



Universidade de Aveiro
2009

Departamento de Electrónica
Telecomunicações e
Informática

JOÃO MIGUEL
RIBEIRO PAIVA FERRAZ

LABORATÓRIO VIRTUAL DE
SISTEMAS DE CONTROLO



Universidade de Aveiro
2009

Departamento de Electrónica
Telecomunicações e
Informática

JOÃO MIGUEL
RIBEIRO PAIVA FERRAZ

LABORATÓRIO VIRTUAL DE
SISTEMAS DE CONTROLO

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Dra. Petia Georgieva, Professora auxiliar do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro

O Júri

Presidente

Professora Doutora Ana Perfeito Tomé

Professora associada da Universidade de Aveiro

Vogal – Arguente Principal

Professor Doutor Fernando Gomes Martins

Professor associado da Faculdade de Engenharia da Universidade do Porto

Vogal – Orientador

Professor Doutora Petia Georgieva Georgieva

Professora associada da Universidade de Aveiro

agradecimentos

Queria agradecer pela ajuda na conclusão desta Dissertação de mestrado em Engenharia Electrónica e Telecomunicações:

À minha orientadora a Dra. Petia Georgieva, Profesora auxiliar no Departamento de Electrónica e Telecomunicações da Universidade de Aveiro, por se ter proposto a desenvolver esta dissertação comigo e que sempre demonstrou imensa disponibilidade quando necessitei.

Um excelente trabalho como orientadora.

Aos meus pais pelo suporte que sempre me deram e pela força que me transmitiram;

Aos meus irmãos pela alegria demonstrada;

À minha tia com quem vivi ao longo de todos estes anos e que sempre me apoiou;

A todos os meus amigos que foram sempre impecáveis e que sem eles, eu não teria conseguido;

A uma pessoa em especial que sempre me apoiou e sempre acreditou em mim.

palavras-chave

Laboratório virtual de controlo; Modelos virtuais com animação; Controlo de velocidade de um automóvel; Controlo de posição de um pêndulo invertido; Suspensão de um autocarro.

resumo

Esta dissertação tem como principal objectivo a criação de um laboratório virtual de sistemas de controlo no ambiente computacional Matlab utilizando a ferramenta SimMechanics do Simulink. Foram criados três modelos virtuais de objectos mecânicos de controlo e a sua respectiva animação on-line em SimMechanics: i) Modelo virtual de um automóvel; ii) Modelo virtual de um pêndulo invertido e iii) Modelo virtual do sistema de suspensão de um autocarro.

Para testar o funcionamento do laboratório virtual e os respectivos modelos físicos foram desenvolvidos vários esquemas de controlo implementando diferentes algoritmos de controlo clássico.

- O controlo da velocidade de um automóvel foi realizado com três controladores - PID, Atraso e Avanço.
- O controlo da posição do pêndulo invertido foi testado com dois controlador - PID e controlo por realimentação das variáveis do estado.
- O controlo da suspensão de um autocarro foi realizado com um controlador PID e um controlador por cancelamento de pólo zero.

Os controladores clássicos foram desenhados através das técnicas de controlo de sistemas lineares. Para tal, primeiro foram determinados os modelos lineares dos três casos mecânicos e depois os parâmetros dos controladores foram ajustados baseado nestes modelos lineares.

Para cada objecto foram testados duas malhas fechadas de controlo: com o modelo linear e com o modelo virtual. Os resultados experimentais verificaram que os modelos virtuais desenvolvidos representam bem a dinâmica dos três objectos físicos e que o laboratório pode ser usado, nos próximos anos, pelos alunos das cadeiras de Sistemas e Controlo I e II.

keywords

Virtual Control Laboratory; Animated virtual models; Speed control of an automobile; Position control of an inverted pendulum; Bus suspension control.

abstract

This Master Thesis has as main purpose the creation of a virtual laboratory of control systems in the Matlab computational environment using the SimMechanics toolbox of Simulink. Three virtual models of controlling mechanical objects and their respective on-line animation in SimMechanics were created: i) Virtual Model of an automobile; ii) Virtual Model of an Inverted pendulum and iii) Virtual Model of the system of a bus suspension.

To test the virtual laboratory and his respective physical models several control schematics have been developed which implemented different classical control algorithms.

- The control of the velocity of an automobile was tested with three controllers: PID, advanced phase and delaying phase controller.
- The control of the position of the inverted pendulum was tested with two controllers: PID and feedback of state space variables.
- The bus suspension controller was achieved with a PID controller and a root locus design.

The classic controllers were designed through control techniques of linear systems. First of all the linear models for the three cases of study were determined and then the parameters of the controllers were adjusted based on these linear models.

For each object were tested two close loop control designs: with the linear model and with the virtual model. The experimental results have checked that the developed virtual models represent the dynamics of the three objects and that the laboratory might as well be used for the next upcoming years by the Control Systems I and II students.

Índice

1	Introdução.....	9
2	Objectivos.....	9
3	Estado da Arte.....	9
3.1	Laboratórios virtuais de sistemas de controlo.....	9
3.1.1	VCLab (Virtual Control Laboratory).....	9
3.1.2	VL –CSD (Virtual Lab – Control System Design).....	11
4	Descrição da ferramenta de trabalho.....	13
4.1	Introdução ao ambiente de trabalho.....	13
4.1.1	Introdução.....	13
4.2	Introdução aos blocos mais significativos.....	15
4.2.1	Blocos do menu “Bodies”.....	15
4.2.2	Blocos do menu “Force Elements”.....	19
4.2.3	Blocos do menu “Joints”.....	21
4.2.4	Blocos do menu “Sensors & Actuators”.....	23
5	Casos de estudo.....	28
5.1	Controlo da velocidade de um automóvel.....	28
5.1.1	Introdução.....	28
5.1.2	Modelo virtual do automóvel.....	28
5.1.3	Especificações da dinâmica do automóvel.....	35
5.1.4	Esquema do controlo do automóvel com modelo linear e controlador PID.....	36
5.1.5	Esquema do controlo do automóvel com modelo linear e controladores Atraso/Avanço.....	37
5.1.6	Simulação de controlo do automóvel no Laboratório Virtual.....	39
5.1.7	Simulação de controlo do automóvel com alteração da referência.....	41
5.2	Controlo da posição do pêndulo invertido.....	43
5.2.1	Introdução.....	43
5.2.2	Modelo do Pêndulo Invertido.....	44
5.2.3	Especificações da dinâmica do Pêndulo Invertido.....	51
5.2.4	Problemas na visualização da animação do modelo do Pêndulo Invertido.....	53

5.2.5	Controlo do Pêndulo Invertido com modelo linear e controlador PID	54
5.2.6	Controlo do Pêndulo Invertido no Laboratório Virtual	55
5.3	Controlo da suspensão de um autocarro	61
5.3.1	Introdução.....	61
5.3.2	Modelo virtual da suspensão do autocarro	62
5.3.3	Especificações da dinâmica da suspensão de um autocarro	70
5.3.4	Modelo virtual da suspensão do autocarro na presença de uma perturbação sem controlador.....	71
5.3.5	Esquema do controlo do modelo virtual da suspensão do autocarro	73
6	Virtual Reality Toolbox	75
6.1	Introdução.....	75
6.2	Objectivos.....	76
6.3	VRealm Builder 2.0.....	76
6.3.1	Introdução ao editor VRealm Builder 2.0	76
6.4	Modelo virtual do automóvel	79
6.4.1	Ambiente virtual do automóvel criado em VRealm Builder	79
6.4.2	Ligação do modelo criado em VRealm com o modelo em SimMechanics	80
6.5	Modelo virtual do pêndulo invertido.....	83
6.5.1	Ambiente virtual do pêndulo invertido criado em VRealm Builder	83
6.5.2	Ligação do modelo criado em VRealm com o modelo em SimMechanics	84
6.6	Modelo virtual da suspensão do autocarro	86
6.6.1	Ambiente virtual da suspensão do autocarro criado em VRealm Builder	86
6.6.2	Ligação do modelo criado em VRealm com o modelo em SimMechanics	87
6.7	Conclusões.....	89
7	Bibliografia.....	90
8	Anexos.....	92
8.1	Anexo 1	92
8.2	Anexo 2	94
8.3	Anexo 3	97

1 Introdução

Actualmente podem encontrar-se alguns laboratórios virtuais de sistemas de controlo, embora a maior parte não possua a versatilidade que se pretende implementar ou então têm bastantes problemas inerentes devido a incompatibilidades entre o sistema do utilizador e as ferramentas de trabalho.

Com este trabalho pretende-se que um utilizador do software Matlab® possa usufruir de alguns modelos clássicos de controlo para posterior estudo e análise.

2 Objectivos

Com esta tese pretende-se criar um Laboratório Virtual de Sistemas de Controlo através do ambiente de Matlab® e das suas toolboxes.

Este laboratório irá ser usado nos anos subsequentes pelos alunos de Sistemas de Controlo I e II ou outros que tenham interesse na área de Controlo.

Vão criar-se simulações para algoritmos clássicos de controlo, tais como controladores PID, compensadores atraso/avanço, actuando num conjunto de plantas básicas académicas assim como o controlo da velocidade de um automóvel, suspensão de um autocarro e o pêndulo invertido.

3 Estado da Arte

3.1 Laboratórios virtuais de sistemas de controlo

3.1.1 VCLab (Virtual Control Laboratory)

Um Laboratório Virtual de Sistemas de Controlo pode ser encontrado em [20].

Este dá-se pelo nome de Virtual Control Lab (VLC) e é bastante parecido com o laboratório que se pretende implementar. Os casos de estudo disponíveis neste site são o modelo da bola-haste, os três tanques e o pêndulo com giroscópio. Todas as imagens foram retiradas do site referido.

Modelo virtual da bola e da haste:

Figura 3-1

Com este modelo pretende-se que uma esfera corra ao longo da haste. O controlo a ser efectuado será actuando na haste fazendo com que esta balance de modo a compensar o deslocamento da esfera. O objectivo final será estabilizar a esfera no centro da haste.

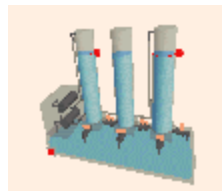
Modelo virtual dos três tanques:

Figura 3-2

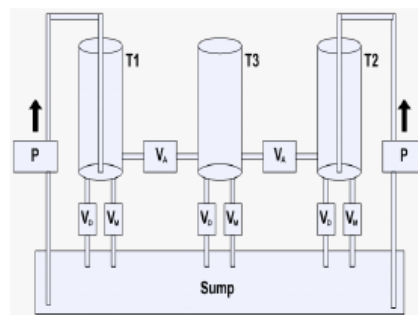


Figura 3-3 – Esquema do modelo dos três tanques

Esta simulação virtual consiste num conjunto de três tubos (tanques) com ligações entre eles e com um conjunto de válvulas (V_D, V_M) ligadas a um depósito. Os tanques T1 e T2 (ver Figura 3-3) possuem também duas bombas (P) que transferem a água do depósito para dentro destes. O objectivo neste caso será controlar o nível da água nos três tanques actuando nas válvulas e/ou nas bombas conforme o pretendido.

Modelo virtual do pêndulo com giroscópio:

Figura 3-4

Neste modelo tem-se um conjunto de um pêndulo com um giroscópio na extremidade. Note-se que o eixo de rotação do giroscópio é perpendicular ao eixo de rotação do pêndulo. Assim, a força de rotação do giroscópio, vai provocar um deslocamento do pêndulo da sua posição original. O objectivo deste modelo virtual é actuar na velocidade do giroscópio para manter o pêndulo numa dada posição do seu eixo de rotação.

O grande problema deste laboratório é o facto de ter demasiadas incompatibilidades entre o sistema operativo e/ou de navegação na Internet do utilizador e o próprio laboratório, visto que este encontra-se já bastante desactualizado.

As incompatibilidades começam por resultar do facto de ser necessária uma versão bastante antiga do Matlab® e só pode ser acedido usando o Netscape®. Para além disso, é necessário um programa que possui incompatibilidades com o Vista e o XP, garantido apenas o funcionamento para Windows 98.

3.1.2 VL –CSD (Virtual Lab – Control System Design)

Outro laboratório virtual e já mais actualizado é o VL-CSD ([21]).

Neste caso verifica-se que o laboratório é dividido em módulos podendo ser adquiridos desde que se proceda ao pagamento para se poder trabalhar com o mesmo. Nem todos os módulos são de teoria de controlo de sistemas. Os que o são e que vão abordar temas semelhantes aos desta dissertação, resumem-se a um Servomecanismo Electromecânico, dois tanques de água acoplados, uma Válvula contínua e Unidade laminadora a frio. Todas as imagens foram retiradas do site referido.

Servomecanismo Electromecânico:

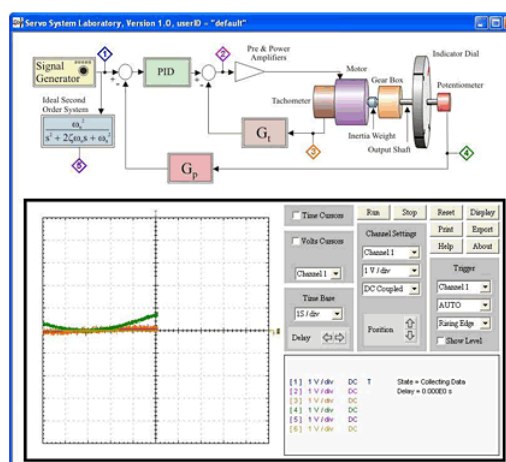


Figura 3-5

O Servomecanismo Electromecânico pretende simular um Servomecanismo DC comum que se pode encontrar em muitas Universidades, sendo que a própria simulação se pretende aproximar de um Servomecanismo que se pode encontrar na Universidade de Newcastle, Austrália. Um Servomecanismo deste género serve para carregamento de cargas por exemplo. Como núcleo deste sistema podemos encontrar um motor DC conectado, normalmente, a uma caixa de mudanças. Neste laboratório virtual pretende-se desenhar controlador(es) para o ângulo do motor.

Dois tanques de água acoplados:

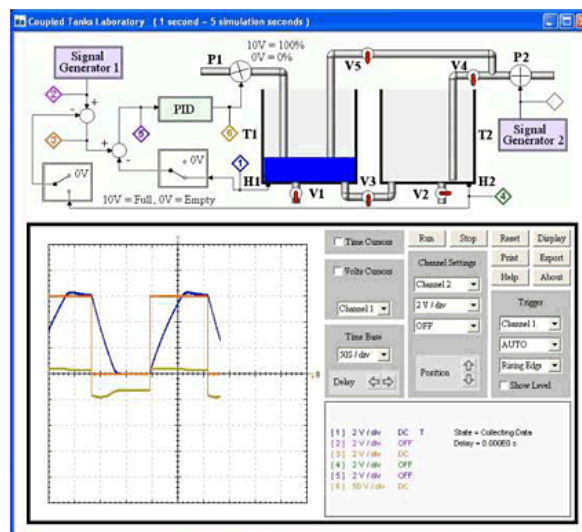


Figura 3-6

No laboratório de dois tanques de água acoplados, como o próprio nome indica, são dois tanques de armazenamento de água que se encontram ligados por um tubo. Mais uma vez pretende-se desenhar controlador(es) para, controlar o caudal de fluido à saída dos dois tanques.

Válvula Contínua:

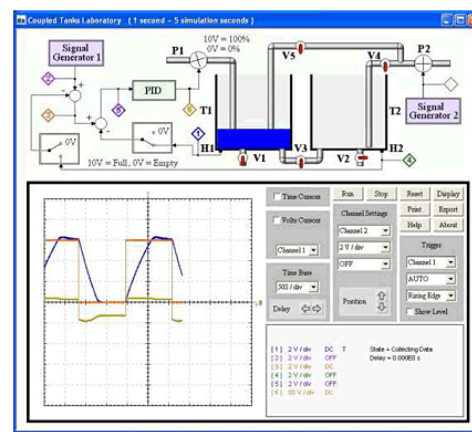


Figura 3-7

No caso da Válvula Contínua (meio bastante comum para solidificar metal líquido) pretende-se regular a válvula que conduz o metal ao molde e a que o faz sair deste ainda sob uma forma semi-sólida. O metal já tem de estar com a forma desejada mas suficientemente líquido para escorrer do molde permitindo a entrada de mais metal. O objectivo nesta simulação será a criação de controladores específicos de P, I e PI para o controlo da válvula que liga o metal líquido ao molde. Como se deve compreender este controlo tem de ser extremamente preciso visto que a temperatura é crítica no ponto semi-líquido desejado.

