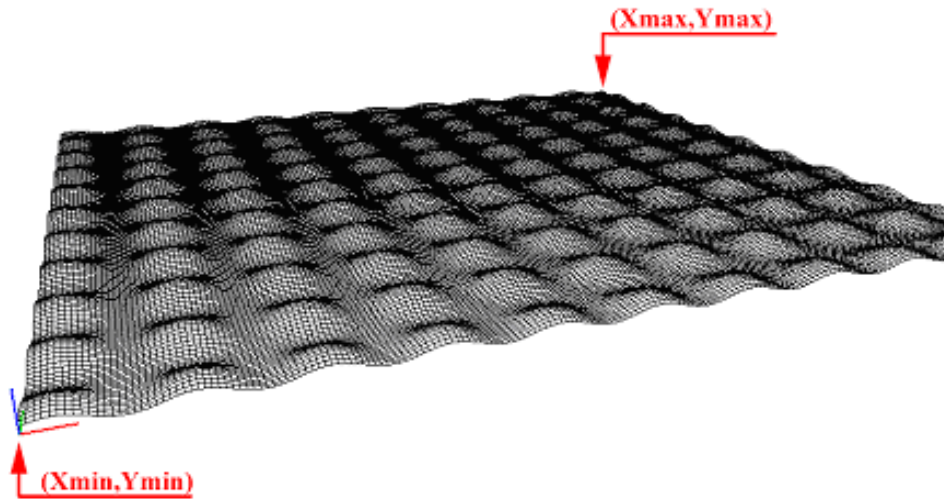


Figura 18 – Extremos do domínio do terreno no VirtualBotz 3D

$$x = x_{min} + i \frac{(x_{max} - x_{min})}{(n_x - 1)} \quad (3.1)$$

$$y = y_{min} + j \frac{(y_{max} - y_{min})}{(n_y - 1)} \quad (3.2)$$

Para os elementos  $i$  e  $j$  da matriz  $\underline{M}$ ,  $n_x$  e  $n_y$  são respectivamente o número de linhas e colunas de  $\underline{M}$ .

Para estabelecer a altura do terreno entre os elementos da matriz  $\underline{M}$ , foi implementado um algoritmo de interpolação linear que busca o elemento da matriz imediatamente anterior, tanto em  $x$  quanto em  $y$ , no ponto desejado, e o pondera, utilizando os outros três elementos vizinhos da seguinte forma:

$$z_1 = \underline{M}_{ix+1,iy} (aux_x - i_x) + \underline{M}_{ix,iy} (1 - (aux_x - i_x)) \quad (3.3)$$

$$z_2 = \underline{M}_{ix+1,iy+1} (aux_x - i_x) + \underline{M}_{ix,iy+1} (1 - (aux_x - i_x)) \quad (3.4)$$

$$z_f = z_2 (aux_y - i_y) + z_1 (1 - (aux_y - i_y)) \quad (3.5)$$

onde  $(i_x, i_y)$  são as coordenadas do elemento de  $\underline{M}$  imediatamente anterior ao elemento desejado,  $aux_x$  e  $aux_y$  são as coordenadas do ponto desejado convertidas para o sistema de coordenadas da matriz  $\underline{M}$ ,  $\underline{M}_{i,j}$  é o elemento  $(i, j)$  de  $\underline{M}$ , e  $z_f$  é o resultado da interpolação dos elementos do terreno na posição desejada. A Figura 19 ilustra as variáveis descritas.

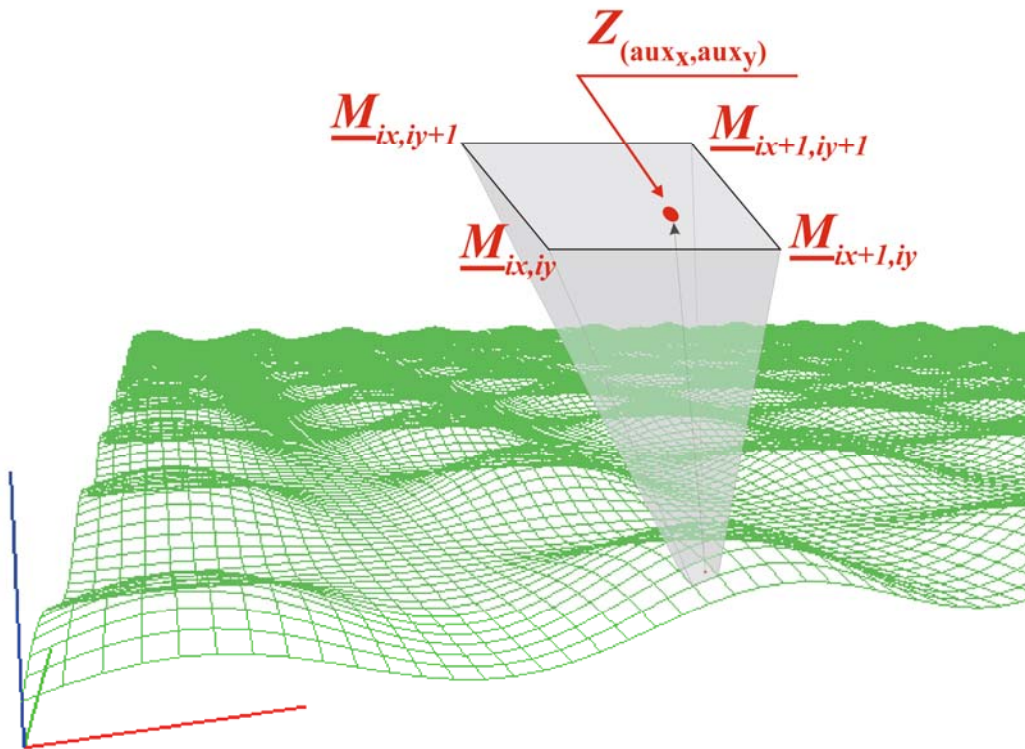


Figura 19 – Ilustração das variáveis de interpolação do terreno

### 3.3.

#### Modelagem do Veículo Robótico

##### 3.3.1.

##### Parâmetros Geométricos do Veículo Robótico

O veículo é modelado como um chassi rígido, contendo  $n$  rodas com suspensão independente do tipo mola-amortecedor não linear. Para visualizar a configuração do veículo robótico no simulador, vide o **Anexo I**.

Na Figura 20 encontra-se o veículo robótico e a localização de seu centro de massa, que recebe o nome de  $X_c$ . A sua orientação é obtida a partir de três vetores unitários  $\mathbf{n}$ ,  $\mathbf{t}$  e  $\mathbf{b}$ , respectivamente nas direções  $x$ ,  $y$  e  $z$ . As dimensões do chassi do veículo são definidas nessas três direções, havendo a possibilidade de posicionar o centro de massa de forma assimétrica com o uso de duas constantes em cada direção. Desta forma, pode-se dizer que a largura do chassi é dada pela soma de  $w_1$  com  $w_2$ , o comprimento pela soma de  $l_1$  e  $l_2$ , e a altura pela soma de  $h_{topo}$  e  $h_{base}$ , onde o primeiro é a distância entre o teto e o centro de massa do veículo e o segundo é a distância até o fundo do chassi. A posição do seu centro de massa é representada pelo vetor  $X_c$  representado em um sistema inercial global.

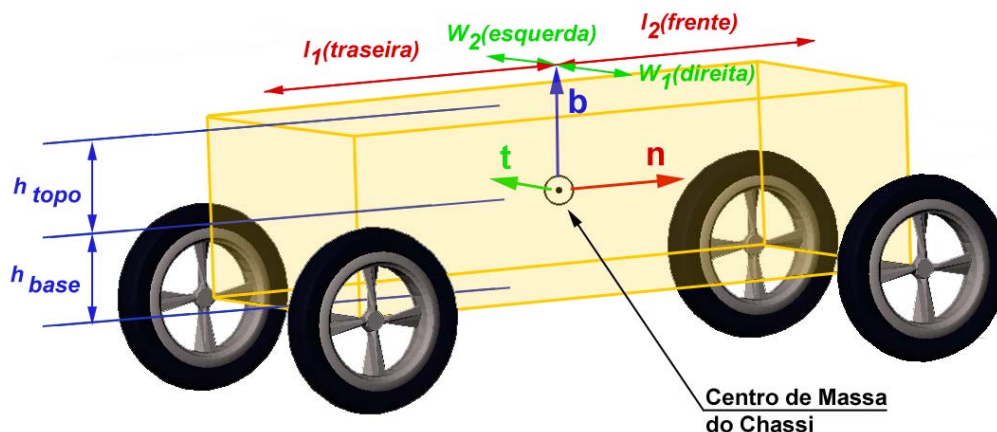


Figura 20 – Esquema com as dimensões do chassi



Cada roda do veículo tem o seu centro de massa (Figura 21) representado por três possíveis vetores:  $X_{teo}$  (posição teórica, sem nenhuma deformação da suspensão segundo o referencial global),  $X_{ntb}$  (posição teórica, sem nenhuma deformação da suspensão segundo o referencial **ntb** local) e  $X_{cm}$  (posição real, considerando os efeitos da suspensão segundo o referencial global).

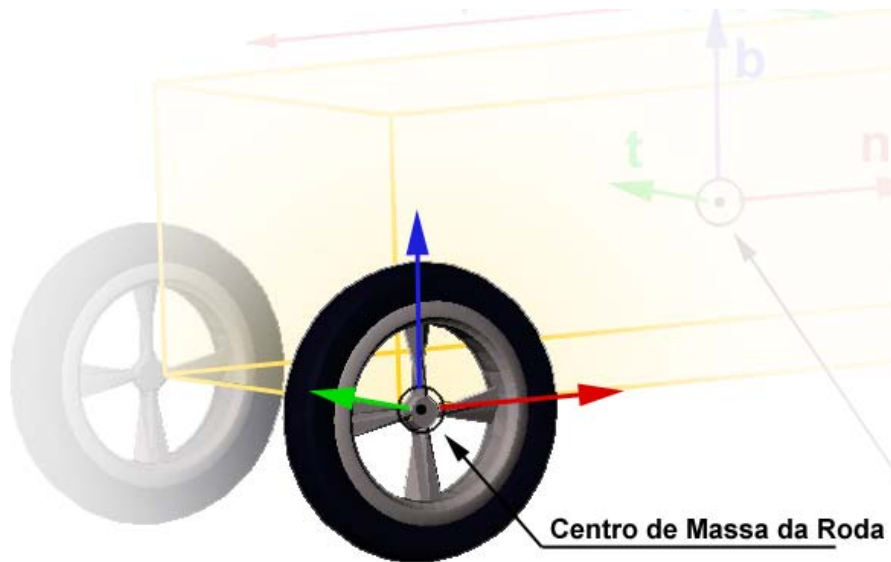


Figura 21 – Centro de massa teórico da roda, sem qualquer deformação na suspensão

A Figura 22 ilustra as dimensões da roda, apresentando o centro de massa teórico da roda sem qualquer deformação na suspensão, a uma distância  $(x, y, z)$  do centro de massa do veículo robótico, com raio  $r$  e largura representada por  $w$ . Na ilustração, a roda em destaque tem dois aros, um na parte mais externa e outro na parte mais interna da roda com o chassi. Logo, a coordenada  $y$  local de cada roda deverá ser o deslocamento em  $y$  do centro de massa do veículo  $(\pm w/2)$ .

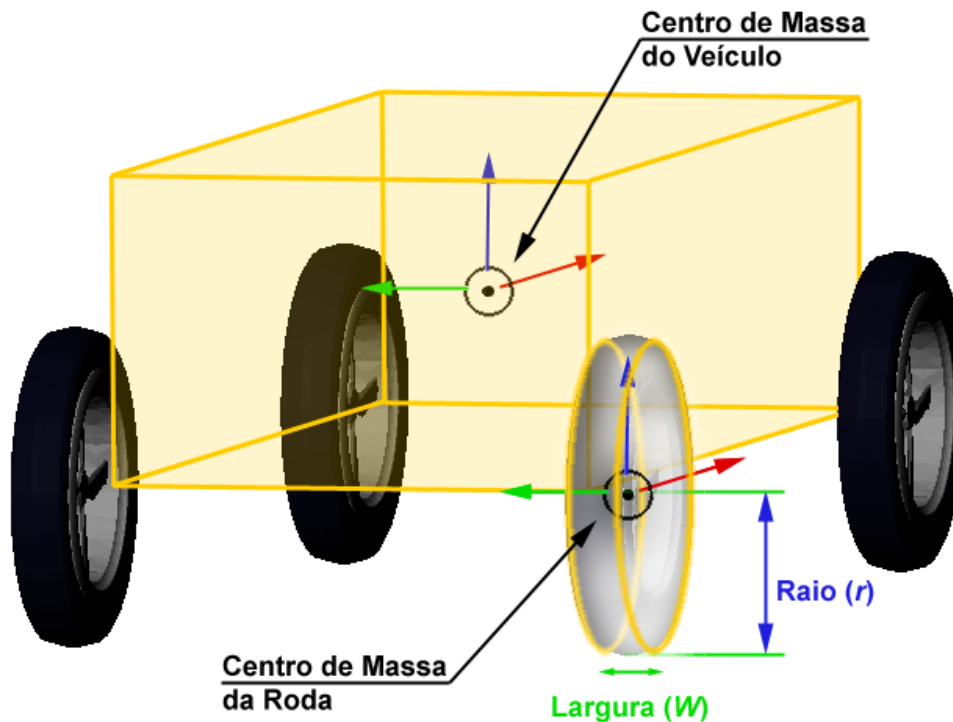


Figura 22 – Dimensões da roda

As definições das coordenadas locais do centro de massa de cada roda devem ser feitas no atributo “coordenadas local do centro de massa ( $x$ ,  $y$ ,  $z$ )” de cada roda do veículo robótico, localizado no arquivo script de configuração do simulador. Para maiores detalhes, vide os **Anexos E, G, I, J e K**.

### 3.3.2.

#### Forças Externas sobre o Veículo Robótico

As forças externas (Figura 23) sobre o veículo robótico podem ser descritas pelas forças de interação com o terreno e a força gravitacional  $\mathbf{P}$ . A força de interação com o terreno pode ser decomposta em três vetores ortogonais:  $\mathbf{F}_x$ , que representa a força de tração;  $\mathbf{N}$ , a força normal; e  $\mathbf{F}_y$ , a componente transversal.