Máquina laminadora a frio:

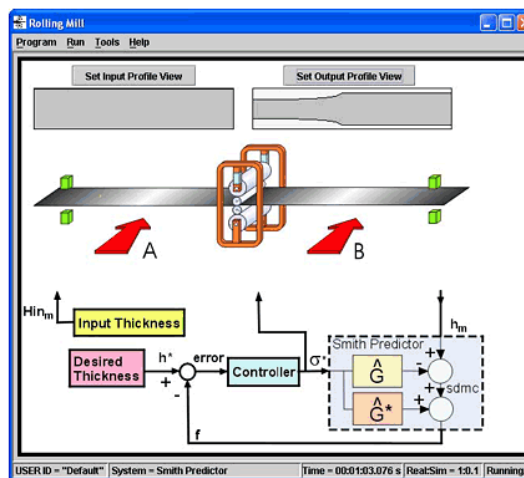


Figura 3-8

O laboratório virtual da máquina laminadora a frio pretende simular uma grande evolução tecnológica na sociedade actual, cujo objectivo é mudar ou reduzir a espessura (e assim as propriedades metalúrgicas) duma fita de metal. Esta terá de passar entre dois pares de rolos de cilindros como se pode verificar na Figura 3-8. Neste ambiente simulado tem-se uma introdução a transformadas de Laplace e a controladores PID.

Este conjunto de módulos constitui um laboratório bastante interessante (no entanto apenas foi verificado pelos exemplos fornecidos pelo site). O inconveniente destes módulos será o facto de se ter de pagar para usufruir dos mesmos.

4 Descrição da ferramenta de trabalho

4.1 Introdução ao ambiente de trabalho

4.1.1 Introdução

A ferramenta usada para desenvolvimento dos modelos referidos é o Matlab® e as toolboxes fornecidas. A versão de trabalho usada foi o Matlab® R2008b (versão 7.7.0.471). Usando o Simulink e as suas livrarias podem construir-se modelos dos mais variados tipos, incluindo modelos eléctricos, hidráulicos, mecânicos, etc.

Nesta dissertação a toolbox mais importante será o *SimMechanics* (uma toolbox parte da livraria *Simscape*) visto ter-se como objectivo a representação de vários sistemas mecânicos que envolvam plantas académicas de Sistemas de Controlo I/II.

Os modelos realizados possuem blocos do *SimMechanics* mas incluem também blocos do Simulink.

Qualquer questão que surja acerca dos blocos mencionados deverá poder ser resolvida recorrendo a [18] ou introduzindo o comando ">help 'nome do bloco' " na linha de comandos do Matlab.

Nota: AVS decorre da abreviatura de Ambiente Virtual de Simulação

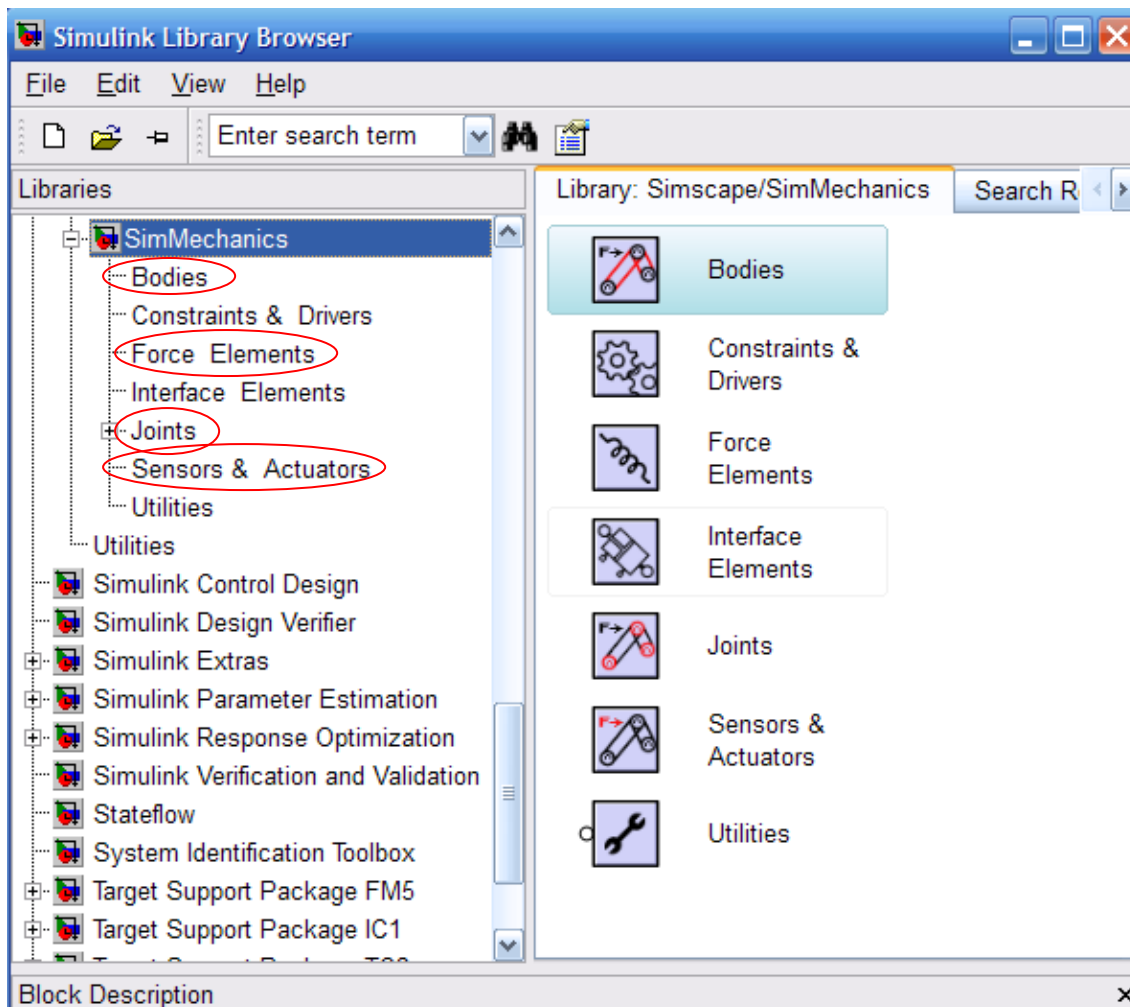


Figura 4-1

4.2 Introdução aos blocos mais significativos

4.2.1 Blocos do menu “Bodies”

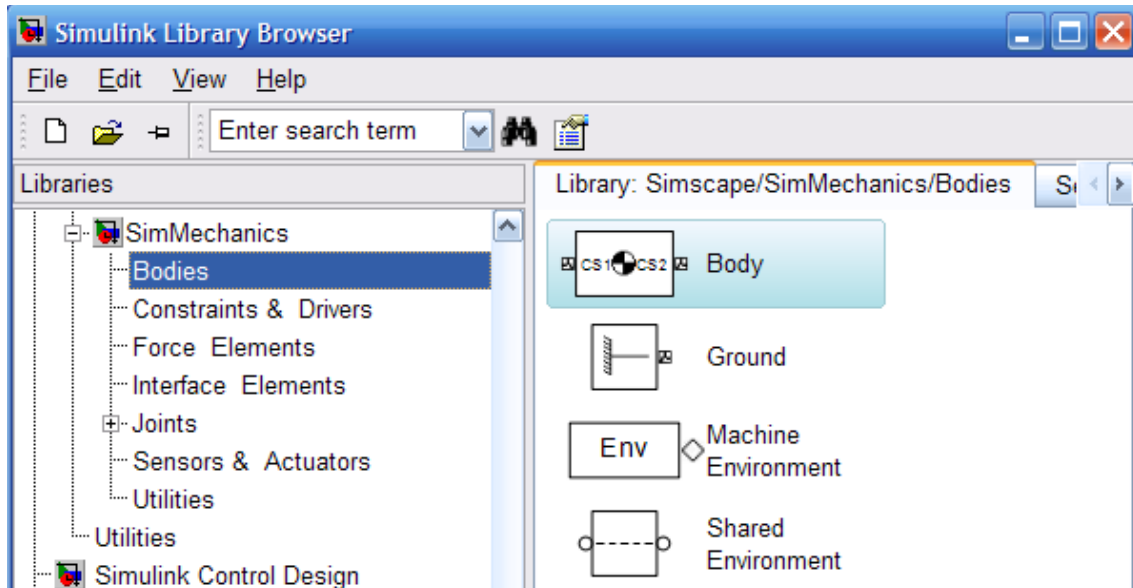


Figura 4-2

Indica-se, seguidamente, a descrição dos principais blocos do menu “Bodies” da ferramenta “SimMechanics”.

Bloco “BODY”

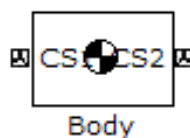


Figura 4-3

Tendo em vista todos os projectos realizados, o bloco essencial desta toolbox seria o *Body* (corpo) na Figura 4-3. Este é o bloco que representa as formas com massa. Pode ser ligado a blocos como o *Body Actuator* ou *Body Sensor* (que se encontram noutra separador diferente e permitem actuar ou medir valores do corpo). No entanto, a maior parte das vezes, os blocos *Body* encontram-se ligados a blocos de juntas. Estas juntas vão permitir movimento de acordo com os graus de liberdade específicos de cada junta.

Quando se procede à visualização na máquina virtual são os blocos *Body* que vão aparecer na animação. Dependendo do tipo de visualização que se pretende podem

observar-se estes blocos como: *convex hulls* (formas predefinidas que são sempre semelhantes); dependendo dos seus vectores de Inércia (em que adquirem várias formas como cones, esferas,...); dependendo dos seus pontos de ligação ao AVS. Estes pontos de conexão ao AVS, assim como o centro de massa do corpo, têm de ter as suas coordenadas para a posição inicial definidas.

O bloco *Body* tem como variáveis a Massa (M), a Inércia (I) e todos os pontos (CG, CS1, CS2,...) necessários para a posição inicial do bloco no AVS.

Como se pode verificar na Figura 4-4, as coordenadas dos pontos de conexão do bloco *Body*, para facilitar, podem ser definidas segundo um vector deslocamento em relação: à origem do AVS em si, ou seja, ao vector [0 0 0] (“Translated from origin of World”); ao centro de massa do corpo (“Translated from origin of CG”); umas às outras (“Translated from origin of CS1, CS2,...”); ao bloco adjacente (“Translated from origin of Adjoining”).

Definiu-se no modelo como variáveis a, b e c correspondentes respectivamente ao comprimento, largura e altura, do corpo.

Por exemplo o ponto CS5 encontra-se deslocado em relação à origem do AVS ([0 0 0]) com o vector [a 0 0] ou seja, tem deslocamento linear no valor de “a” unicamente no eixo do x. Mas, por exemplo, CS7 já se encontra no ponto [a 0 c] em relação à origem do AVS (isto porque se encontra deslocado de [0 0 c] de CS5, e este já estava deslocado de [a 0 0] da origem do AVS).

É preciso ter em conta que cada bloco *Body* não pode apresentar os pontos com uma relação cíclica. Por exemplo, se um bloco *Body* tem apenas três portas CG, CS1 e CS2, e se este tiver CS1 deslocado de [0 0 0] de CS2, CS2 deslocado de [1 1 1] de CG e CG deslocado de [-1 -1 -1] de CS1, obteve-se uma relação cíclica que não permite ao simulador obter uma posição inicial em relação ao AVS.

Em resumo, os pontos (CS6, CS5, CS4, CS3, e CG) têm um deslocamento em relação à origem do AVS ([0 0 0]), os pontos (CS8, CS7, CS10, CS9) têm deslocamento relativamente a outros pontos dos mesmo bloco (CS6, CS5, CS4, CS3) e o ponto CS1 está deslocado de [0 0 0] do bloco adjacente (“Adjoining”) que pode ser unicamente um bloco *Body* ou um bloco *Ground*.

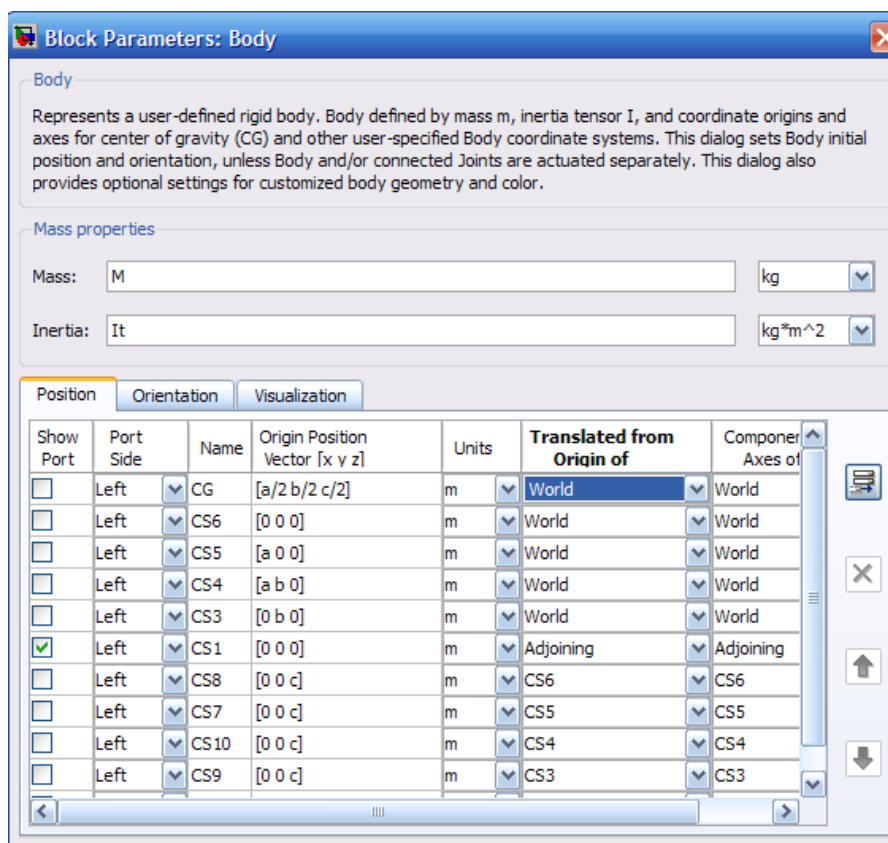


Figura 4-4

Bloco "GROUND"

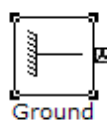


Figura 4-5

O bloco *Ground* (Figura 4-5) fixa elementos do objecto modelado ao AVS. É necessário sempre pelo menos um bloco *Ground* em cada projecto e, pelo menos um dos blocos de *Ground* do projecto, tem que estar conectado a um bloco *Machine Environment* (este bloco, também pertencente ao menu *Bodies*, será apresentado a seguir). Quando se quiser adicionar um bloco *Machine Environment* a um bloco *Ground*, é necessário activar a checkbox "Show Machine Environment Port", como se pode observar na Figura 4-6.

O bloco *Ground* precisa da localização relativa dentro do AVS definida, como se pode observar na Figura 4-6, em que, neste caso, se encontra na origem do AVS ([0 0 0]). Esta localização pode ser alterada quando se pretender e/ou quando a situação assim o exigir, bastando, para isso, alterar o vector deslocamento para a posição que se pretende dentro do AVS.

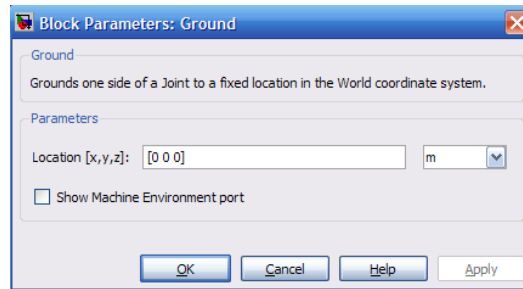


Figura 4-6

Bloco “MACHINE ENVIRONMENT”



Figura 4-7

O bloco *Machine Environment* (para ambiente de visualização da animação), cujo ícone se encontra na Figura 4-7, é necessário sempre que se pretende obter uma simulação virtual animada do objecto modelado. Este bloco terá de ser acoplado a um bloco *Ground* dentro do projecto do qual se pretende visualizar o objecto animado. O bloco *Machine Environment* serve para mudar as configurações do ambiente de visualização, para definir a orientação da gravidade segundo um dos eixos ou activar a opção de definir a gravidade como um sinal de entrada (ver Figura 4-8). Os “Configuration Parameters” de maior relevância são a alteração da forma geométrica dos corpos e a alteração dos valores temporais da simulação.