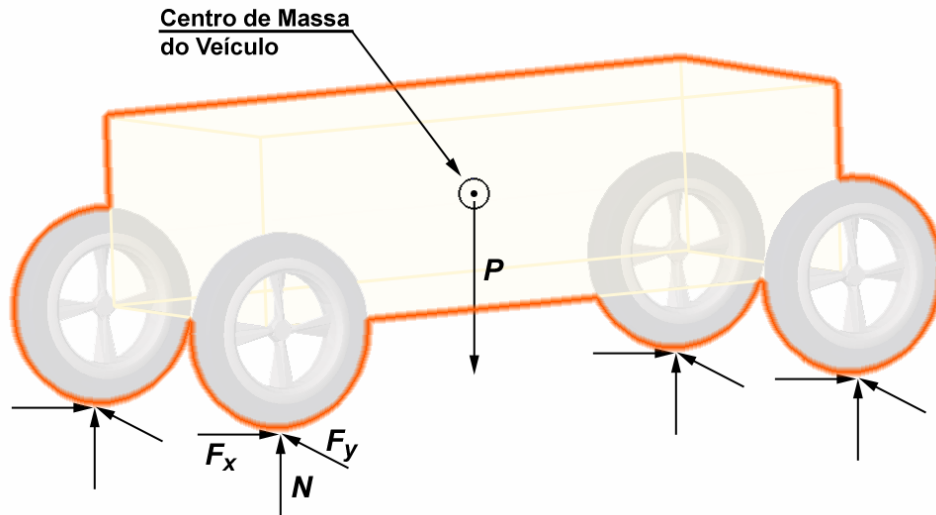


Figura 23 – Esquema de forças externas atuando sobre o veículo robótico

### 3.3.3.

#### Forças Internas entre a Roda e o Chassi

A Figura 24 mostra as forças de atuação nos eixos de cada roda. Para cada centro de massa de cada roda, tem-se a posição  $X_{cm}$ , que corresponde à posição real da roda com deformação da suspensão, segundo o referencial global. Há, também, no centro de massa de cada roda, vetores representando as componentes da força:  $F_{susp}$ , componente que representa a força da roda na direção  $b$ , atuando diretamente sobre a suspensão; e  $F_n$ , componente da força que impulsiona o chassi para a frente, na direção  $n$ .

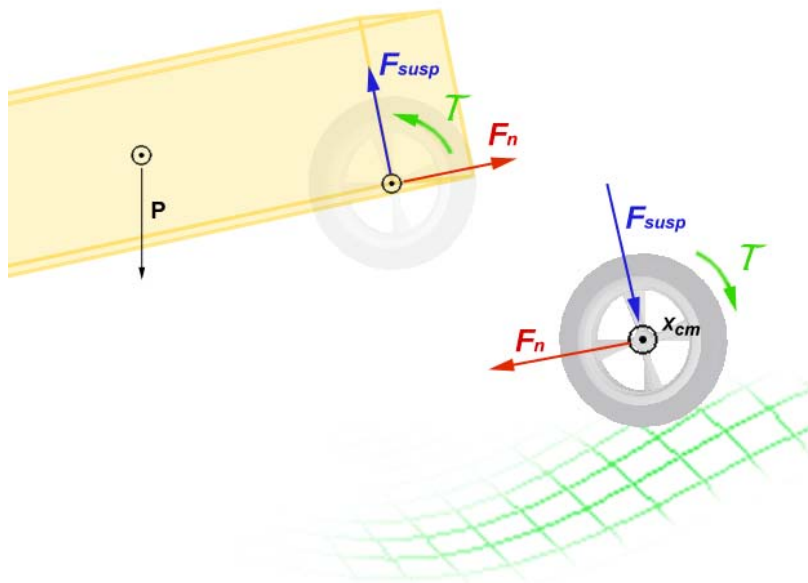


Figura 24 – Esquema de forças interna entre a roda e o chassi

### 3.3.4.

#### Algoritmo de Busca do Ponto de Contato entre Roda e Terreno

Nesta seção, todo o cálculo numérico das deformações das suspensões se acomodando ao terreno é apresentado, para determinar os pontos de contato entre cada roda e o terreno.

Para encontrar o ponto de contato da roda com o terreno, primeiro é preciso verificar o deslocamento da suspensão imposto pelo terreno (Figura 25). Na implementação do simulador, para evitar vibrações de alta frequência das suspensões, que exigiriam um passo de integração muito reduzido, modelaram-se as rodas como se não tivessem massa. Assim, a suspensão estará sempre em repouso, se não houver interação com o terreno, ou comprimida, se houver interação com o terreno, sendo impossível que a suspensão seja tracionada. Note, porém, que o momento de inércia de cada roda não é desprezado, o que influencia na aceleração do sistema.

Para o deslocamento de cada suspensão utiliza-se uma variável  $h$ . Admite-se que, caso esse deslocamento esteja dentro da faixa de valores de 0 (zero) até um valor de saturação  $h_{sat}$ , as equações para a força exercida pela suspensão são lineares; e, caso o deslocamento seja maior que o limite  $h_{sat}$ , utiliza-se uma relação não linear para a força.

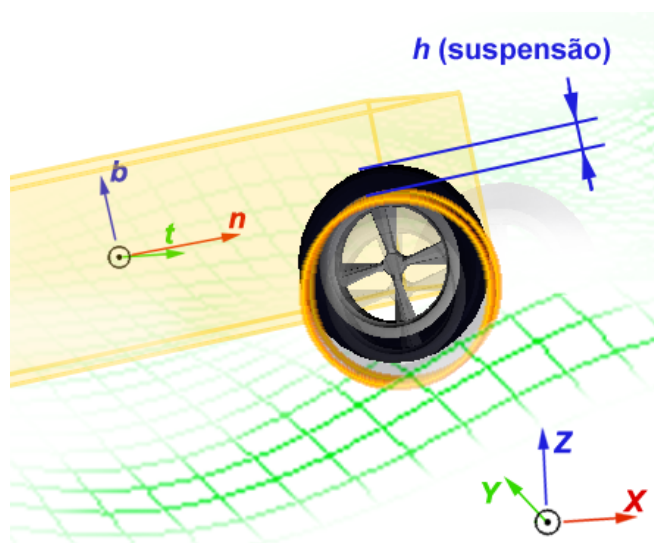


Figura 25 – Deslocamento imposto sobre a suspensão

Para determinar o ponto de contato da roda com o terreno, é preciso levar em conta que cada roda tem o seu centro geométrico representado por três possíveis vetores (como visto na Seção 3.3.1.):

- $X_{ntb}$  – posição teórica do centro geométrico da roda sem nenhuma deformação da suspensão, segundo o referencial **ntb** local
- $X_{teo}$  – posição teórica do centro da roda caso não houvesse nenhuma deformação da suspensão, segundo o referencial global
- $X_{cm}$  – posição real do centro da roda considerando os efeitos da suspensão, segundo o referencial global

O algoritmo de determinação do ponto de contato entre a roda e o terreno calcula primeiro o  $X_{teo}$ :

$$X_{teo} = X_{cm} + X_{ntb}(x) \cdot \mathbf{n} + X_{ntb}(y) \cdot \mathbf{t} + X_{ntb}(z) \cdot \mathbf{b} \quad (3.6)$$

Logo, uma vez determinado o centro geométrico teórico de cada roda, é preciso calcular o seu centro de massa, levando em consideração os efeitos da suspensão. Com isso, tem-se também o ponto de contato da roda com o terreno. Para determinar o deslocamento  $h$  da suspensão e o ponto de contato da roda com o terreno  $X_{cp}$ , discretiza-se a parte inferior da circunferência da roda em  $n_{fatias}$ , ou pontos (Figura 26).

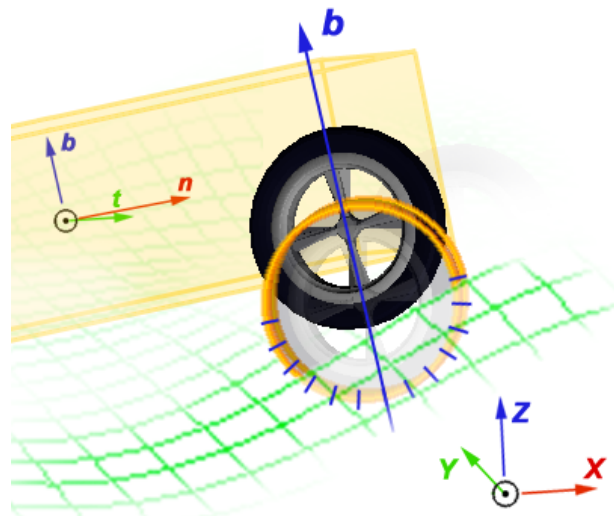


Figura 26 – Ilustração da discretização da circunferência inferior da roda

No script de simulação, descrito no **Anexo E e I**, há um atributo que permite escolher o valor da discretização dessa circunferência:

```
...
quantidade de divisoes da roda = 18
...
```

Um número muito elevado permite maior precisão no cálculo do ponto de contato roda-terreno, mas pode reduzir a velocidade de cálculo do simulador, dependendo do computador usado.

Para o algoritmo, cada ponto de discretização corresponderá a um ângulo  $\gamma$  entre a linha de trabalho da suspensão e a força normal (Figura 27). Desse modo, é possível estimar os efeitos da suspensão mediante um deslocamento  $h$ . O ponto de contato entre a roda e o terreno é representado pelo vetor  $X_{cp}$ , de acordo com o sistema de coordenadas globais. Entretanto, conhecer o  $X_{cp}'$  é fundamental para encontrar o deslocamento da suspensão.

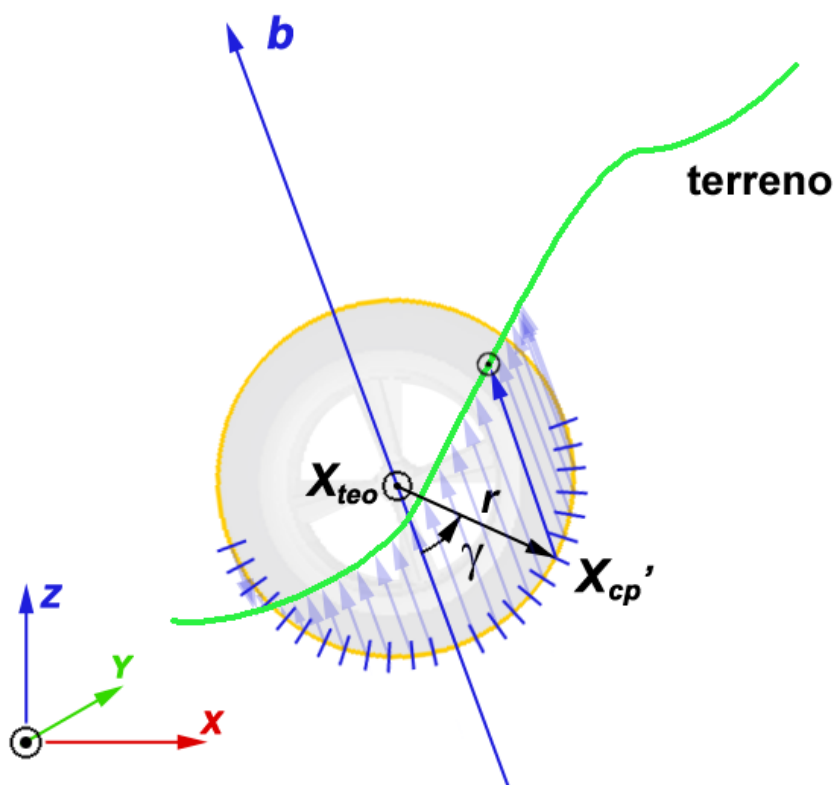


Figura 27 – Ilustração do esquema do cálculo de  $X_{cp}'$

Para encontrar  $\mathbf{X}_{cp}$  é preciso, em primeiro lugar, procurar dentre todos os pontos discretizados  $\mathbf{X}_{cp}$  aquele que gera o maior deslocamento de  $h$  (Figura 28).

$$\mathbf{X}_{cp}' = \mathbf{X}_{teo} + r \cdot \sin(\gamma) \cdot \mathbf{n} - r \cdot \cos(\gamma) \cdot \mathbf{b} \pm \frac{w}{2} \cdot \mathbf{t} \quad (3.7)$$

Nesse ponto, há o interesse em descobrir o ponto de interseção entre a roda e o terreno a partir do vetor  $X_{cp}'$  (Figura 28).

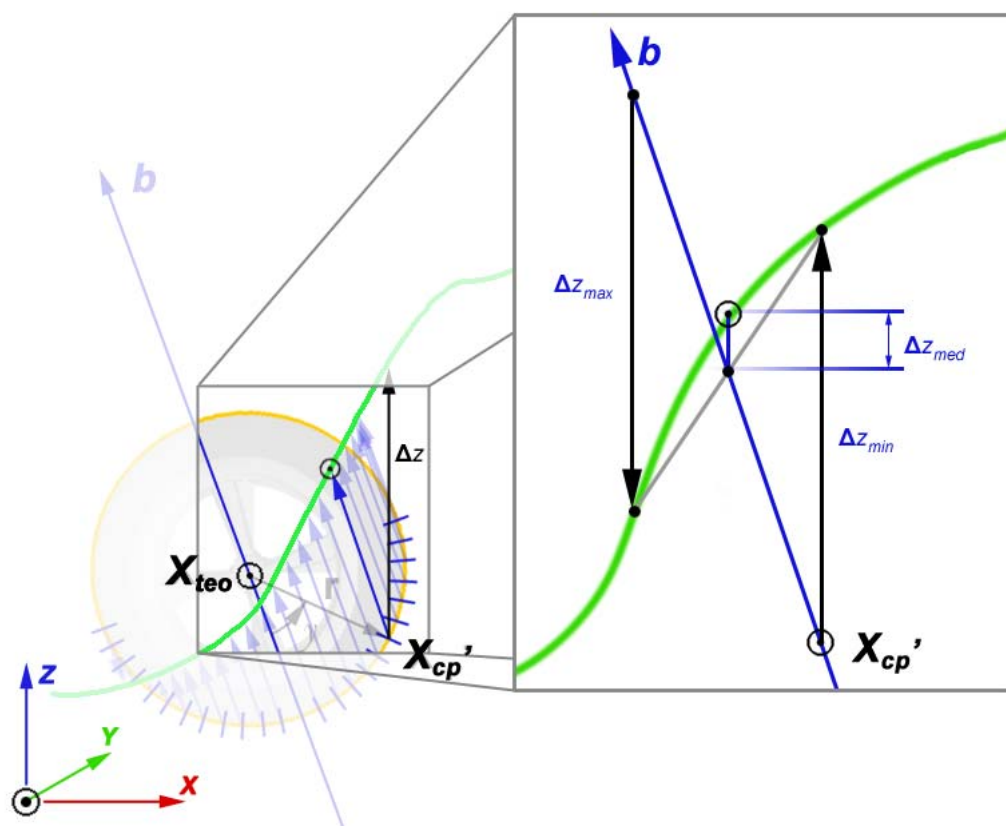


Figura 28 – Ilustração da convergência do ponto de contato da roda com o terreno

Com  $X_{cp}$ ’ calculado, procura-se então o valor de  $\Delta z_{med}$ , definido na figura, de modo a encontrar um deslocamento  $h$  maior ou igual a zero e menor ou igual ao deslocamento  $h_{sat}$ . Para a estimativa do deslocamento  $h$ , utiliza-se um método iterativo em que se procura um valor de  $h$  que satisfaça a equação  $Z_{cp} - f(X_{cp}, Y_{cp}) = 0$ . Para isso, inicialmente calcula-se, para um dado  $\gamma$ , a distância  $\Delta z$  entre a coordenada  $Z_{cp}$  do ponto discretizado da roda, para os valores