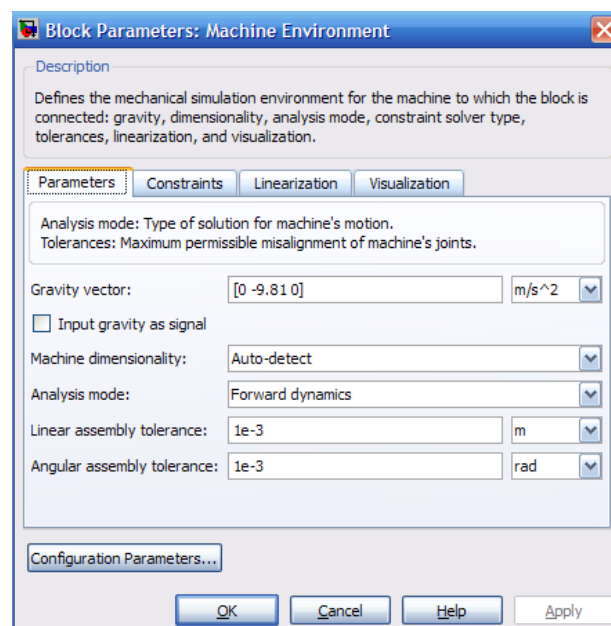


Figura 4-8

4.2.2 Blocos do menu “Force Elements”

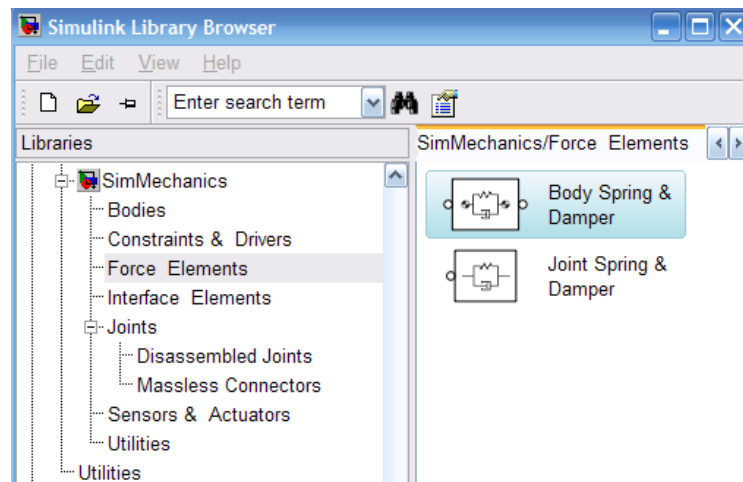


Figura 4-9

No menu *Force Elements* (Figura 4-9) encontram-se os blocos que definem e simulam os movimentos de molas e amortecedores. Estes blocos funcionam como actuadores em blocos *Body* ou blocos *Joint*. Existem dois tipos de blocos: um para ligar entre dois blocos *Body* e outro para ligar a um bloco *Joint* (a referir noutra capítulo desta Dissertação) e efectuar actuação com movimento oscilatório nessa junta.

Bloco “BODY SPRING & DAMPER”

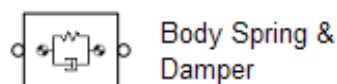


Figura 4-10

O bloco *Body Spring & Damper* (ver Figura 4-10) tem de estar conectado a dois blocos *Body*. Simula o grau de elasticidade e amortecimento entre dois blocos *Body* a que se encontra conectado e segundo o eixo de orientação destes corpos. Estas características são equivalentes à existência de uma mola com constante de elasticidade (k), constante de amortecimento (b) e comprimento natural (r_0). Estes parâmetros são definidos na janela da Figura 4-11.

O efeito deste bloco é a introdução de uma força de reacção que actua entre os dois corpos ligados e é proporcional ao deslocamento (r) e da velocidade (v) da mola segundo a fórmula $F = -k(r - r_0) - b \cdot v$.

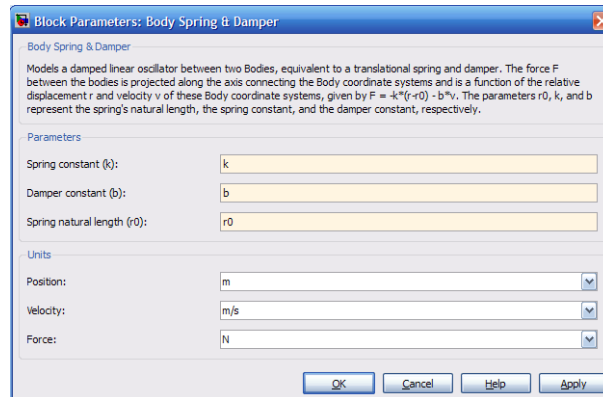


Figura 4-11

JOINT SPRING & DAMPER

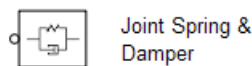


Figura 4-12

O bloco *Joint Spring & Damper* (Figura 4-12) permite simular a existência de uma mola (semelhante ao bloco *Body Spring & Damper*). O bloco *Joint Spring & Damper* actua segundo uma força de reacção semelhante ao bloco *Body Spring & Damper* mas num bloco *Joint* (a ser referido à frente neste capítulo). Assim, o movimento proporcionado pelo(s) grau(s) de liberdade do bloco *Joint* (podendo ser movimento de translação, e/ou de rotação), vai estar sujeito às forças de uma mola (constante de atenuação e de amortização, ...). Este bloco pode servir para actuar em mais que um bloco *Joint*. As constantes que definem este bloco (podem ser verificadas na Figura 4-13) são as constantes de elasticidade da mola (k), constante de amortecimento (b) e o offset da mola (x_0). Se se pretender que a mola esteja na sua posição original, para o momento inicial, o offset será nulo.

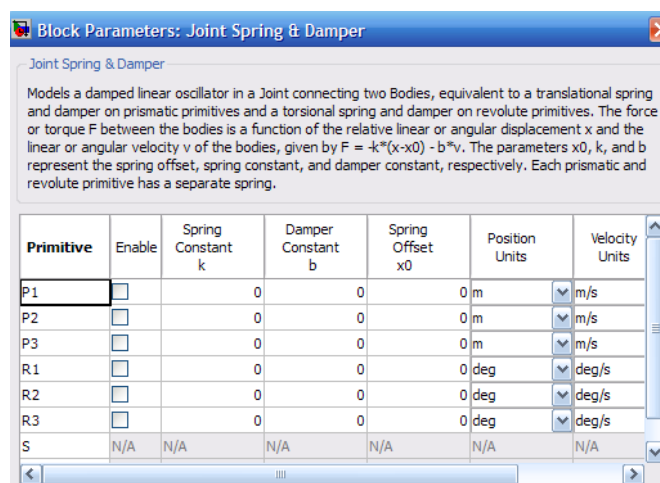


Figura 4-13

4.2.3 Blocos do menu “Joints”

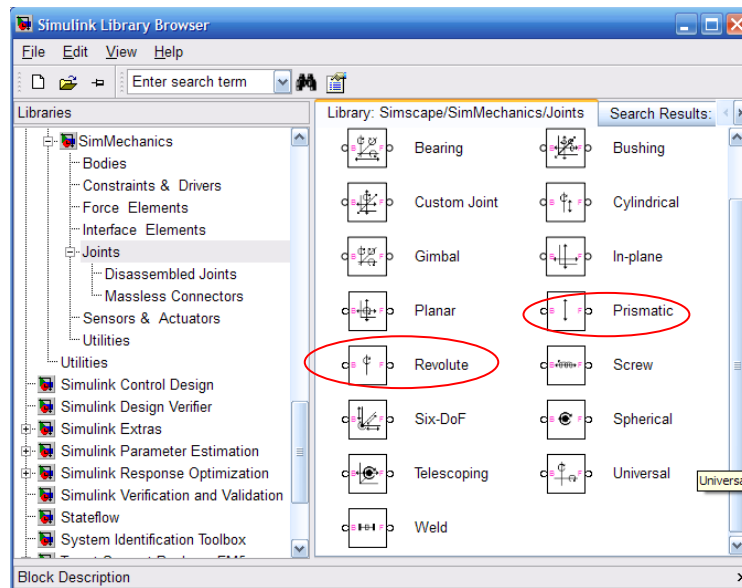


Figura 4-14

Noutro separador em *SimMechanics* pode-se encontrar o menu *Joints* (juntas). Aqui encontram-se todas as juntas necessárias para o movimento e articulação da máquina. A diferença entre estes blocos tem a ver com a permissão em termos de graus de liberdade.

Bloco “PRISMATIC”

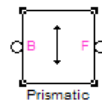


Figura 4-15

As juntas do tipo *Prismatic* (Figura 4-15) são conectadas, normalmente, entre blocos *Body* e/ou *Ground*. Esta junta tem apenas um grau de liberdade que é um movimento de translação segundo o eixo em que se deseja o movimento. Este eixo pode ser alterado nas propriedades do bloco. Assim, pode-se ter movimento translacional entre dois blocos segundo o eixo dos x, y ou z (ver Figura 4-16). Nas juntas também se podem ligar sensores e actuadores (*Joint Sensor* e *Joint Actuator* que serão explicados noutra capítulo). Na Figura 4-16 pode-se verificar a existência de movimento de translação segundo o eixo dos z. Se o vector colocado fosse $[0 \ 0 \ -1]$, por exemplo, implicaria uma alteração no sentido do movimento (em vez do sentido positivo dos z seria no sentido negativo).

Uma junta apenas pode ter um bloco *Joint Actuator* (o bloco *Joint Spring & Damper* referido anteriormente é também um actuador), por outro lado uma junta pode estar conectada a múltiplos blocos *Joint Sensor*. Outro bloco que pode ser ligado a um *Joint* é o *Joint Initial Condition* que define a posição e velocidade inicial da junta (será explicado noutro capítulo).

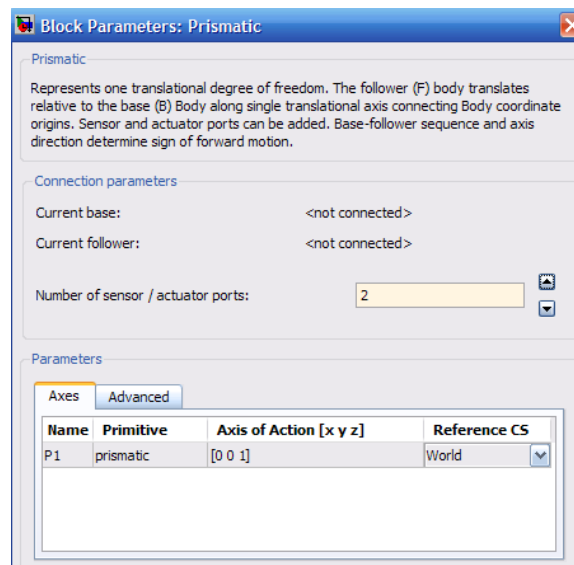


Figura 4-16

Bloco “REVOLUTE”

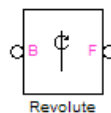


Figura 4-17

O bloco *Revolute* (Figura 4-17), tem as características semelhantes ao bloco *Prismatic*. A diferença é que, apesar de também autorizar apenas um grau de liberdade, permite obter um movimento rotacional na direcção de um eixo para o outro. Permite movimentos de 360° entre dois blocos. O eixo de acção é definido nas propriedades do bloco. Quando, por exemplo, se pretende movimento no sentido dos ponteiros do relógio (de z para y no sistema tradicional de três eixos) o vector a colocar nas propriedades será [1 0 0] (ver Figura 4-18). Também para este bloco, a alteração do sinal do vector de deslocamento implicaria apenas alteração no sentido do movimento.

Como referido para o bloco *Prismatic*, também neste bloco se pode colocar o número que se desejar de portas para actuadores e/ou sensores (mas, também neste, apenas se pode adicionar um actuador).

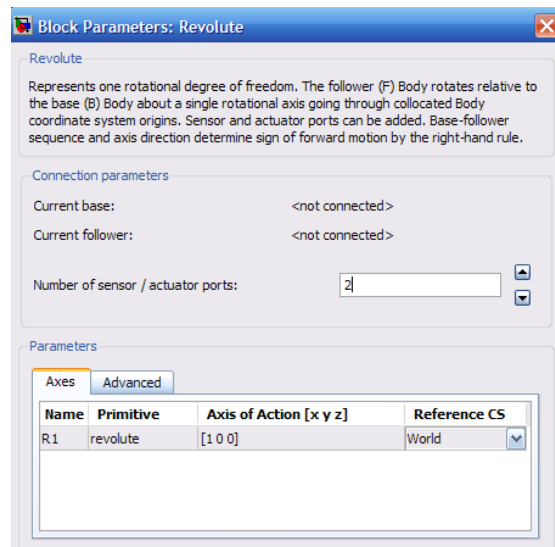


Figura 4-18

O resto das juntas presentes neste separador têm todas propriedades idênticas, apenas variando o número de graus de liberdade que estas permitem. Existem várias combinações de juntas em blocos pré-definidos que, por exemplo, se for uma combinação de juntas do tipo *Prismatic* e *Revolute*, dá origem a uma junta cilíndrica.

4.2.4 Blocos do menu “Sensors & Actuators”

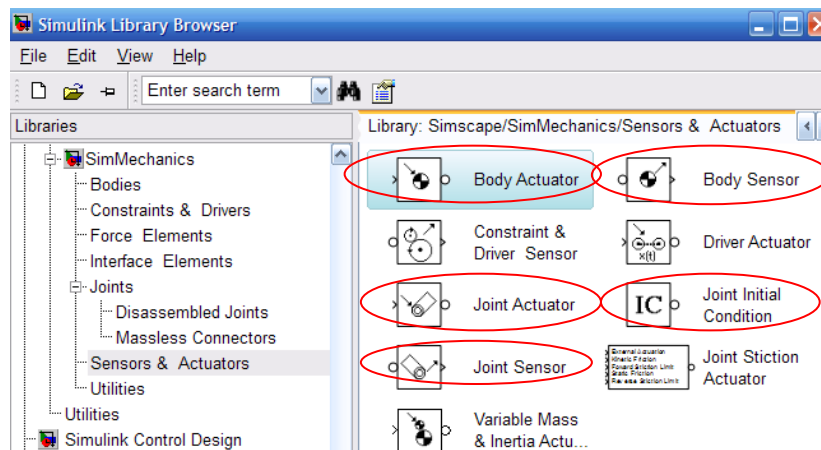


Figura 4-19

Neste menu na Figura 4-19 encontram-se todos os sensores e actuadores para condicionarmos os blocos *Joints* ou *Bodies*.

Bloco “BODY ACTUATOR”

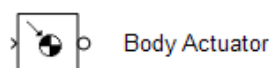


Figura 4-20

O bloco *Body Actuator* (Figura 4-20) tem de ser conectado a um bloco *Body* e serve para actuar nesse bloco com uma força/torque. A entrada deste bloco pode ser um sinal de simulink. Terá de ser um vector coluna com 3 linhas em que cada linha tem o valor aplicado num dado eixo, ou seja, terá de ser um vector do tipo [valor segundo x; valor segundo y; valor segundo z]. Se a força que se pretende aplicar no corpo for apenas na direcção de um dos eixos, coloca-se o vector com valores nulos nos outros eixos, por exemplo se quisermos aplicar no corpo uma força de 10N segundo o eixo dos y, o sinal de entrada poderia ser um vector do tipo [0; 10; 0]. Na Figura 4-21 tem-se os parâmetros do bloco *Body Actuator*.

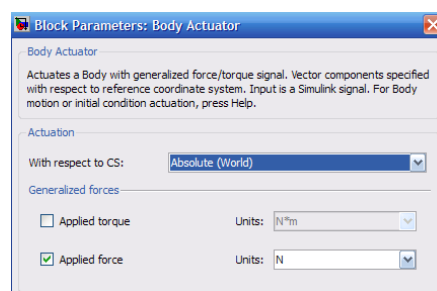


Figura 4-21

Bloco “BODY SENSOR”

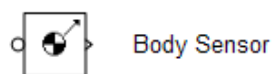


Figura 4-22

O bloco *Body Sensor* (Figura 4-22) tem de ser conectado a um bloco *Body* e serve para se medir a posição, velocidade linear, velocidade angular, matriz de rotação, aceleração e aceleração angular (ver Figura 4-24). Para cada corpo (bloco *Body*) é suficiente um bloco *Body sensor* para se medirem as variáveis referidas bastando activar as respectivas checkboxes (ver Figura 4-24). O sinal de saída para cada variável será uma matriz com 3 linhas segundo cada um dos eixos do tipo [x;y;z].

Pode optar-se por colocar qualquer uma das variáveis a medir na mesma matriz de saída aumentando assim o número de colunas por cada variável diferente (ver Figura 4-24). Por outro lado pode-se optar por desactivar a opção “Output selected parameters as one signal” (Figura 4-24) obtendo-se assim uma porta de saída diferente para cada variável a medir. Como exemplo, se se activassem as checkboxes de “Position” e “Velocity” obtinha-se um bloco *Body Sensor* com duas portas de saída, cada uma com uma matriz [x;y;z] referente à posição e velocidade (Figura 4-23).

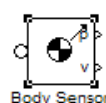


Figura 4-23

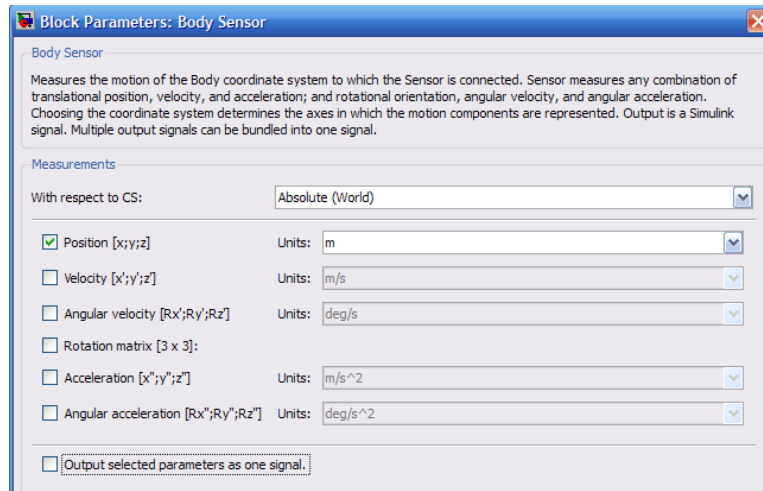


Figura 4-24

Bloco “JOINT ACTUATOR”



Figura 4-25

O bloco *Joint Actuator* (Figura 4-25) serve para se actuar numa junta. Tem de ser ligado com um sinal de entrada do Simulink (que pode ser um bloco do Simulink, ou um vector da área trabalho do Matlab). Esse sinal vai ser aplicado na junta conforme as definições escolhidas no bloco *Joint Actuator*. Contrariamente ao sinal aplicado ao bloco *Body Actuator*, este sinal a aplicar no bloco *Joint Actuator* é um vector linha que actua apenas no(s) eixo(s) do grau de liberdade da junta. O bloco permite actuar na junta com forças (propriedades na Figura 4-26) ou com movimento (propriedades na Figura 4-27).