extremos admissíveis para  $h$ . Com a ponderação dos valores de  $\Delta z$  encontrados, estima-se  $h_{med}$ :

$$\begin{aligned} Z_{cp}(h_{min}) &= \mathbf{X}_{cp}' + h_{min} \cdot \mathbf{b}(z) \\ \Delta z_{min} &= Z_{cp}(h_{min}) - f(\mathbf{X}_{cp} + h_{min} \cdot \mathbf{b}(x), Y_{cp} + h_{min} \cdot \mathbf{b}(y)) \end{aligned} \quad (3.8)$$

$$\begin{aligned} Z_{cp}(h_{max}) &= \mathbf{X}_{cp}' + h_{max} \cdot \mathbf{b}(z) \\ \Delta z_{max} &= Z_{cp}(h_{max}) - f(\mathbf{X}_{cp} + h_{max} \cdot \mathbf{b}(x), Y_{cp} + h_{max} \cdot \mathbf{b}(y)) \end{aligned} \quad (3.9)$$

$$h_{med} = h_{min} + \frac{(h_{max} - h_{min})(-\Delta z_{min})}{(\Delta z_{max} - \Delta z_{min})} \quad (3.10)$$

$$\begin{aligned} Z_{cp}(h_{med}) &= \mathbf{X}_{cp}' + h_{med} \cdot \mathbf{b}(z) \\ \Delta z_{med} &= Z_{cp}(h_{med}) - f(\mathbf{X}_{cp} + h_{med} \cdot \mathbf{b}(x), Y_{cp} + h_{med} \cdot \mathbf{b}(y)) \end{aligned} \quad (3.11)$$

Determinadas as distâncias  $\Delta z_{min}$ ,  $\Delta z_{med}$  e  $\Delta z_{max}$ , verificam-se os seus sinais. Caso  $\Delta z_{max}$  seja positivo e  $\Delta z_{med}$  negativo, ou vice-versa, então o valor desejado encontra-se entre os valores de  $h_{max}$  e  $h_{med}$ : logo,  $h_{med}$  passará a ser o  $h_{min}$  da iteração seguinte. Para o caso em que  $\Delta z_{med}$  é negativo e  $\Delta z_{min}$  positivo, ou vice-versa, o  $h_{med}$  encontrado passará a ser o novo  $h_{max}$  da próxima interação. Se ambos  $\Delta z_{max}$  e  $\Delta z_{min}$  forem positivos, então a roda não faz contato com o terreno e, portanto,  $h = 0$ . Finalmente, se ambos forem negativos, então a suspensão está trabalhando na região não linear e, com isso, o  $h_{med}$  passa a ser o  $h_{sat}$ . O método iterativo termina quando a variação absoluta entre valores sucessivos de  $h_{med}$  é menor que um valor percentual de  $h_{sat}$  predeterminado ou quando  $h_{med}$  atinge um dos valores extremos 0 (zero) ou  $h_{sat}$ .

Tendo encontrado a melhor estimativa do deslocamento  $h$  de cada suspensão, calcula-se o centro de massa da roda e, em seguida, o seu ponto de contato com o terreno:



$$\mathbf{X}_{cm} = \mathbf{X}_{teo} + h \cdot \mathbf{b} \quad (3.12)$$

$$\mathbf{X}_{cp} = \mathbf{X}_{cm} + r \cdot \sin(\gamma) \cdot \mathbf{n} - r \cdot \cos(\gamma) \cdot \mathbf{b} \pm \frac{w}{2} \cdot \mathbf{t} \quad (3.13)$$

A Figura 29 mostra a posição real do centro de massa da roda, considerando o deslocamento da suspensão imposto pelo terreno, para todas as rodas do veículo robótico, além do ponto de contato da roda com o terreno, ambos segundo o referencial global.

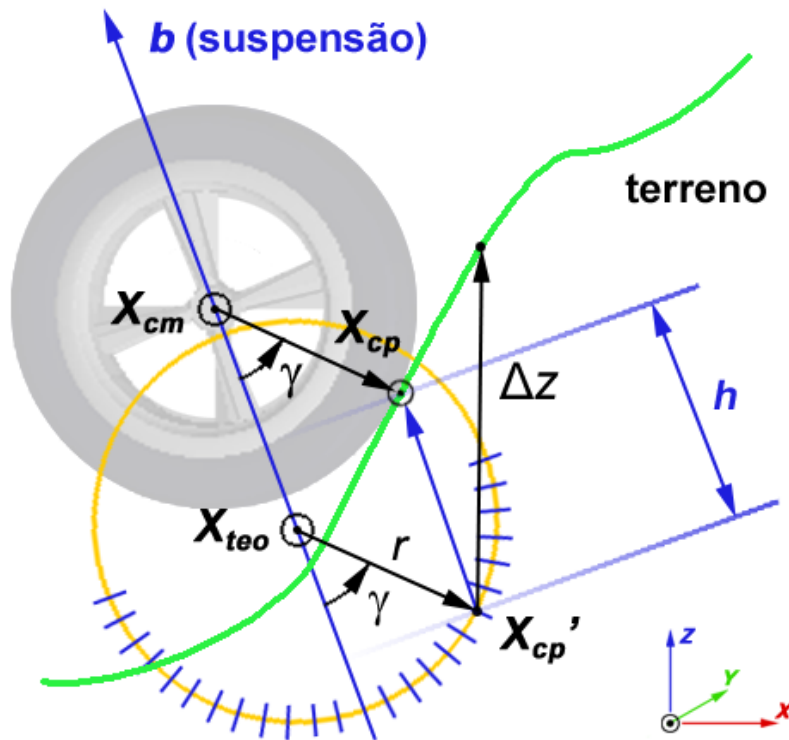


Figura 29 – Esquema do ponto de contato da roda com o terreno

### 3.4.

#### Modelagem Dinâmica do Veículo Robótico

O modelo proposto para o cálculo dos esforços que agem sobre o veículo robótico não leva em consideração o esterçamento das rodas, que normalmente

não existe na maioria dos veículos robóticos, pois eles têm o *tank steering* como diferencial de locomoção. Assim, todas as rodas sempre estarão paralelas entre si.

A partir do cálculo do ponto de contato  $X_{cp}$ , mostrado na Seção 3.3.4, determina-se o deslocamento imposto à suspensão e sua velocidade e, com isso, pode-se calcular a amplitude da força da suspensão  $F_{susp}$  – incluindo os efeitos de rigidez e amortecimento.