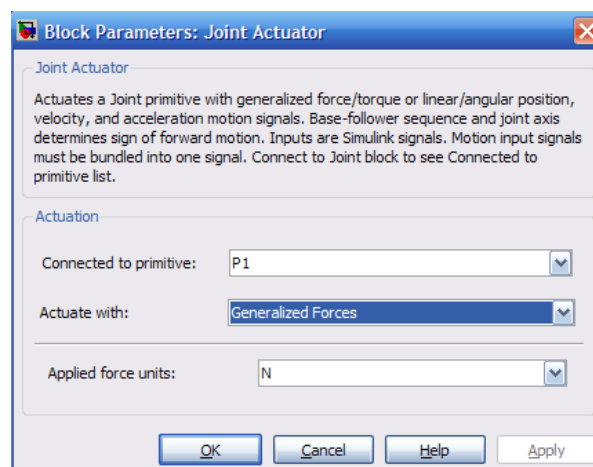


Figura 4-26

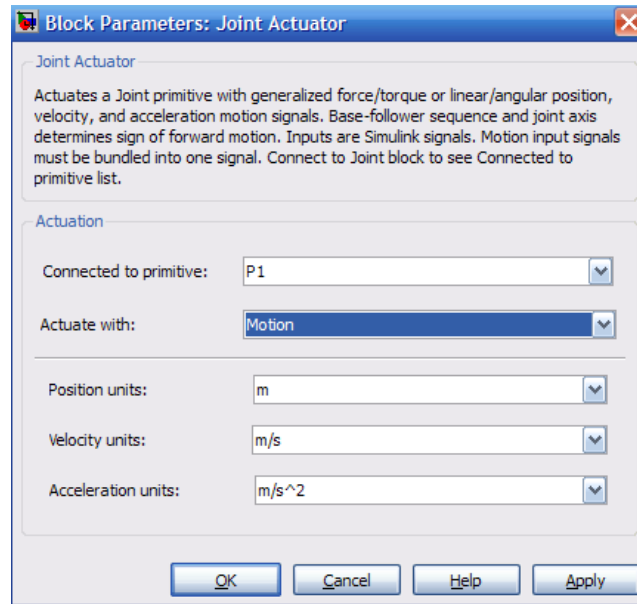


Figura 4-27

Bloco “JOINT SENSOR”

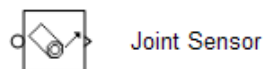


Figura 4-28

O bloco *Joint Sensor* (Figura 4-28) deve ser ligado a um bloco do menu *Joints*. Permite obter valores da posição, velocidade, aceleração, força, torque reactivo e força de reacção (ver Figura 4-30). Se este bloco se encontrar ligado a uma junta que permita movimento de rotação (em vez de translação como no exemplo da Figura 4-30), o bloco permite obter o ângulo, velocidade angular, aceleração angular, torque, torque reactivo e a força reactiva. Como no caso do bloco *Body Sensor*, tem-se a opção de inserir os sinais seleccionados num vector apenas. Optando-se por desligar a checkbox “Output selected parameters as one signal” e seleccionando, por exemplo, a posição e a velocidade como sinais a medir, o bloco toma o aspecto de acordo com a Figura 4-29 em que numa saída tem-se a posição e noutra a velocidade.

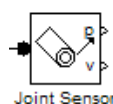


Figura 4-29

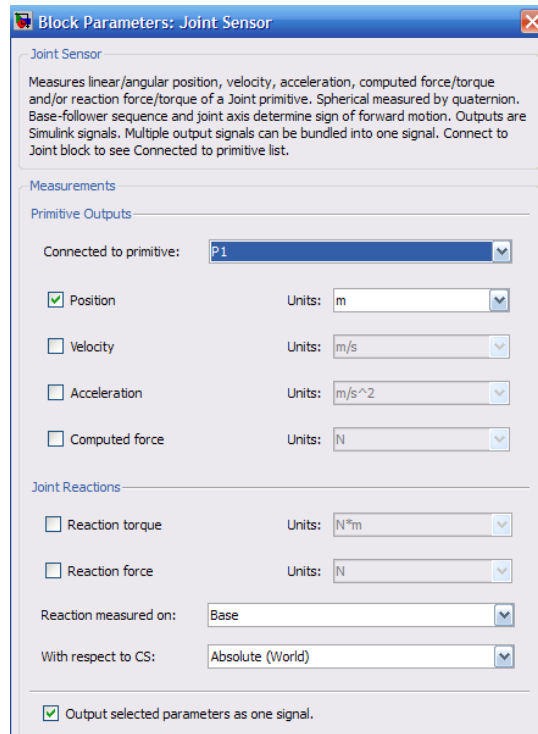


Figura 4-30

Bloco “JOINT INITIAL CONDITION”

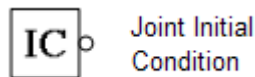


Figura 4-31

O bloco *Joint Initial Condition* (Figura 4-31) serve para definir as condições iniciais dum bloco *Joint* a que esteja ligado. Serve, por exemplo, para a inserção da gravidade (se esta não tiver sido inserida através do bloco *Machine Environment*) ou para se, no início da simulação, se pretender que uma junta se encontre em movimento com uma dada velocidade ou numa posição diferente da do bloco a que se encontra ligada (Figura 4-32). Se este bloco se encontrar conectado a uma junta que permita movimento rotacional os campos iniciais a alterar serão o do ângulo e da velocidade angular

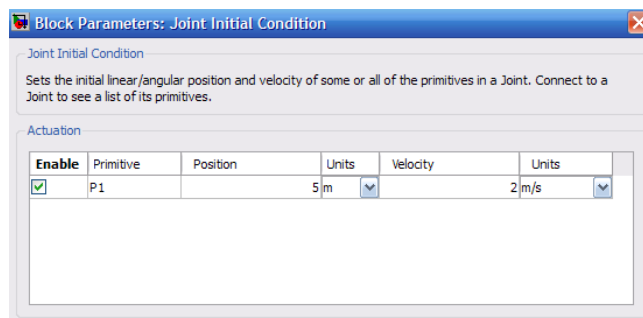


Figura 4-32

5 Casos de estudo

5.1 Controlo da velocidade de um automóvel

5.1.1 Introdução

Pretende-se determinar o modelo matemático de uma viatura com massa (M) que tem como entrada a força gerada pelo motor (F) e como saída a velocidade da viatura (v) (Figura 5-1). Considera-se a inércia das rodas desprezável e tendo como única oposição ao movimento o atrito gerado pelo ar (f).

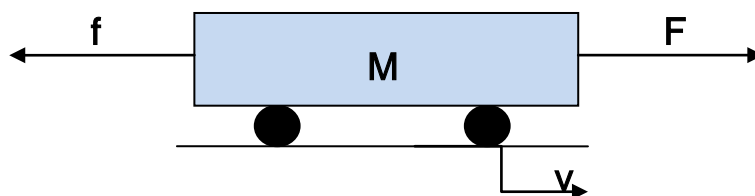


Figura 5-1 – Esquema de Forças aplicadas no Carro

Os dados de massa, comprimento, altura no Anexo 1 com a seguinte estrutura:

$a=3$; %comprimento do carro (m)

$b=2$; %largura do carro (m)

$c=1.5$; %altura do carro (m)

$M=1000$; %massa do carro (Kg)

$f=50$; %Constante de atrito (N)

$ref=10$; %velocidade a atingir - referência (m/s)

$I_t=[(m/12)*(b^2+c^2) \ 0 \ 0; \ 0 \ (m/12)*(a^2+c^2) \ 0; \ 0 \ 0 \ (m/12)*(a^2+b^2)]$ %Matriz de Inércia

Os dados da massa, constante de atrito e a referência foram retirados do site [5].

Os dados de comprimento, largura e altura foram escolhidos próximos dos valores reais de um carro.

Por fim, I_t , a Matriz de Inércia do carro, foi retirada de [16] sendo a Matriz de Inércia de um elipsóide com comprimento, largura e altura definidos como a , b e c respectivamente.

5.1.2 Modelo virtual do automóvel

Na Figura 5-2 tem-se o bloco de simulink que representa o sistema mecânico do carro.

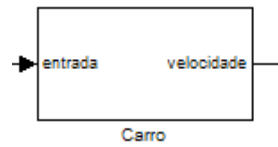


Figura 5-2

Na Figura 5-3 encontra-se o modelo virtual em SimMechanics para o automóvel que é o subsistema do bloco “Carro” da Figura 5-2.

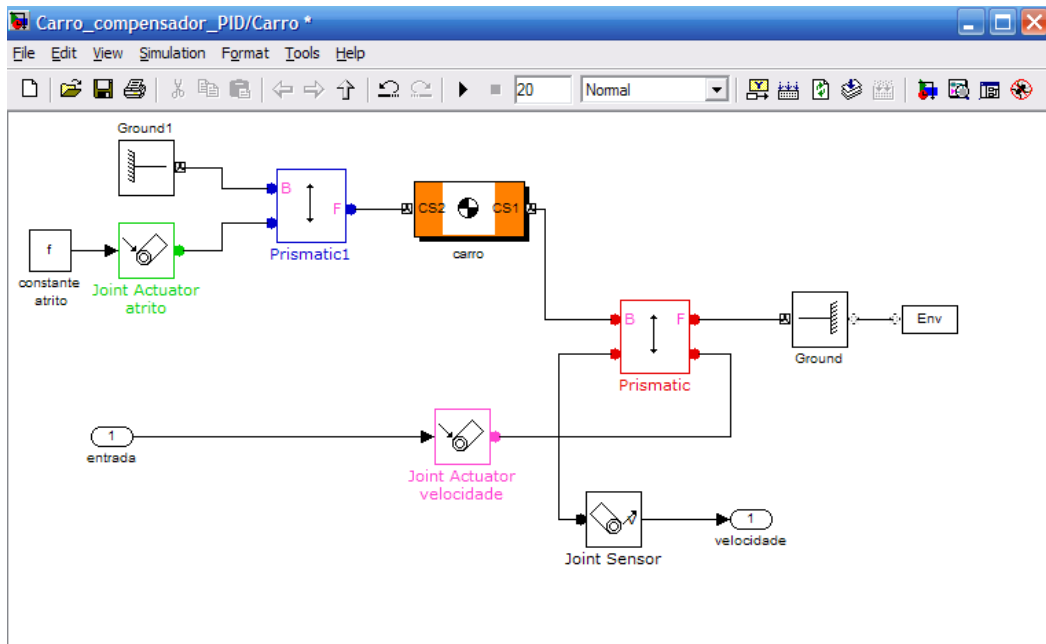


Figura 5-3

Tem-se um bloco *Body* (com nome *carro*) que simula o carro ligado no ponto CS2 a uma junta (*Prismatic1*). Nesta junta é aplicado o atrito (no eixo do x) através de um bloco *Joint Actuator atrito* que actua através da introdução de uma constante de atrito f . No outro ponto de ligação do bloco *Body* (CS1) está conectada uma segunda junta (*Prismatic*) que actua no eixo x como a anterior mas no sentido oposto que simula o movimento provocado pelo motor do carro. Este movimento é aplicado pelo bloco *Joint Actuator velocidade*.

As propriedades do bloco *Body* (*carro*) encontram-se definidas na Figura 5-4. A massa do carro é uma variável m cujo valor é atribuído no “Workspace” do Matlab. A Inércia I_t é definida segundo o momento de inércia de um elipsóide (ver [16] onde estão todas as matrizes dos momentos de Inércia para formas mais comuns).

$$I_t = \begin{bmatrix} (m/12)*(b^2+c^2) & 0 & 0; \\ 0 & (m/12)*(a^2+c^2) & 0; \\ 0 & 0 & (m/12)*(a^2+b^2) \end{bmatrix}$$

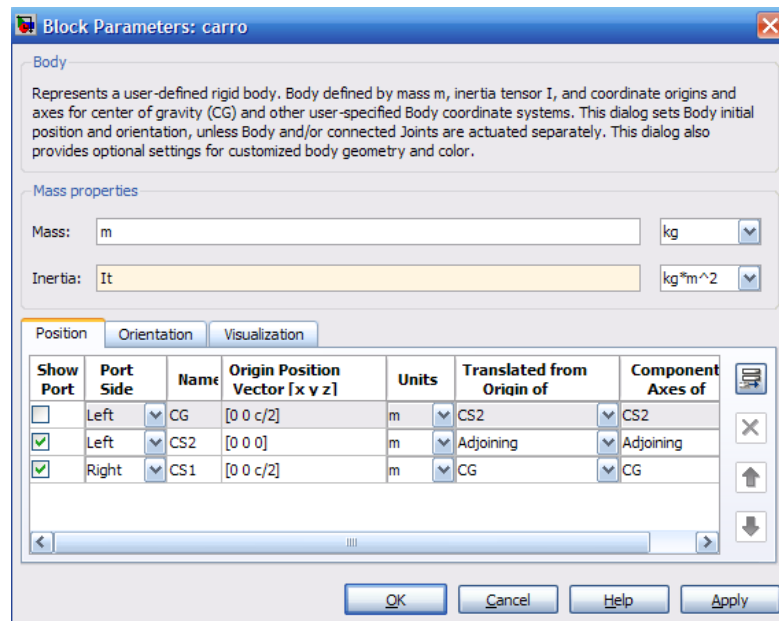


Figura 5-4

O bloco *carro* encontra-se conectado a duas juntas *Prismatic Joint*. As duas juntas permitem um movimento horizontal segundo o eixo do x.

Na junta *Prismatic* é aplicada a força que provoca o movimento do carro. Assim, na Figura 5-5 pode-se verificar as propriedades desta junta que possui duas portas: uma para o actuador de velocidade e um sensor de movimento (*Joint Sensor*). Tem definido como eixo de acção o eixo do x (“Axis Of Action” [-1 0 0]). O ponto de referência é o bloco *Ground* (“Current Follower”) a que esta está conectada.

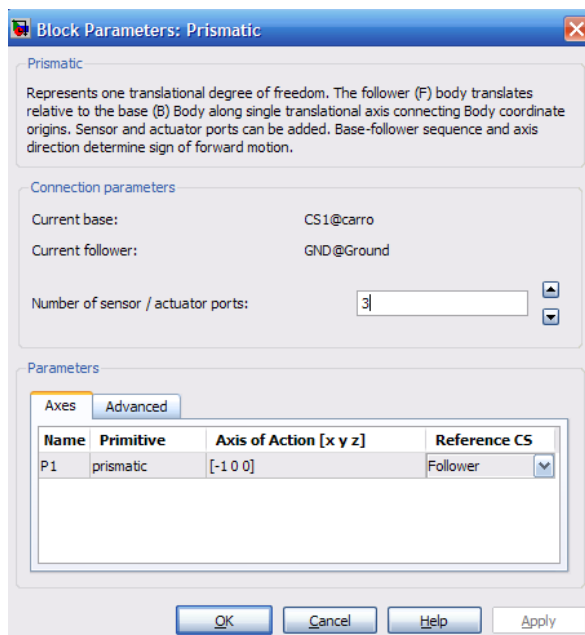


Figura 5-5

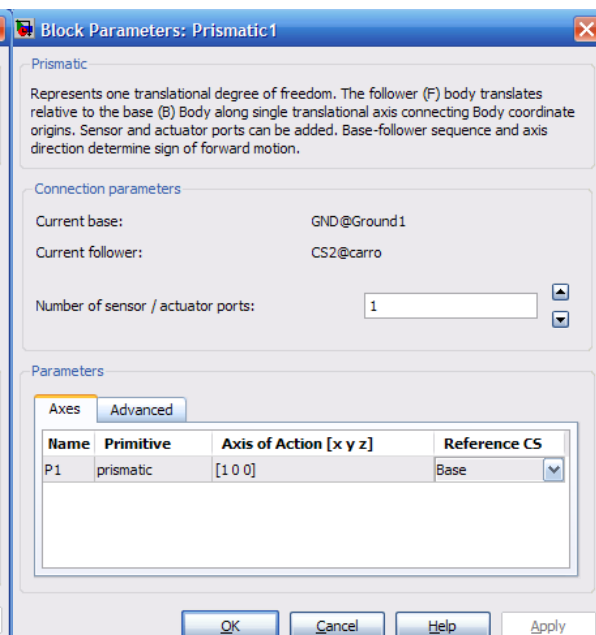


Figura 5-6

Na junta *Prismatic1* é aplicado movimento no mesmo eixo que a anterior, mas no sentido contrário, pois é onde actua o atrito provocado pelo ar. Como se pode observar na Figura 5-6, esta junta possui apenas uma porta para um actuador (a força de atrito) e o vector de acção será [1 0 0] (simétrico ao anterior).

O atrito é aplicado na junta *Prismatic1* através de um bloco *Joint Actuator atrito* que tem como entrada uma constante f (definida no Anexo 1). Nas propriedades deste bloco basta definir que se actua com “Generalized Forces” e unidades “N” (ver Figura 5-7).

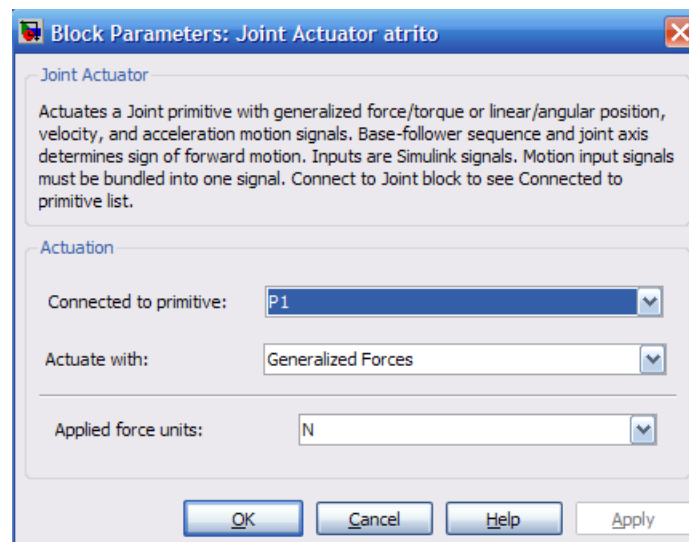


Figura 5-7

A junta *Joint Sensor* encontra-se ligada a um bloco *Joint Actuator velocidade* que é onde é aplicada a força gerada pelo motor para o movimento do carro. As propriedades deste bloco são as idênticas à da Figura 5-7.

A junta *Prismatic* possui ainda um bloco *Joint Sensor* para se medir a velocidade a que se move o carro. Na Figura 5-8 pode-se verificar que, visto que a dimensão a ser medida é a velocidade, encontra-se activa a checkbox correspondente à velocidade (unidades em m/s).

O movimento é medido na base da junta (“Base”, ou seja, no bloco carro) em relação à origem do AVS (“World”).

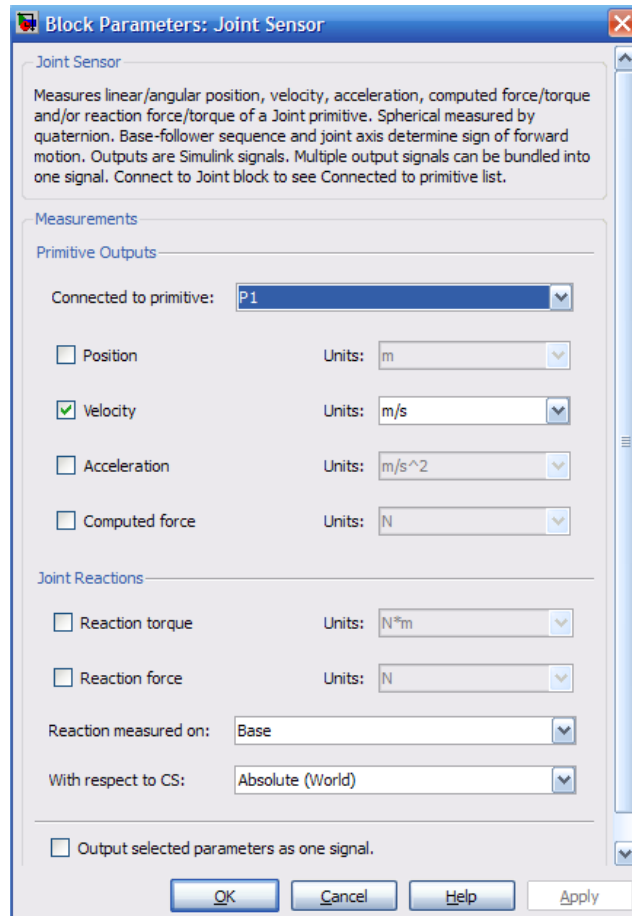


Figura 5-8

Neste modelo os pontos de localização para as posições iniciais começaram a ser definidos no bloco *Ground1* (ver Figura 5-3). As propriedades deste bloco estão definidas na Figura 5-9. Optou-se por colocar o bloco *Ground1* no fundo e no centro do carro em relação ao AVS (visto que a sua localização é metade do comprimento do carro, metade da largura do carro e com altura nula $[a/2 \ b/2 \ 0]$).

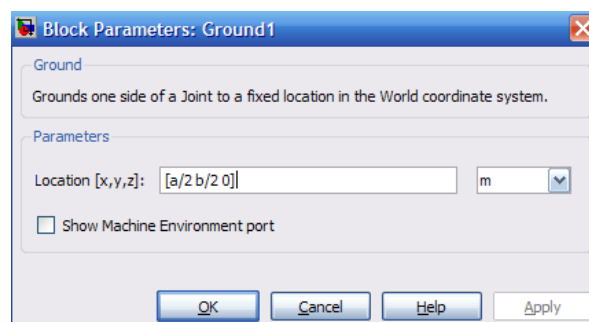


Figura 5-9

De acordo com a Figura 5-4 pode-se verificar que o ponto CS2 se encontra deslocado de $[0 \ 0 \ 0]$ do bloco adjacente, neste caso o bloco *Ground1*. Como o bloco *Ground1* tem

localização $[a/2 \ b/2 \]$ o ponto CS2 encontra-se na mesma posição. Este é o ponto onde será aplicada a força de atrito provocado pelo ar.

O centro de massa (ponto “CG”) do bloco carro encontra-se no centro deste, visto que esse ponto está definido segundo uma translação de $[0 \ 0 \ c/2]$ do ponto “CS2” (ver Figura 5-4). Sendo assim, em relação à origem do AVS, “CG” está localizado em $[a/2 \ b/2 \ c/2]$.

O ponto onde é aplicado o movimento provocado pelo motor é o ponto “CS1” (ver Figura 5-3). Encontra-se localizado em $[a/2 \ b/2 \ c]$ (no cimo do carro). O deslocamento dá-se, neste caso, segundo um vector $[0 \ 0 \ c/2]$ em relação às coordenadas do centro de massa ($[a/2 \ b/2 \ c/2]$) (Figura 5-4).

Por fim, falta definir a posição inicial do bloco *Ground*, que é a mesma posição do ponto “CS1” (ver Figura 5-10).

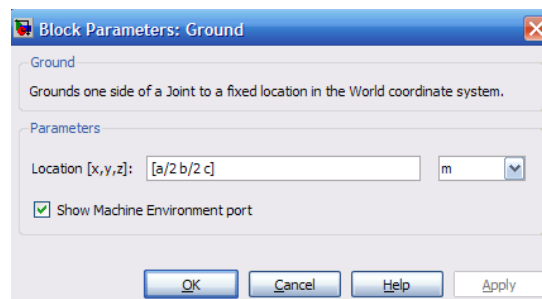


Figura 5-10

As propriedades do bloco da constante de atrito encontram-se na Figura 5-11.

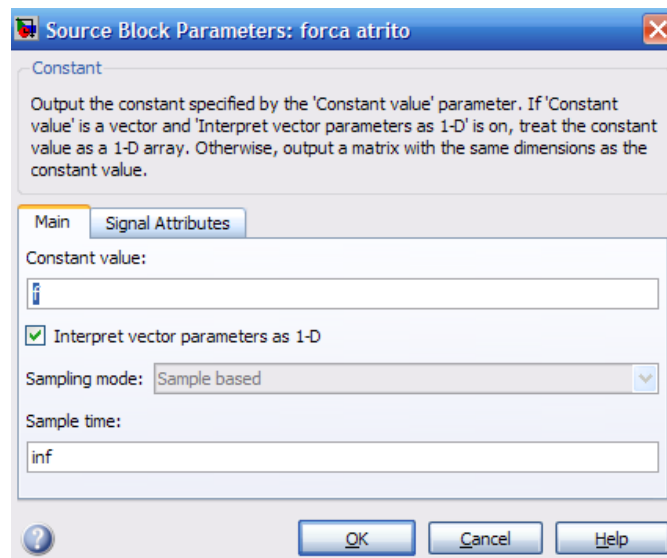


Figura 5-11

Por fim, o bloco *Machine Environment* na Figura 5-12 (o separador “Parameters”) e Figura 5-13 (o separador “Visualization”).

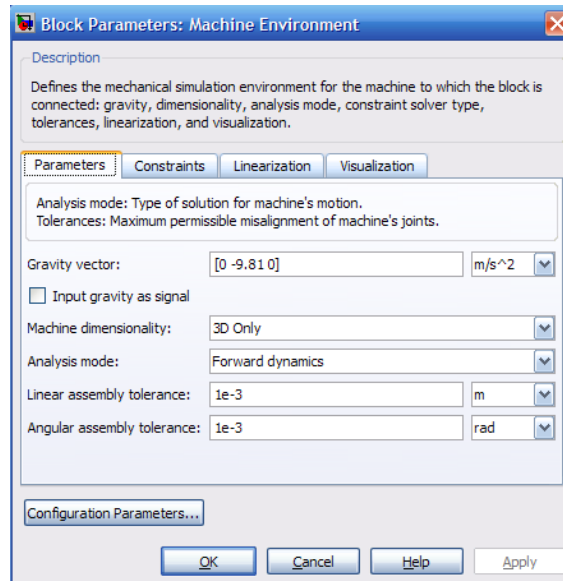


Figura 5-12

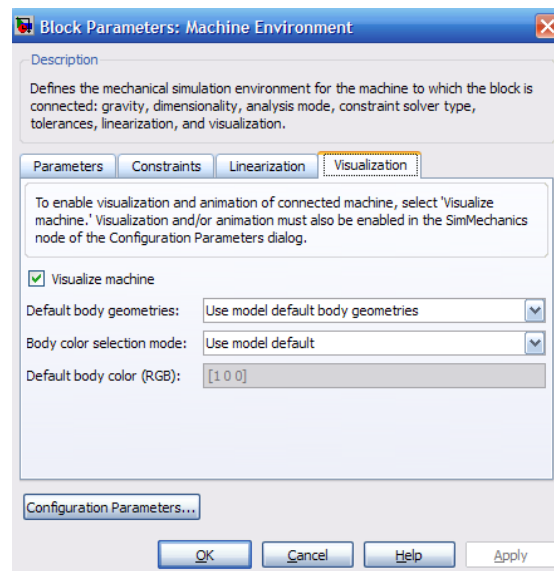


Figura 5-13

O que se alterou no bloco *Machine Environment* foi o eixo da actuação da gravidade ([0 - 9.18 0] – ver Figura 5-12), o marcador na checkbox “Visualize machine” e o campo “Default body geometries” para “Use model default body geometries” (ver Figura 5-13). Todos os outros campos, mesmo dentro dos outros separadores, foram deixados como pré-definidos do bloco.

O esquema de eixos de actuação no carro encontra-se na Figura 5-14:
Note-se que o eixo vertical é, neste caso, o eixo y.

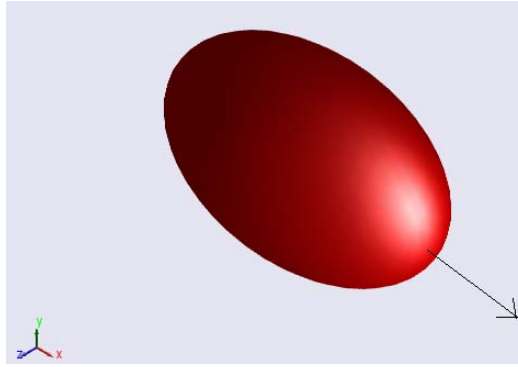


Figura 5-14 – Esquema de eixos

5.1.3 Especificações da dinâmica do automóvel

As especificações formuladas para este caso foram de acordo com as referidas no site [5].

Um automóvel deve conseguir, quando um motor fornece uma força de 500N, chegar a uma velocidade de, aproximadamente, 10 m/s (36 Km/h). Deve, também, conseguir atingir essa velocidade dentro de 7 segundos. Com este modelo pretende-se controlar a velocidade de um automóvel.

Assim, as especificações para o regime transitório são:

- Tempo de subida < 5 seg;
- Sobrelevação (“Overshoot”) < 10%;

Para o regime estacionário as especificações são:

- Referência = 10 m/s (referência para a velocidade);
- Erro em regime estacionário < 2%;

O principal objectivo desta dissertação é implementar e testar os modelos virtuais desenvolvidos através da ferramenta SimMechanics em esquemas de controlo. Para determinar os parâmetros dos controladores nestes esquemas utilizou-se técnicas de controlo de sistemas lineares. Para tal, primeiro precisou-se de determinar um modelo linear que representasse a dinâmica do objecto de controlo.

Neste contexto, a equação diferencial que vem do balanço das forças aplicadas ao automóvel é:

$$M \cdot \dot{y} + f \cdot y = u \quad (1)$$

A variável y é a velocidade do automóvel (a variável que se pretende controlar) e u é a força aplicada ao motor (a entrada do automóvel).

Aplicando a transformada de Laplace na equação (1), assumindo condições iniciais nulas vem:

$$M \cdot s \cdot Y(s) + f \cdot Y(s) = U(s) \quad (2)$$

Obtendo-se a seguinte função de transferência:

$$\frac{Y(s)}{U(s)} = \frac{1}{M \cdot s + f} \quad (3)$$

O modelo da equação (3) é utilizado em “Simulink” para simular o automóvel. Todos os comandos necessários em Matlab se encontram no Anexo 1.

5.1.4 Esquema do controlo do automóvel com modelo linear e controlador PID

Equação dinâmica do controlador PID:

$$u_{PID}(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int e(t) dt \quad (4)$$

$$\text{onde } e(t) = \text{referência} - y(t) \quad (5)$$

e K_p , K_d e K_i são, respectivamente, a constante de proporcionalidade, constante de ganho derivador e constante de ganho integrador.

Aplicando a transformada de Laplace na equação (4) obtém-se a seguinte função de transferência para o controlador PID:

$$C_{PID} = \frac{U(s)}{E(s)} = K_p + K_d \cdot s + \frac{K_i}{s} \quad (6)$$

Através das técnicas aprendidas nas aulas práticas da cadeira de Sistemas e Controlo 2 calcularam-se os valores para os coeficientes do controlador (ver [1]):

$K_p=800$; $K_d=100$; $K_i=40$;

O modelo da equação (6) do controlador PID foi realizado em “Simulink” com o diagrama presente na Figura 5-15 e inserido na malha de controlo com realimentação unitária negativa representada na Figura 5-16.

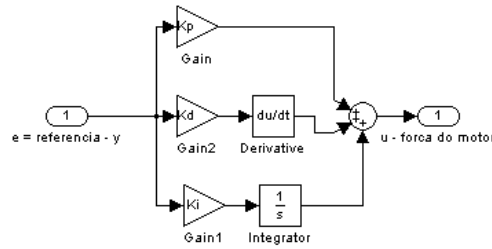


Figura 5-15 - Diagrama de blocos do controlador PID

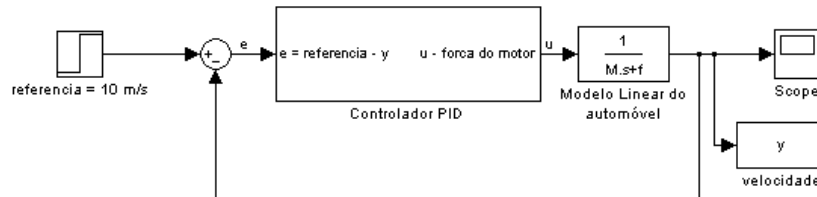


Figura 5-16 - Diagrama de blocos da malha de controlo com a Função de Transferência do carro e com Controlador PID

O sistema foi simulado durante 20 segundos e os resultados encontram-se na Figura 5-17. Como se pode verificar, a variável a controlar (velocidade do automóvel) atinge a referência pretendida de 10 segundos e mantém o valor desejado dentro das especificações referidas.



Figura 5-17

5.1.5 Esquema do controlo do automóvel com modelo linear e controladores Atraso/Avanço

Os modelos dos controladores Atraso e Avanço representados com a Função de Transferência (FT) no domínio das frequências têm estruturas idênticas mas valores de parâmetros diferentes.

FT do controlador Atraso:

$$C_{Atraso}(s) = \frac{U(s)}{E(s)} = K \frac{s + A}{s + L \cdot A} \quad \text{com} \begin{cases} 0 < L < 1 \\ A > 1 \end{cases} \quad (7)$$

FT do controlador Avanço:

$$C_{Avanço}(s) = \frac{U(s)}{E(s)} = K \frac{s + A}{s + L \cdot A} \quad \text{com} \begin{cases} 0 < A < 1 \\ L > 1 \end{cases} \quad (8)$$

Através das técnicas clássicas ([1]) foram calculados os seguintes valores para o controlador Atraso: $A=1$, $L=0.1$; e para o controlador Avanço: $A=0.1$, $L=10$;

Os dois algoritmos de controlo foram realizados com a malha realimentada da Figura 5-18, alterando-se os valores de A e de L conforme acima referidos. Os resultados, após 20 segundos de simulação, são os resumidos na Figura 5-19.