Pelo equilíbrio de forças, a amplitude da força de impulsão do veículo  $F_n$  e a amplitude da força normal  $N$  são obtidas em função de  $F_{susp}$  e  $F_x$  por meio de

$$F_n = \frac{F_x - F_{susp} \sin(\gamma)}{\cos(\gamma)} \quad (3.14)$$

$$N = \frac{F_{susp} - F_x \sin(\gamma)}{\cos(\gamma)} \quad (3.15)$$

lembrando que o  $\gamma$  é o ângulo entre  $N$  e  $\mathbf{b}$  (linha de trabalho da suspensão).

Assumindo que inicialmente a roda não esteja escorregando, a amplitude da força longitudinal  $F_x$  entre a roda e o terreno será dada pela relação entre o torque aplicado à roda e o seu raio. Logo, a princípio, tem-se que  $F_x = \tau/r$ , onde  $\tau$  é o torque aplicado à roda e  $r$ , o raio. No entanto, é preciso verificar se é possível aplicar  $F_x$ , pois se a roda estiver acima do limite de aderência do pneu, esse valor deve ser ajustado utilizando o coeficiente de atrito dinâmico.

A Figura 30 mostra os esforços agindo entre a roda e o terreno. Pode-se ver que se  $\cos(\gamma) \leq 0$  o ponto de contato entre a roda e o terreno encontra-se no hemisfério superior da roda em relação ao sistema do chassi. Nesse caso específico, por se tratar de uma colisão ou travamento do veículo, assume-se que  $F_n = 0$ , levando a  $F_x = F_{susp}$  e  $N = 0$ . Para  $\cos(\gamma) \geq 0$ , é necessário cogitar dois casos:

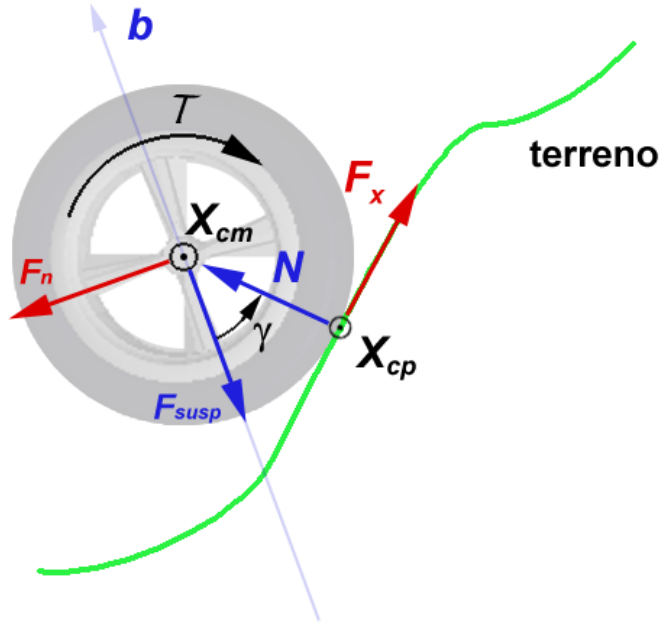


Figura 30 – Esforços agindo entre a roda e o terreno

1. Se  $F_{susp} \leq (F_x \sin(\gamma))$ , tem-se que, pela Equação (3.15),  $N \leq 0$ .

Como não é possível ter uma força normal negativa, assume-se que  $N = 0$ . Se a força normal é nula, a roda não faz contato com o terreno, logo a força de tração  $F_x$  também é nula e a suspensão encontra-se relaxada (pois desprezou-se a inércia de translação das rodas) e portanto  $F_{susp} = 0$ ;

2. Também pela Equação (3.15), se  $F_{susp} > (F_x \sin(\gamma))$  então  $N > 0$ .

Sabendo que  $-\mu N \leq F_x \leq \mu N$ , onde  $\mu$  é o coeficiente de atrito estático, a força de tração pode ser analisada de duas maneiras:

- a. Se  $F_x \geq 0$ , então  $F_x \leq \mu N$ . Da Equação (3.15) tem-se que

$$F_x \leq \left( \mu \frac{F_{susp}}{\cos(\gamma)} - \mu F_x \frac{\sin(\gamma)}{\cos(\gamma)} \right) \quad (3.16)$$

logo,

$$F_x (\cos(\gamma) + \mu \sin(\gamma)) \leq \mu F_{susp} \quad (3.17)$$

Se  $(\cos\gamma + \sin(\gamma)) > 0$ , então

$$F_x \leq \frac{\mu F_{susp}}{\cos(\gamma) + \mu \sin(\gamma)} \quad (3.18)$$

caso a inequação (3.18) não seja satisfeita, o valor de  $F_x$  deverá ser ajustado de acordo com

$$F_x = \frac{\mu^* F_{susp}}{\cos(\gamma) + \mu^* \sin(\gamma)} \quad (3.19)$$

onde  $\mu^*$  é o coeficiente de atrito dinâmico, pois a roda estaria escorregando. No entanto, se  $(\cos(\gamma) + \mu \sin(\gamma)) \leq 0$ , então o valor encontrado para  $F_x$  não precisa ser corrigido, pois o primeiro termo encontrado da inequação (3.17) será sempre negativo e menor que o segundo termo porque, nesse caso,  $F_{susp}$  é sempre positivo;

- b. Se  $F_x < 0$ , então  $-F_x \leq \mu N$ . Da mesma maneira descrita acima, da Equação (3.15) chega-se a

$$F_x (\cos(\gamma) - \mu \sin(\gamma)) \geq -\mu F_{susp} \quad (3.20)$$

Se  $(\cos(\gamma) - \sin(\gamma)) > 0$ , então

$$F_x \geq \frac{-\mu F_{susp}}{\cos(\gamma) - \mu \sin(\gamma)} \quad (3.21)$$

Caso a inequação (3.21) não seja satisfeita, o valor de  $F_x$  deverá ser ajustado de acordo com

$$F_x = \frac{-\mu^* F_{susp}}{\cos(\gamma) - \mu^* \sin(\gamma)} \quad (3.22)$$

No entanto, se  $(\cos(\gamma) - \mu \sin(\gamma)) \leq 0$ , então o valor encontrado para  $F_x$  não precisa ser corrigido, pois  $F_{susp} > 0$ .

Uma vez determinada a força de tração  $F_x$  admissível e a força da suspensão  $F_{susp}$ , a partir da deformação  $h$  e da taxa de deformação  $dh/dt$  da suspensão, calcula-se a força de tração  $F_n$  das rodas

$$F_n = \left( \frac{F_x - F_{susp} \sin(\gamma)}{\cos(\gamma)} \right) n \quad (3.23)$$

onde  $n$  é o vetor unitário que define a direção “para a frente” do referencial local do veículo. A força  $F_b$  (força que a suspensão exerce sobre o chassi) será o produto do módulo da força da suspensão  $F_{susp}$  por  $b$  (vetor unitário que define a direção “para cima” do referencial local do veículo), logo

$$F_b = F_{susp} b \quad (3.24)$$

Para a força lateral/transversal (na direção de  $t$ ), foi implementado o modelo de atrito simples proporcional à força normal de maneira que  $F_y = (\mu N)t$ .

No entanto, esse modelo de força transversal deixa o sistema próximo a uma instabilidade numérica, gerando oscilações devido ao fato de  $F_y$  nesse modelo poder assumir apenas três valores,  $-\mu N$ ,  $0$  ou  $\mu N$ , sem incluir os valores intermediários. Com isso, foi necessário usar um modelo mais sofisticado, descrito a seguir.