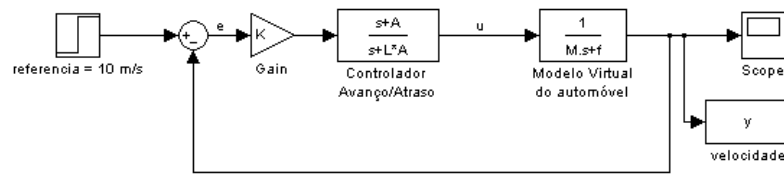


Figura 5-18 - Diagrama de blocos da malha de controlo com Função de Transferência do Carro e com controlador Atraso/Avanço

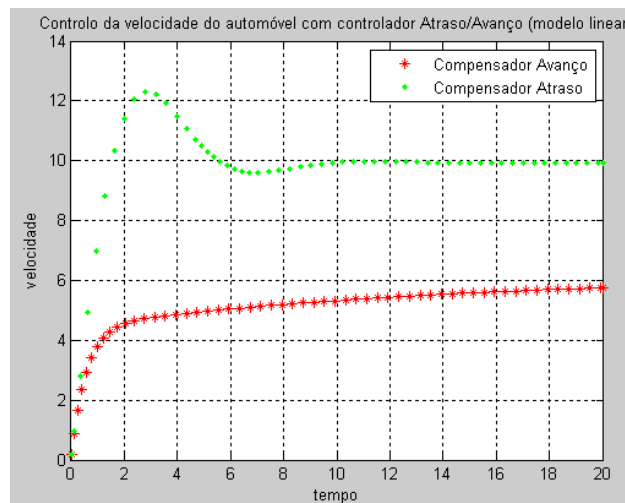


Figura 5-19

Como se pode verificar, o controlador Avanço não cumpre as especificações. Enquanto o controlador Atraso, apresenta um comportamento notoriamente melhor com sobrelevação superior a 10%.

As três simulações, com controladores diferentes (PID, Atraso e Avanço) levam à conclusão que o controlador PID é a melhor forma de cumprir com as especificações delineadas para o automóvel e o controlador Avanço não é adequado para este caso de estudo.

5.1.6 Simulação de controlo do automóvel no Laboratório Virtual

Neste capítulo, os três controladores projectados no capítulo anterior serão aplicados no Laboratório Virtual para controlar a velocidade do automóvel representado com o modelo virtual do SimMechanics (ver capítulo 5.1.2).

Controlador PID

O esquema realimentado de controlo com controlador PID é representado na Figura 5-20. O bloco *Controlador PID* é o mesmo usado na Figura 5-15, com os mesmos valores dos parâmetros K_p , K_d e K_i referidos no capítulo anterior. O bloco *Modelo Virtual do Carro* é o mesmo da Figura 5-3. A simulação decorreu ao longo de 20 segundos.

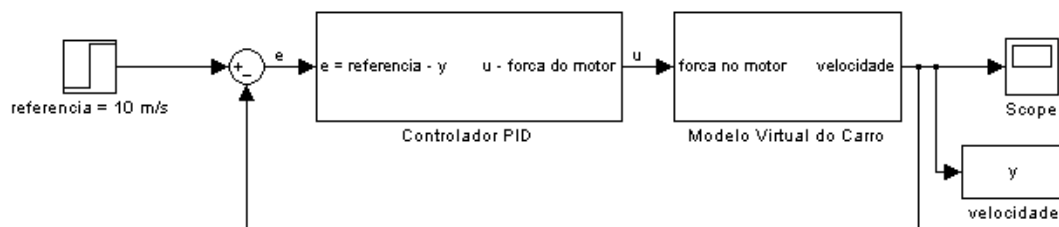


Figura 5-20 - Diagrama de blocos da malha de controlo do modelo virtual do automóvel com controlador PID

Na Figura 5-21 estão sobrepostas as trajectórias da velocidade do automóvel que foi controlada usando o mesmo controlador PID para o modelo linear (Figura 5-16) e para o modelo virtual (Figura 5-20).

A semelhança das curvas leva à conclusão que o modelo virtual desenvolvido no capítulo 5.1.2 representa adequadamente a dinâmica do automóvel e que o controlador consegue manter as especificações pretendidas.

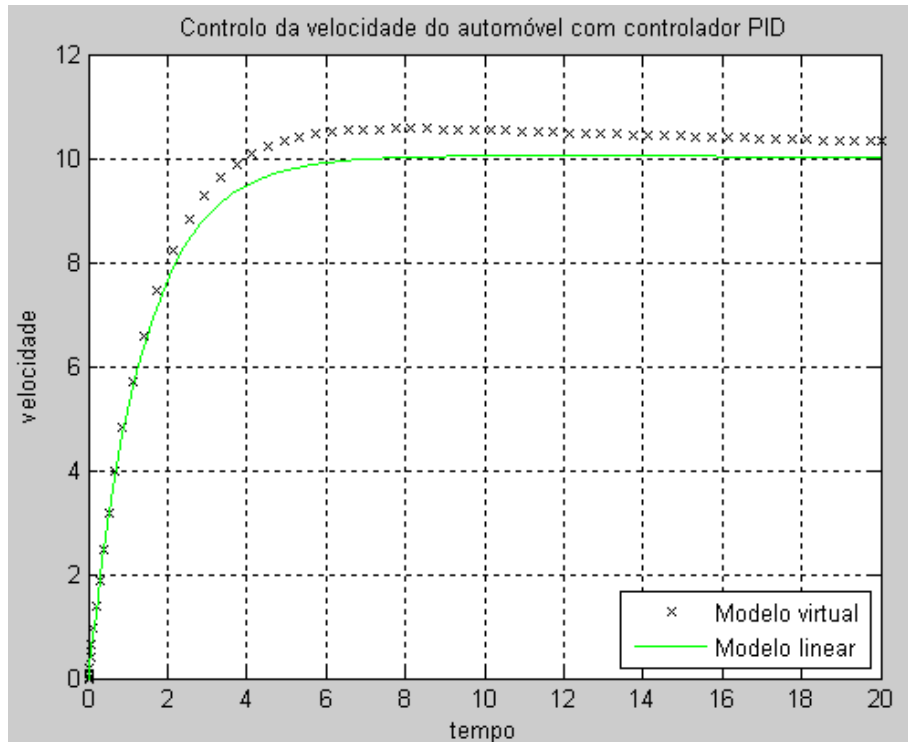


Figura 5-21

Controladores Atraso/Avanço

A malha de controlo para o controlador Atraso é a mesma para o controlador Avanço onde se alteraram apenas os parâmetros A e L (Figura 5-22). Os valores de A e L são os mesmos do capítulo anterior e o bloco *Modelo Virtual do Carro* é o representado na Figura 5-3.

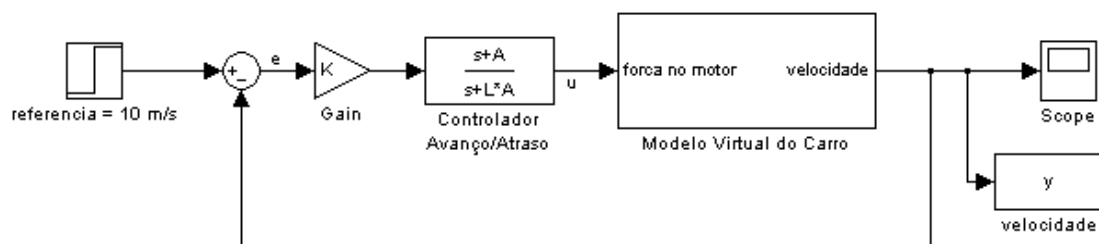


Figura 5-22 - Diagrama de blocos da malha de controlo do modelo virtual do automóvel com controlador Atraso/Avanço

Após executar a simulação durante 20 segundos para os controladores Atraso e Avanço (Figura 5-22) colocou-se as duas curvas juntamente com as curvas obtidas durante a simulação do modelo linear com os controladores Atraso e Avanço (Figura 5-18). Os resultados estão esquematizados na Figura 5-23.

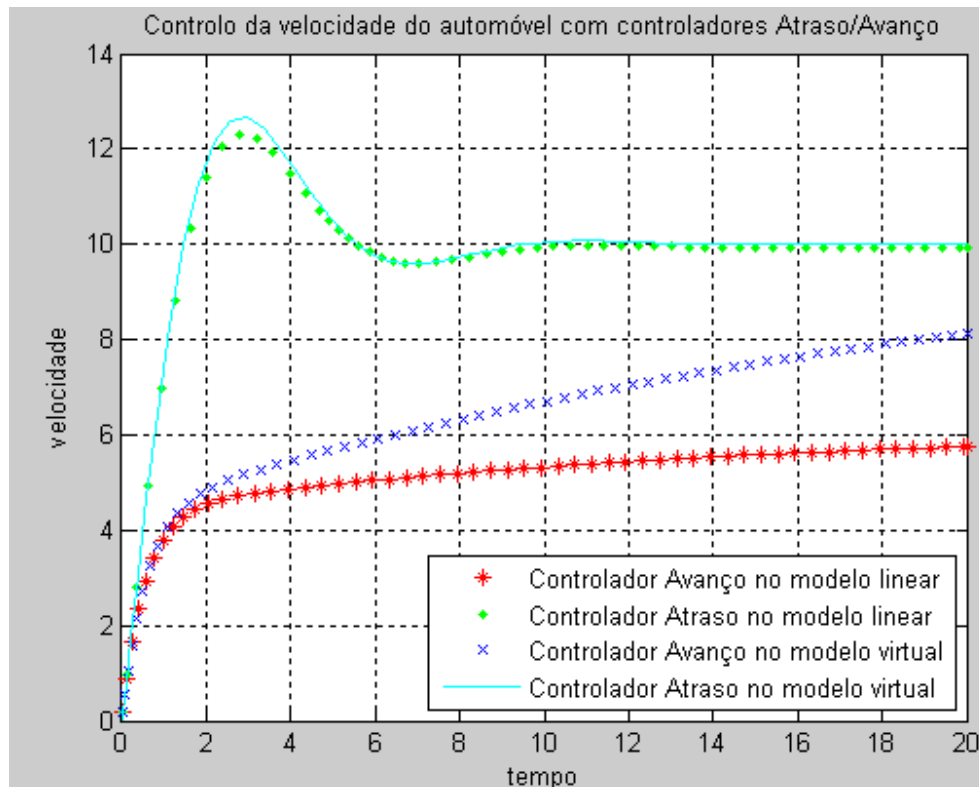


Figura 5-23

Através da observação das curvas de velocidade, e como estas são similares, pode-se concluir que o modelo elaborado no Laboratório Virtual se encontra de acordo com a dinâmica do automóvel. Verifica-se, mais uma vez, que o controlador Avanço não é adequado para este modelo e que o controlador Atraso cumpre com todas as especificações excepto com o limite proposto para a sobrelevação.

5.1.7 Simulação de controlo do automóvel com alteração da referência

Para verificar o comportamento do sistema numa situação com alteração na referência de entrada optou-se por simular uma situação em que o automóvel acelera até aos 10m/s e, após decorridos 10 segundos, aumenta a velocidade até aos 20m/s.

Como o controlador PID foi o que melhor cumpriu com as especificações optou-se por usar este controlador para a simulação definida (o mesmo bloco usado na Figura 5-15). A malha de controlo aplicada ao modelo linear encontra-se na Figura 5-24. O bloco “referencia” tem as suas propriedades definidas na Figura 5-25.

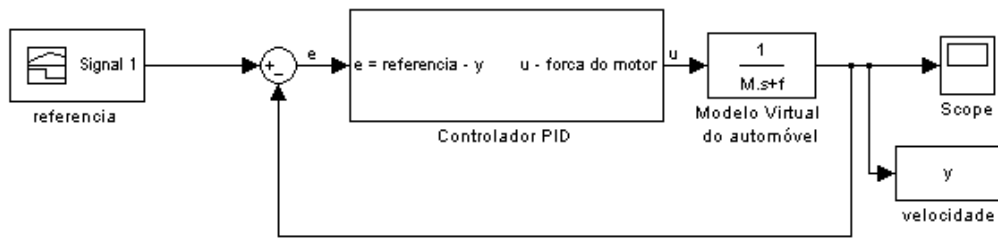


Figura 5-24 - Diagrama de blocos da malha de controlo com Função de Transferência do Carro, com Controlador PID e com uma referência variada

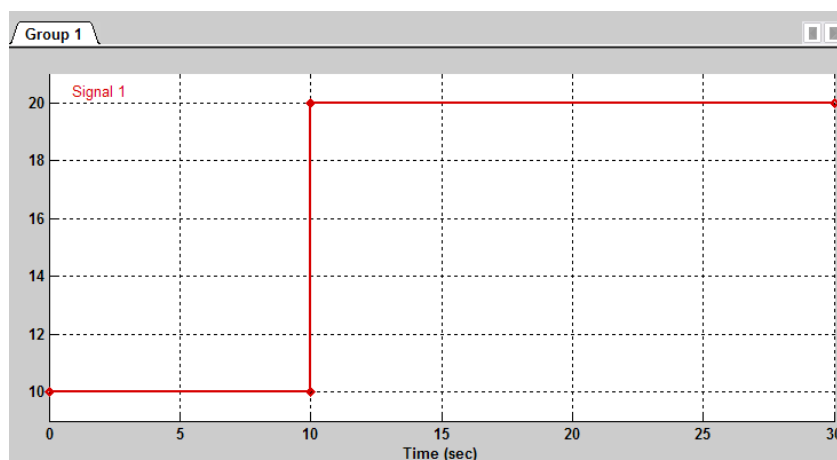


Figura 5-25 – Diagrama do sinal de referência - bloco referencia

Para a simulação usando o modelo virtual usou-se um bloco *referencia* com as propriedades idênticas às da Figura 5-25. A malha de realimentação está esquematizada na Figura 5-26.

Os resultados das duas simulações, ao longo de 20 segundos, encontram-se resumidos com a sobreposição das duas curvas de velocidade na Figura 5-27.

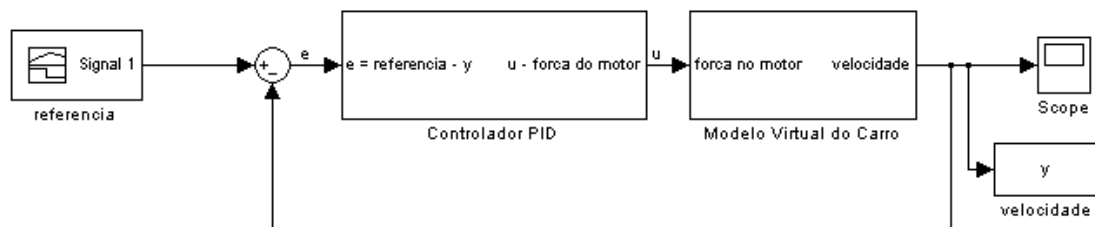


Figura 5-26 - Diagrama de blocos da malha de controlo do modelo virtual do automóvel com controlador PID e com uma referência variada

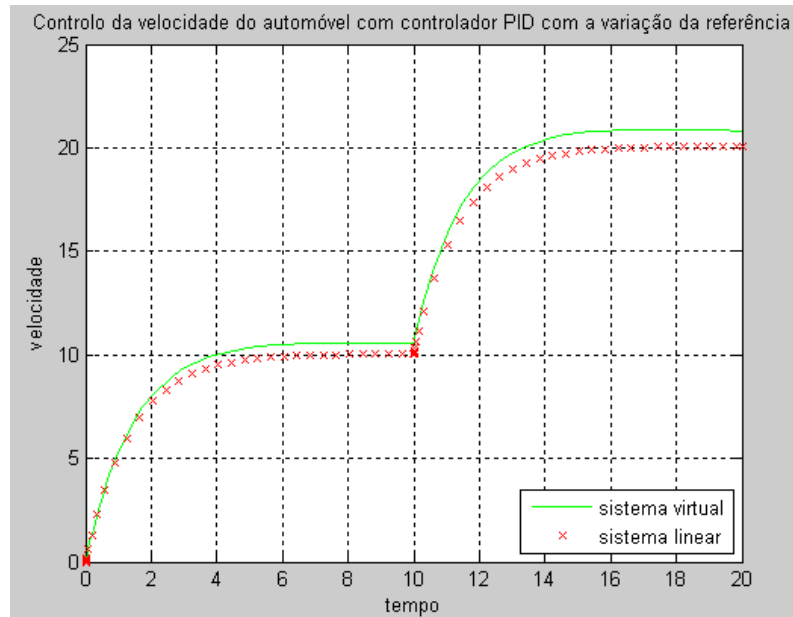


Figura 5-27

Pode-se verificar que as curvas são semelhantes, o que leva a concluir que o modelo virtual se encontra de acordo com a dinâmica real de um automóvel e que o controlador PID usado se adequa, também, nesta situação, com alteração do sinal de referência.

5.2 Controlo da posição do pêndulo invertido

5.2.1 Introdução

O Pêndulo Invertido é um objecto clássico de controlo que se baseia num carro com um pêndulo invertido colocado em cima deste. O sistema em repouso é definido pelo pêndulo se encontrar na vertical com o carro parado. O objectivo é que, após ser aplicada uma perturbação no pêndulo, o carro exerça um movimento contrário ao do pêndulo de modo a compensar a perturbação que este sofreu. Este sistema encontra-se esquematizado na Figura 5-28.

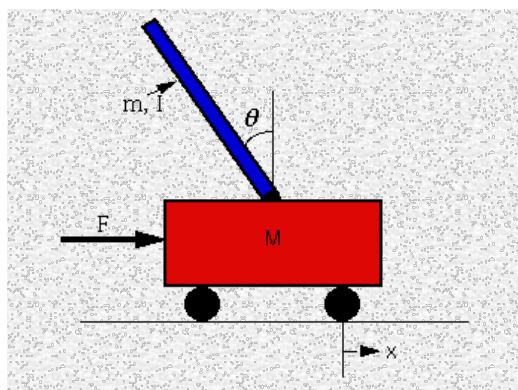


Figura 5-28 – Esquema de Forças do Pêndulo Invertido

Os dados referentes a este problema (massa, comprimento, altura e largura) encontram-se no Anexo 2 com a seguinte estrutura:

```
M=0.5; %massa do carro
m=0.2; %massa do pêndulo
a=0.6; %comprimento do carro
b=0.2; %largura do carro
c=0.3; %altura do carro
ap=0.1; %comprimento do pêndulo
bp=0.2; %largura do pêndulo
cp=0.6; %altura do pêndulo
g=9.8; %gravidade
l=cp/2; %distância do pêndulo ao seu centro de massa
It=[(M/12)*(b^2+c^2) 0 0; 0 (M/12)*(a^2+c^2) 0; 0 0 (M/12)*(a^2+b^2)];
%Matriz de Inércia do carro
Itp=[(m/12)*(bp^2+cp^2) 0 0; 0 (m/12)*(ap^2+cp^2) 0; 0 0 (m/12)*(ap^2+bp^2)];
%Matriz de Inércia do pêndulo
%Para efeitos de cálculo das variáveis de estado é necessário ter um valor para a Inércia
do pêndulo (em vez duma matriz). Assim, considerou-se o maior momento de Inércia de
cada um dos eixos da matriz de inércia do pêndulo:
l=max(max(Itp));
```

Os dados da massa do carro, massa do pêndulo, altura do pêndulo e momento de inércia do pêndulo foram retirados do livro [11].

Os dados do comprimento, largura e altura do carro e comprimento e largura do pêndulo foram arbitrados com valores razoáveis de um modelo real desta simulação.

Os valores das Matrizes de Inércia foram retirados de [16] referentes à Matriz de Inércia de um paralelepípedo.

5.2.2 Modelo do Pêndulo Invertido

Na Figura 5-29 tem-se o bloco em “SimMechanics” que contem o modelo mecânico do pêndulo invertido.

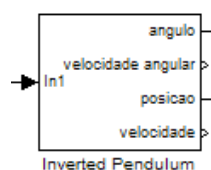


Figura 5-29

Na Figura 5-30 tem-se o subsistema do bloco da Figura 5-29, o Pêndulo Invertido nos seus blocos constituintes mais simples.

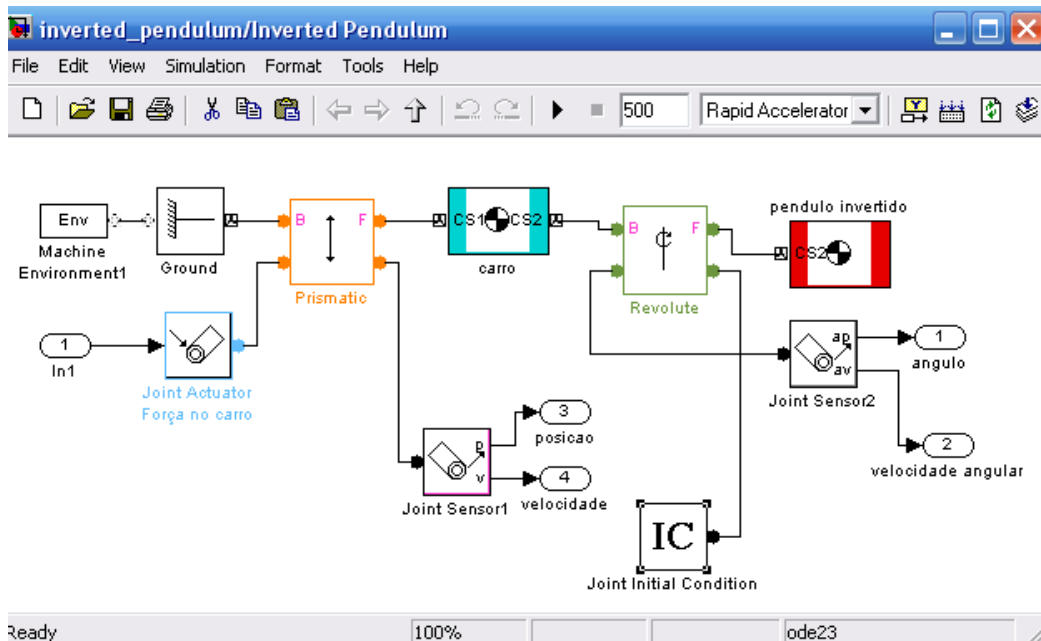


Figura 5-30

O sistema é constituído por dois blocos *Body*, o pêndulo invertido (com nome *pendulo invertido*) e o carro (com nome *carro*). O bloco do pêndulo encontra-se conectado ao carro através de uma junta *Revolute* que permite um movimento rotacional. Esta junta está conectada a um sensor (*Joint Sensor2*) para medir o ângulo e a velocidade angular. O bloco *carro* está conectado a uma junta *Prismatic* que permite o movimento translacional horizontal do carro. Esta junta está ligada a um actuador (*Joint Actuator Força no carro*) onde é aplicada a força que controla a velocidade e a direcção do movimento do carro. A mesma junta está ligada a um sensor (*Joint Sensor1*) para medir a posição e velocidade do carro.

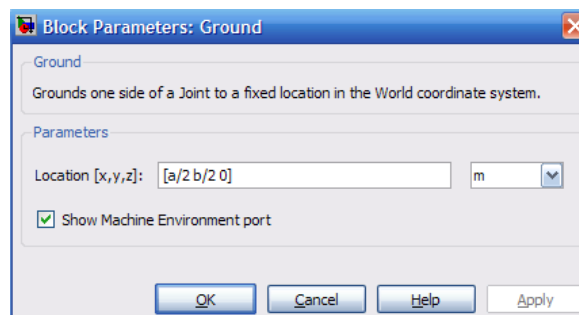


Figura 5-31

O bloco *Ground* (ver Figura 5-31) encontra-se deslocado da origem das posições do AVS. A sua posição ($[a/2 \ b/2 \ 0]$) corresponde a um deslocamento igual a metade do comprimento e da largura do carro.

O bloco *carro* encontra-se definido na Figura 5-32. Pode-se verificar que a massa do carro (M) e a matriz de Inércia (I) estão definidas como variáveis no ambiente de trabalho. A porta “CS1” encontra-se no mesmo ponto no AVS que o bloco *Ground* anterior (visto estar deslocada de $[0\ 0\ 0]$ do bloco adjacente). A porta “CS2” está em condições semelhantes, mas encontra-se no mesmo ponto que a porta “CS2” do bloco *pendulo invertido*. O ponto “CG” (centro de massa) encontra-se no centro do carro ($[a/2\ b/2\ c/2]$). Todos os outros pontos definidos servem, apenas, para dar a forma paralelepípedica ao carro. Assim, estes encontram-se definidos de modo a cada um deles formar um vértice de um paralelepípedo. Pode-se observar que de “CS3” a “CS6” tem-se um quadrado definido e de “CS7” a “CS10” tem-se este mesmo quadrado definido deslocado dos outros pontos apenas da altura do carro (c).

Nota: Neste caso verifica-se uma boa aplicação da translação de pontos, já que temos pontos deslocados do AVS, do bloco adjacente e de pontos dentro do mesmo bloco.

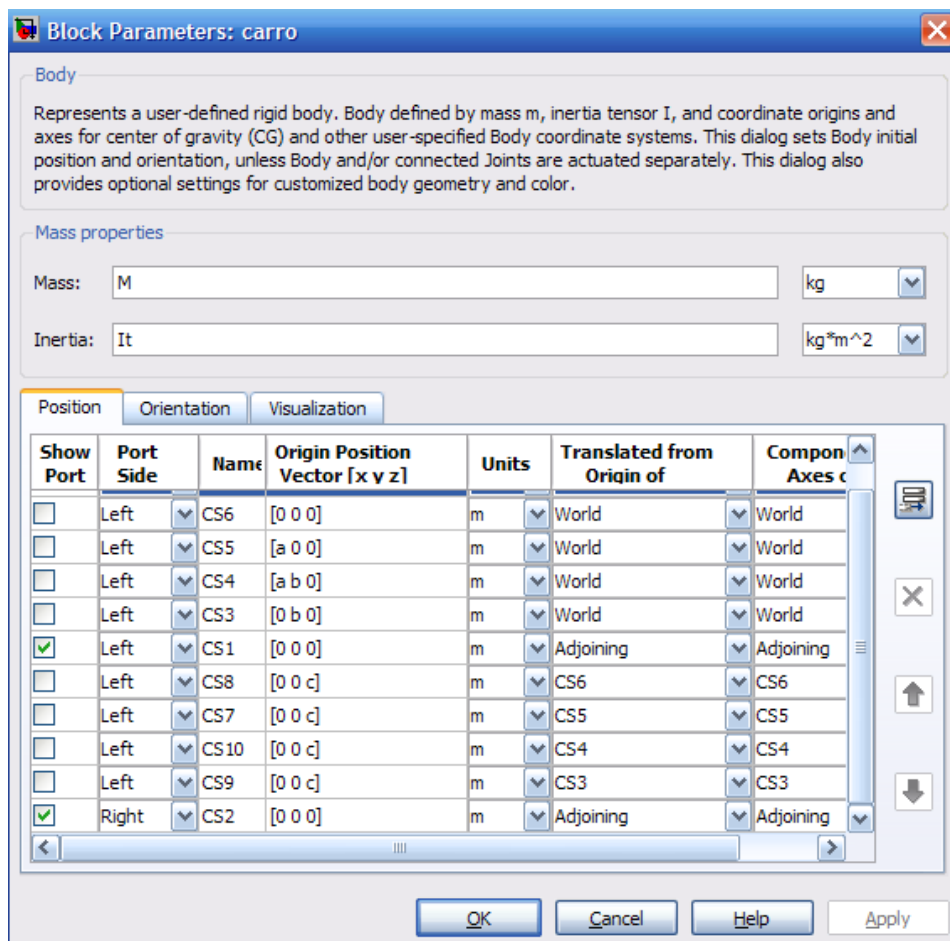


Figura 5-32

Para o bloco *pendulo invertido* tem-se a mesma situação de variáveis em ambiente de trabalho para a massa e a matriz de inércia. Os pontos iniciais de ligação ao AVS estão definidos todos em relação ao AVS (ver Figura 5-33). Visto o pêndulo estar apenas conectado através de um ponto ao carro, a porta “CS2” é a única visível, sendo que “CG” (centro de massa) e “CS1” são definidos apenas para dar forma ao corpo. O ponto de ligação de “CS2” do *carro* com “CS2” do *pendulo invertido* (que é necessariamente o mesmo) encontra-se no centro e em cima do carro ($[a/2 \ b/2 \ c]$). O ponto “CS1” é o extremo do pêndulo e obtém-se acrescentando a “CS2” a altura do pêndulo. “CS1” tem como vector $[a/2 \ b/2 \ c+cp]$. O centro de massa do pêndulo - “CG” – tem como vector $[a/2 \ b/2 \ c+cp/2]$.

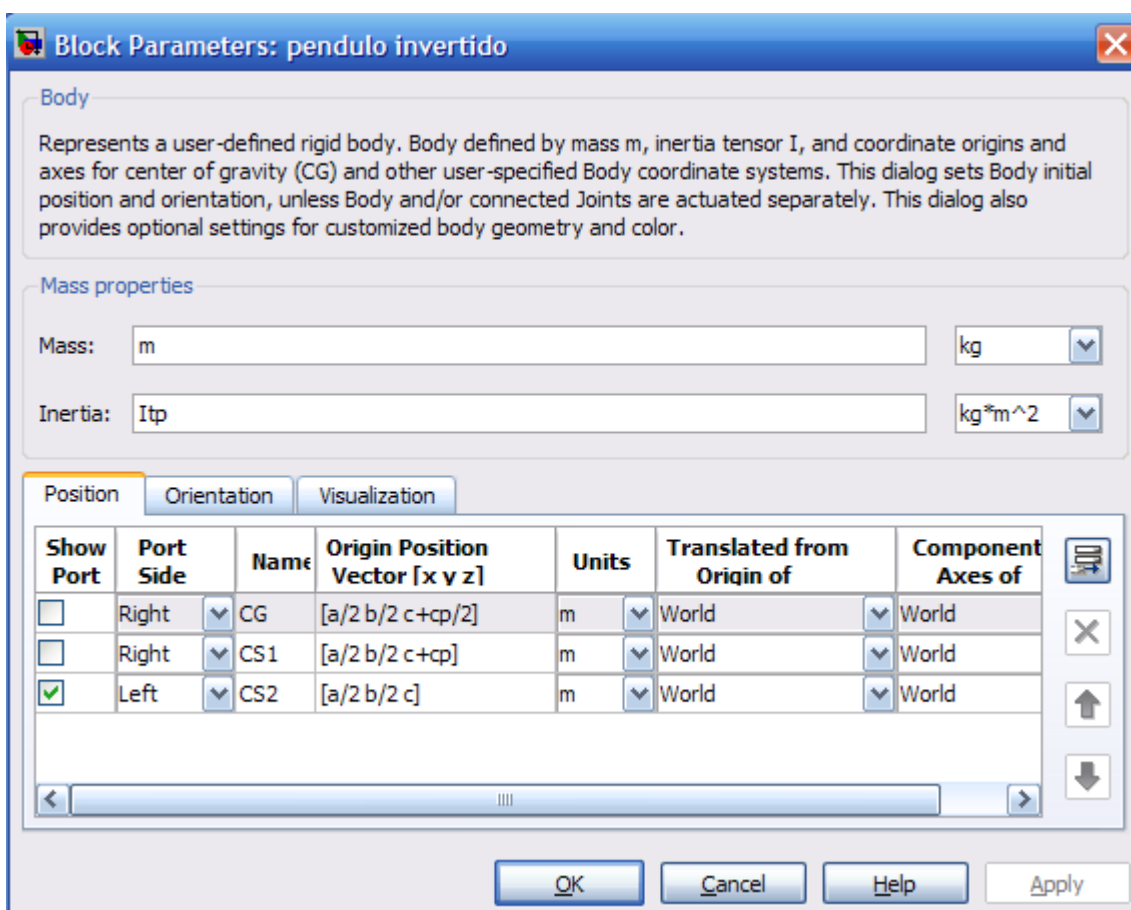


Figura 5-33

A junta *Revolute* tem as propriedades definidas na Figura 5-34. De notar o facto de ter movimento de rotação segundo o eixo do y e de ter um sensor conectado.

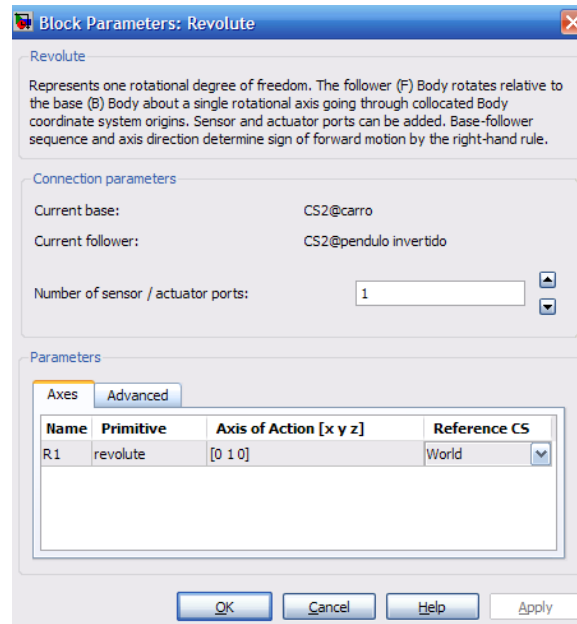


Figura 5-34

A junta *Prismatic* tem conectado um actuador (*Joint Actuator Força no carro*) e um sensor (*Joint Sensor1*). O movimento que esta permite é movimento translacional segundo o eixo do x. As propriedades do bloco estão definidas na Figura 5-35.