### 3.5.

#### Modelo da Força de Contato Pneu-Terreno

Como mencionado no item anterior, um modelo simplificado da força de contato pneu-terreno não é uma boa opção, e por isso foi implementado o modelo de interação intitulado “**Fórmula Mágica**” [53], que será detalhado a seguir.

#### 3.5.1.

##### Derivas Longitudinal e Lateral

Deriva é um fenômeno frequentemente usado como deslizamento em tração ou frenagem. Esse deslizamento é calculado a partir da velocidade angular da roda e a da velocidade linear do seu centro de massa, como pode ser visto nas equações 3.25 e 3.26. No modelo implementado foi necessário encontrar as derivas longitudinal e lateral para o cálculo das forças de interação.

Quando um torque propulsor é aplicado a uma roda de forma que gire a uma determinada velocidade sem patinar, diz-se que a velocidade angular e a translação do centro de massa da roda são proporcionais. Entretanto, quando essas grandezas são desproporcionais, diz-se que está ocorrendo um deslizamento ou deriva, que pode ser tanto de tração quanto de frenagem.

Para a deriva longitudinal, foram implementadas duas equações, uma para tração e outra para frenagem, medidas percentualmente.

Deriva longitudinal para tração:

$$\left( \frac{\omega r - V}{\omega r} \right) \cdot 100 \quad (3.25)$$

onde  $\omega$  é a velocidade angular,  $r$  é o raio e  $V$  é a velocidade linear do centro de massa da roda.

Deriva longitudinal para frenagem:

$$\left( \frac{V - \omega r}{V} \right) \cdot 100 \quad (3.26)$$

Para o cálculo da deriva lateral, usa-se a equação a seguir, em radianos:

$$\arctan\left(\frac{V_x}{V_y}\right) \quad (3.27)$$

onde  $V_x$  é a componente longitudinal e  $V_y$  a componente lateral, ambas pertencentes à velocidade linear do centro de massa da roda.

### 3.5.2.

#### Cálculo das Forças Longitudinal e Lateral

O modelo utilizado para o cálculo das forças da interação pneu-terreno é um método semiempírico chamado de “**Fórmula Mágica**” (“*Magic Formula*”, [53]). Pacejka [53] coloca que essa abordagem, por se tratar de uma curva ajustada de acordo com dados experimentais, é mais exato que outros métodos existentes. As equações envolvidas são descritas a seguir:

$$y = D \sin \left[ \text{Carctan} \left\{ Bx - E \left( Bx - \arctan Bx \right) \right\} \right] \quad (3.28)$$

$$Y(X) = y(x) + S_V \quad (3.29)$$

$$x = X + S_H \quad (3.30)$$

onde

$Y$  – variável de saída  $F_x$  e  $F_y$

$X$  – variável de entrada  $\tan(\alpha)$  ou  $\kappa$

e

$B$  é o fator de rigidez;

$C$  é o fator de forma;

$D$  é o valor de pico;

- $E$  é o fator de curvatura;
- $S_H$  é o offset horizontal;
- $S_V$  é o offset vertical;

A fórmula mágica  $y(x)$  produz uma curva que passa através da origem  $x = y = 0$ , obtendo um valor de pico  $x_m$ , onde  $y_a$  é o valor final para onde a assíntota converge. Na imagem à direita na Figura 31, vê-se como o argumento do  $\sin(\beta)$  varia de acordo com  $x$  e a distorção que isso causa no seno.

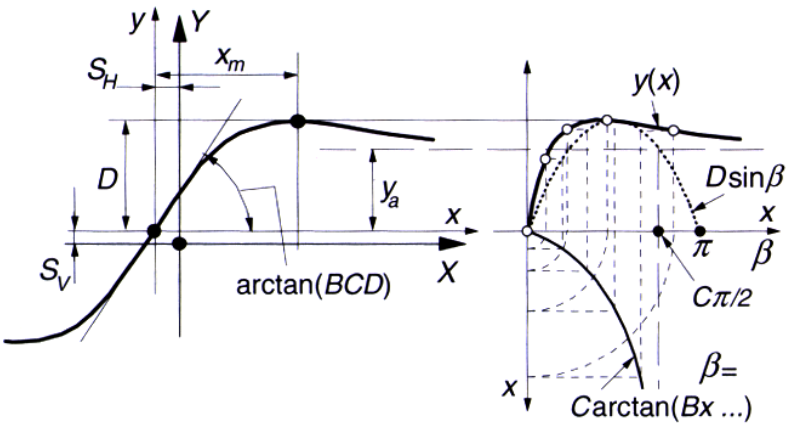


Figura 31 – Curva produzida pela versão original da “Fórmula Mágica” em [53]

Tabela 4 – Coeficientes dos pneus para a “Fórmula Mágica” em [54], valores em  $kN$

Valores dos coeficientes dos pneus de um veículo para a "Fórmula Mágica"								
	Carga, $F_z$ , kN	$B$	$C$	$D$	$E$	$S_h$	$S_v$	$BCD$
$F_y$ , N	2	0.244	1.50	1936	-0.132	-0.280	-118	780.6
	4	0.239	1.19	3650	-0.678	-0.049	-156	1038
	6	0.164	1.27	5237	-1.61	-0.126	-181	1091
	8	0.112	1.36	6677	-2.16	0.125	-240	1017
$M_z$ , N · m	2	0.247	2.56	-15.53	-3.92	-0.464	-12.5	-9.820
	4	0.234	2.68	-48.56	-0.46	-0.082	-11.7	-30.45
	6	0.164	2.46	-112.5	-2.04	-0.125	-6.00	-45.39
	8	0.127	2.41	-191.3	-3.21	-0.009	-4.22	-58.55
$F_x$ , N	2	0.178	1.55	2193	0.432	0.000	25.0	605.0
	4	0.171	1.69	4236	0.619	0.000	70.6	1224
	6	0.210	1.67	6090	0.686	0.000	80.1	2136
	8	0.214	1.78	7711	0.783	0.000	104	2937



Os valores da Tabela 4 foram obtidos a partir de um ajuste (*curve fitting*) de coeficientes com dados experimentais em função da força normal ( $N$ ) aplicada sobre o pneu. Eles já incorporam o coeficiente de atrito entre pneu e um tipo de terreno não especificado pelo autor [54].

Os valores de  $B$ ,  $C$ ,  $D$ ,  $E$ ,  $S_V$  e  $S_H$  dependem da força normal  $F_z$  e do ângulo de cambagem  $\gamma_c$  e devem ser estimados a partir de curvas experimentais do pneu. Porém, na presente dissertação,  $\gamma$  será sempre zero, pois as rodas são perpendiculares ao chassi. A dependência dos coeficientes com relação à força normal foi modelada (mediante ajuste de curvas) de acordo com as equações mostradas a seguir.

Tabela 5 – Coeficientes  $a_1$  até  $a_8$  para o pneu do veículo

Valores dos coeficientes $a_1$ até $a_8$ para o pneu do veículo ( $F_z$ em kN)								
	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$
$F_y$ , N	-22.1	1011	1078	1.82	0.208	0.000	-0.354	0.707
$M_z$ , N · m	-2.72	-2.28	-1.86	-2.73	0.110	-0.070	0.643	-4.04
$F_x$ , N	-21.3	1144	49.6	226	0.069	-0.006	0.056	0.486

Para o cálculo do valor de pico  $D$ , usa-se

$$D = a_1 F_z^2 + a_2 F_z \quad (3.31)$$

onde  $F_z$  está em  $kN$  e  $a_1$  e  $a_2$  são coeficientes empíricos apresentados na Tabela 5. Lembrando que o termo  $D$  é usado tanto para a força transversal ( $F_x$ ) como longitudinal ( $F_y$ ).

Para  $BCD$ , que representa as rigidezes, são apresentadas duas equações, uma para a força transversal e outra para a longitudinal.

Para  $F_x$  tem-se

$$BCD = \frac{a_3 F_z^2 + a_4 F_z}{e^{a_5 F_z}} \quad (3.32)$$

e, para  $F_y$ ,

$$BCD = a_3 \sin \left[ a_4 \arctan(a_5 F_z) \right] \quad (3.33)$$

onde  $a_3$ ,  $a_4$  e  $a_5$  são coeficientes empíricos que podem ser vistos na Tabela 5.

Para o termo  $E$ , o fator de curvatura, tem-se uma única equação tanto para a força longitudinal quanto para a transversal:

$$E = a_6 F_z^2 + a_7 F_z + a_8 \quad (3.34)$$

onde  $a_6$ ,  $a_7$  e  $a_8$  são coeficientes empíricos (ver Tabela 5).

Para o termo  $C$ , que representa o fator de forma, praticamente independente de  $F_z$ , usam-se as constantes a seguir para  $F_y$  e  $F_x$ :

para  $F_y$ ,

$$C = 1.19$$

e para  $F_x$ ,

$$C = 1.69$$

O termo  $B$ , o fator de rigidez, tanto para a força longitudinal quanto lateral, é obtido por

$$B = \frac{BDC}{C \cdot D} \quad (3.35)$$

Neste trabalho, para os termos  $S_H$  e  $S_V$ , que representam o offset horizontal e vertical, respectivamente, foi fixado um valor constante zero tanto para  $F_x$  quanto para  $F_y$ :

$$S_H = S_V = 0 \quad (3.36)$$

### 3.5.3.

#### Combinação das Derivas

Em [53], Pacejka descreve uma combinação eficiente das derivas longitudinal e lateral quando ambas atuam simultaneamente. Esse método foi desenvolvido por Michelin e publicado por Bayle, Forissier e Lafon (1993).

A próxima equação atenua a força longitudinal:

$$G_{xa} = \frac{\cos\left(C_{xa} \arctan\left(B_{xa} \cdot a_{Sa} - E_{xa}\left(B_{xa} \cdot a_{Sa} - \arctan(B_{xa} \cdot a_{Sa})\right)\right)\right)}{\cos\left(C_{xa} \arctan\left(B_{xa} \cdot S_{Ha} - E_{xa}\left(B_{xa} \cdot S_{Ha} - \arctan(B_{xa} \cdot S_{Ha})\right)\right)\right)} \quad (3.37)$$

enquanto a equação a seguir é responsável pela atenuação da força lateral:

$$G_{y\kappa} = \frac{\cos\left(C_{y\kappa} \arctan\left(B_{y\kappa} \cdot \kappa_S\right)\right)}{\cos\left(C_{y\kappa} \arctan\left(B_{y\kappa} \cdot S_{H\kappa}\right)\right)} \quad (3.38)$$

### 3.6.

#### Atuador

Os atuadores são dispositivos com a meta de converter energia elétrica, hidráulica ou pneumática, por exemplo, em energia mecânica, possibilitando, por meio dos sistemas de transmissão, o posicionamento e movimentação das partes móveis do robô. Os atuadores mais comuns em sistemas robóticos são os motores elétricos, que podem ser motores de corrente contínua (DC) com escova (*brushed*) ou sem escova (*brushless*), ou motores de corrente alternada (AC), normalmente usados em indústrias.

A presente dissertação fará uso apenas do motor **Magmotor S28-150**, um atuador motor de corrente contínua com escovas na modelagem do simulador VirtualBotz 3D. Para maiores detalhes de configurações de outras características de motores DC, vide o **Anexo J**.

Tabela 6 – Parâmetros do motor Magmotor [48] utilizado no robô VIVI

<b>Magmotor S28-150</b>	
<b>Parâmetro</b>	<b>Valor</b>
Tensão elétrica nominal (V)	24
Potência máxima de saída (W)	2,183
Massa (kg)	1,7
Potência / Peso (W/kg)	1,284
$K_t$ (N·m/A)	0,03757
$K_v$ (RPM/V)	254
$R_{\text{motor}}$ ( $\Omega$ )	0,064
$I_{\text{sem\_carga}}$ (A)	3,4

Para simular uma força de atrito nos eixos dos motores, foi implementada uma versão modificada do modelo de fricção de LuGre, originalmente proposto por Canudas de Wit *et al.* [61]. A esta versão é dado o nome de Aproximação Contínua do Modelo de Fricção de LuGre [55], na qual um termo descontínuo é aproximado por meio de uma função contínua. A força de atrito que age entre o movimento dos corpos é escrita da seguinte forma:

$$F_a = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{y} \quad (3.39)$$

Nela, têm-se os coeficientes de rigidez, amortecimento e viscosidade.

Em [55] é proposta a seguinte equação para  $\dot{z}$ :

$$\dot{z} = S_1(\dot{y})\dot{y} - \frac{S_2(\dot{y})}{f_s(\dot{y})}z \quad (3.40)$$

A próxima função descreve as características de viscosidade do atrito para velocidades constantes:

$$f_s(\dot{y}) = \frac{1}{\sigma_0} \left[ F_c + (F_s - F_c) e^{-(\dot{y}/\dot{y}_s)^2} \right] \quad (3.41)$$

onde  $F_c$  é a força de atrito de Coulomb e  $F_s$  é a força de atrito estático. Definem-se também

$$S_0(\dot{y}) = \frac{2}{\pi} \arctan(k_v \dot{y}) \quad (3.42)$$

$$S_1(\dot{y}) = (S_0(\dot{y}))^2 \quad (3.43)$$

$$S_2(\dot{y}) = m(\dot{y}) = \dot{y} S_0(\dot{y}) \quad (3.44)$$

onde  $k_v$  é uma constante positiva.

### 3.7.

#### Método de Integração

A principal vantagem de implementar o simulador em C++ não diz respeito apenas à modularização do sistema e consequente manutenção, aspecto que se refere à facilidade de modificar um sistema a fim de corrigir defeitos, adequá-lo a novos requisitos ou a um ambiente novo, ou aumentar a sua suportabilidade, mas também permite a otimização do código a fim tirar proveito de recursos computacionais, como a programação de processos, permitindo assim a execução em tempo real com um passo temporal de integração da ordem de 100ns.

Em vista da complexidade e arbitrariedade das inclinações do terreno, não foi possível usar métodos implícitos de integração como o Newmark-Beta, devido ao fato de o simulador não ajustar uma função ao perfil do terreno e, com isso, não ser possível calcular gradientes de terreno que seriam necessários em métodos implícitos. O simulador usa método explícito e daí a necessidade de um passo de integração pequeno.

Devido ao esforço aplicado na otimização do código para diminuir o passo de integração, o simulador hoje contempla a possibilidade de o veículo robótico ter quantas rodas o usuário definir, em qualquer lugar do chassi.

No próximo capítulo, os algoritmos de controle para o sistema aqui modelado são apresentados.