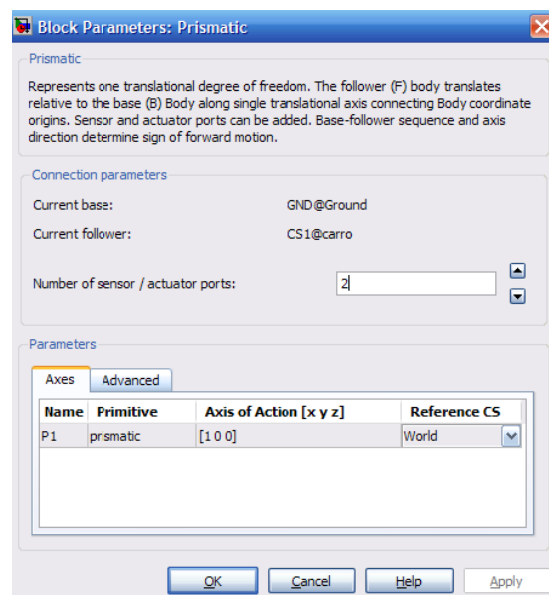


Figura 5-35

Nas duas figuras seguintes pode-se verificar as propriedades dos blocos dos sensores (*Joint Sensor1* - Figura 5-37; e *Joint Sensor2* - Figura 5-36). Seleccionou-se apenas a checkbox das variáveis que se pretendia medir e retirou-se o visto da checkbox que dizia "Output parameters as one signal".

A mudança das medidas a seleccionar nas propriedades do bloco *Joint Sensor* ocorre automaticamente, dependendo do tipo de junta a que se encontra ligado.

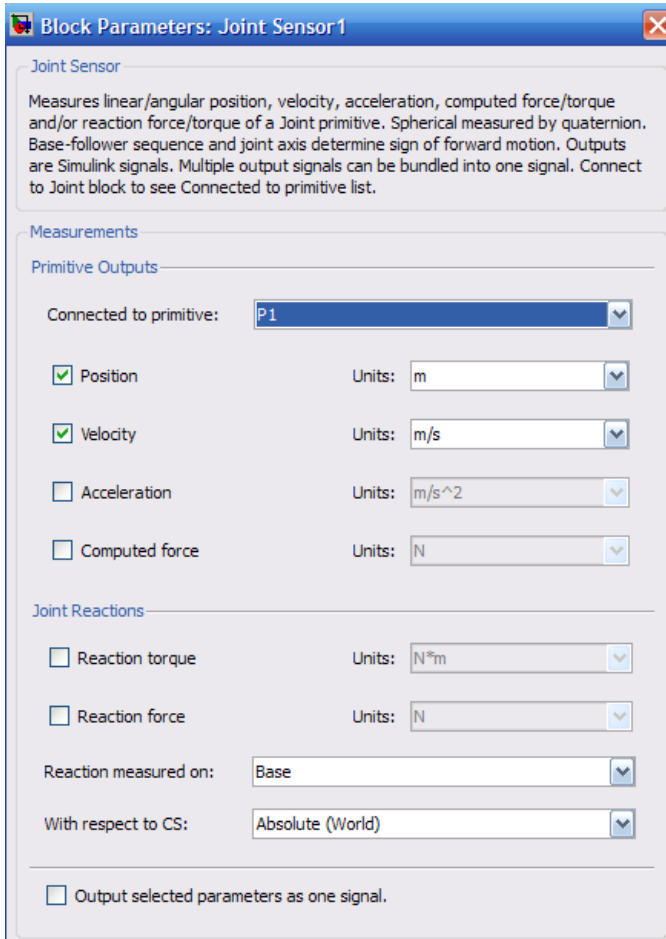


Figura 5-37

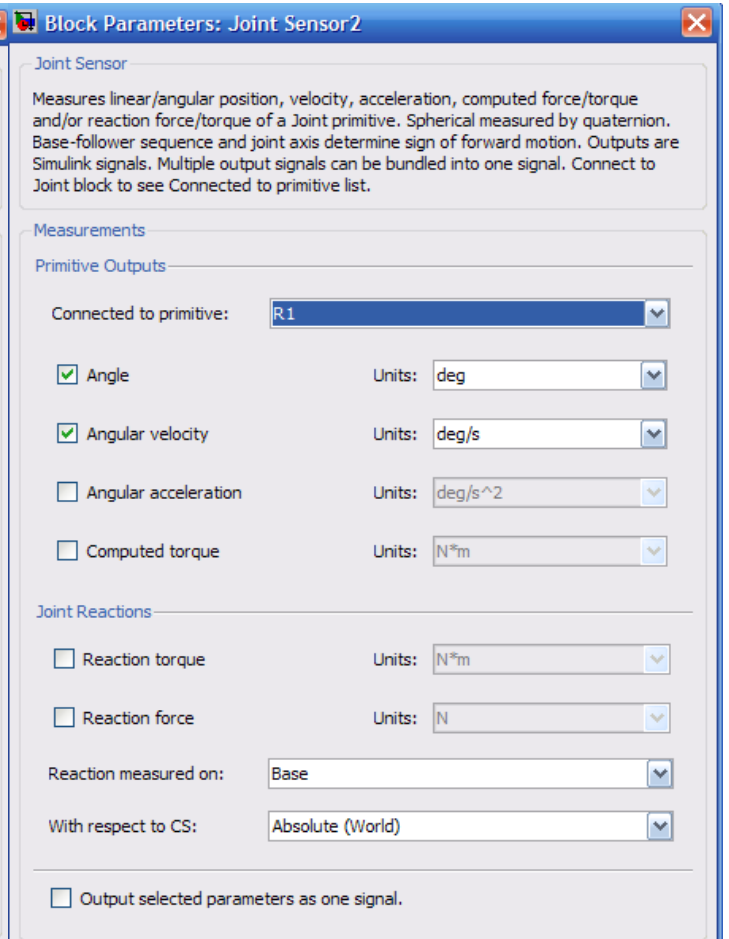


Figura 5-36

O bloco *Joint Actuator* tem as suas definições na Figura 5-38 que consiste, apenas, em definir que se actua com “Generalized Forces” em “N” (Newton).

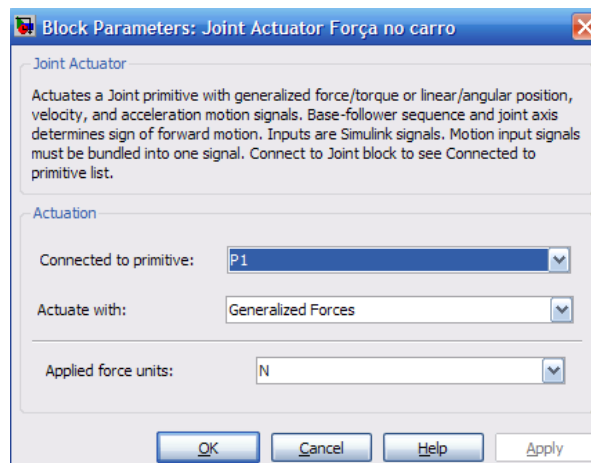


Figura 5-38

O bloco *Environmental Machine*, que a única mudança que sofreu das suas predefinições, foi ter-se alterado o campo “Default Body Geometries” no separador “Visualization” para “Convex Hull from body CS locations” (Figura 5-39).

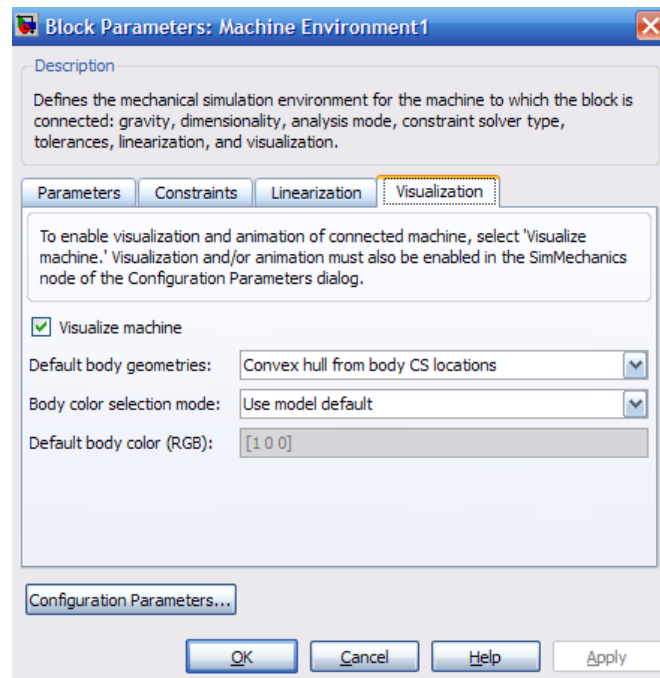


Figura 5-39

Por fim, o bloco “*Joint Initial Condition*”, que se encontra definido na Figura 5-40. Neste caso, este bloco vai deslocar a junta onde se encontra conectado de 10° com a vertical. A velocidade mantém-se inalterada.

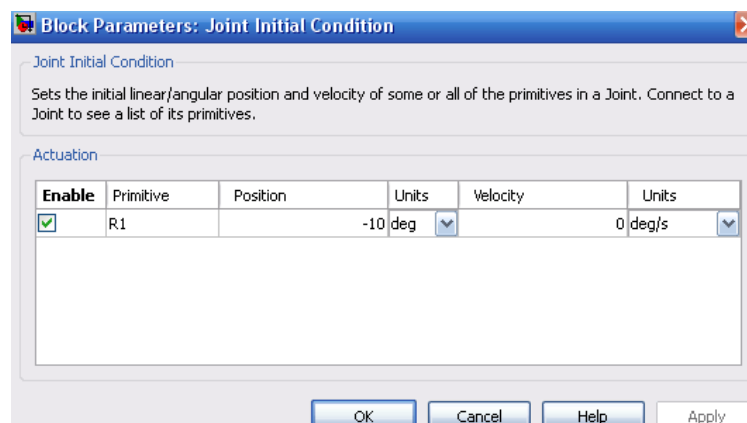


Figura 5-40

Na Figura 5-41 encontra-se o sistema de eixos do conjunto carro mais pêndulo:

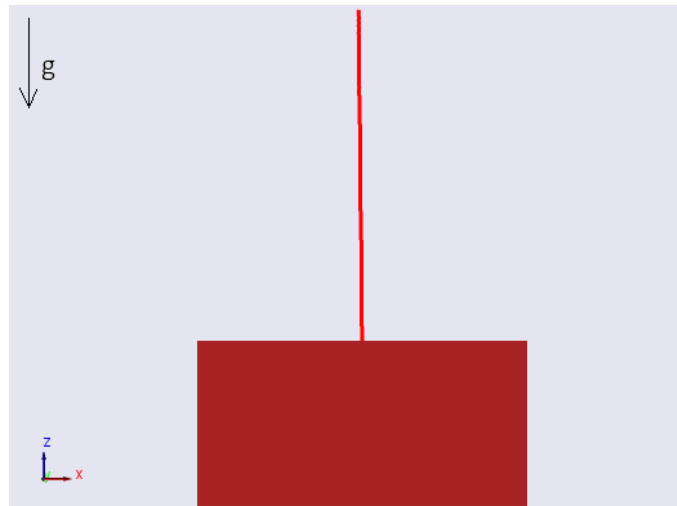


Figura 5-41 – Esquema de eixos

5.2.3 Especificações da dinâmica do Pêndulo Invertido

As especificações da dinâmica para o regime transitório do pêndulo são:

- Tempo de subida $< 0,5$ seg;
- Percentagem de Sobreelevação (“Overshoot”) $< 4\%$ (definidos tendo em conta o movimento máximo do pêndulo, entre -90° e 90° . É necessário obter uma sobreelevação inferior a $7,2^\circ$);

Para o regime estacionário do pêndulo:

- Referência (do ângulo) = 0° ;
- Tempo de estabelecimento < 1 segundos;
- Erro em regime estacionário $< \approx 3^\circ$ (tolerância da posição vertical do pêndulo);

Esta dissertação tem como principal objectivo o desenvolvimento de modelos virtuais através da ferramenta “SimMechanics” e a sua implementação em modelos de controlo. Para determinar os parâmetros dos controladores a aplicar neste sistema utilizaram-se técnicas de controlo de sistemas lineares. Neste caso, foi necessário determinar o modelo linear que representasse a dinâmica do objecto de controlo.

Para a obtenção do modelo linear do sistema no espaço de estados consultou-se a dissertação [14] e o problema 3.7 do livro [12].

Sendo: $x_1 = x$	posição do carro
$x_2 = \dot{x}$	velocidade do carro
$x_3 = \theta$	ângulo do pêndulo com a vertical
$x_4 = \dot{\theta}$	velocidade angular do pêndulo
$M=0.5$ kg;	massa do carro
$m=0.2$ kg;	massa do pêndulo
$cp=0.6$ m;	altura do pêndulo
$g=9.8$ m/s ² ;	aceleração da gravidade
$l=cp/2$ m;	distância do pêndulo ao seu centro de massa
$I=0.06$ kg*m ² ;	momento de inércia do pêndulo
$b=0.1$ N/m/s	fricção do carro

Substituindo nas equações do movimento, vem:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I + m \cdot l^2) \cdot b}{I \cdot (M + m) + M \cdot m \cdot l^2} & \frac{m^2 \cdot g \cdot l^2}{I \cdot (M + m) + M \cdot m \cdot l^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-(m \cdot l \cdot b)}{I \cdot (M + m) + M \cdot m \cdot l^2} & \frac{m \cdot g \cdot l \cdot (M + m)}{I \cdot (M + m) + M \cdot m \cdot l^2} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I + m \cdot l^2}{I \cdot (M + m) + M \cdot m \cdot l^2} \\ 0 \\ \frac{m \cdot l}{I \cdot (M + m) + M \cdot m \cdot l^2} \end{bmatrix} \cdot u \quad (1)$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot u \quad (2)$$

O diagrama de blocos desta representação encontra-se esquematizado na Figura 5-42.

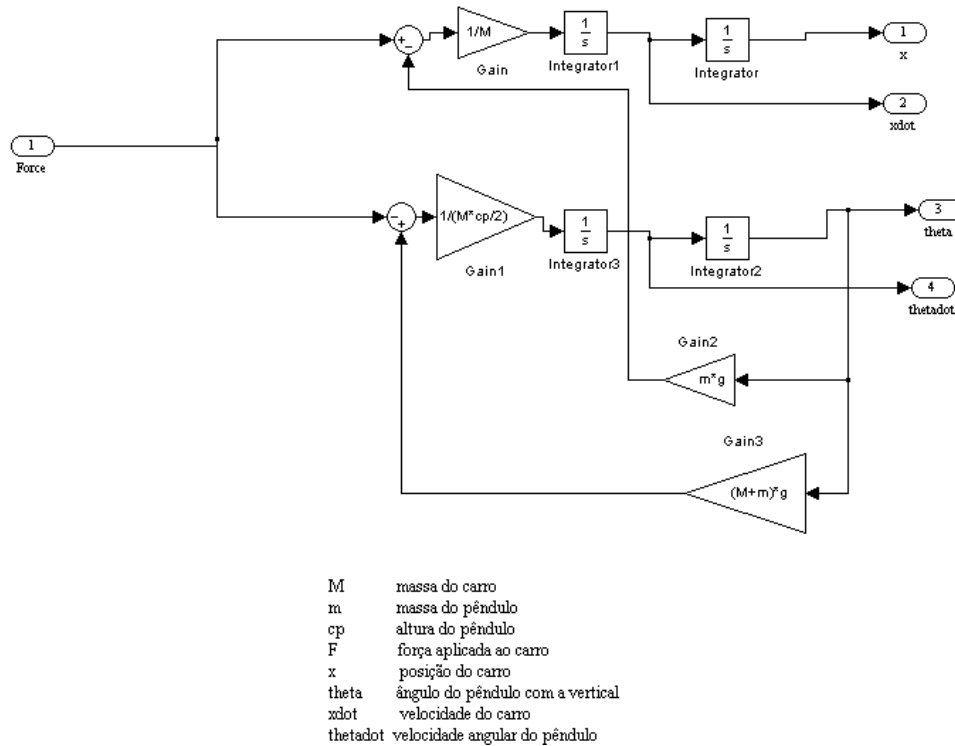


Figura 5-42 – Modelo linear do Pêndulo Invertido – representação em espaço de estados

De notar que as variáveis *theta* e *thetadot* do diagrama de blocos representado em cima são ambas em radianos.

Todos os comandos em Matlab necessários se encontram no Anexo 2.

5.2.4 Problemas na visualização da animação do modelo do Pêndulo Invertido

Como já se referiu em cima, as matrizes de Inércia para o bloco do carro e o bloco do pêndulo são, respectivamente:

$$I_t = [(M/12)*(b^2+c^2) \quad 0 \quad 0; \quad 0 \quad (M/12)*(a^2+c^2) \quad 0; \quad 0 \quad 0 \quad (M/12)*(a^2+b^2)];$$

$$I_p = [(m/12)*(bp^2+cp^2) \quad 0 \quad 0; \quad 0 \quad (m/12)*(ap^2+cp^2) \quad 0; \quad 0 \quad 0 \quad (m/12)*(ap^2+bp^2)];$$

Devido à obtenção de valores decimais nestas matrizes, quando se corre o modelo do pêndulo invertido, cada segundo da simulação demora mais de um minuto, se se pretender visualizar a animação da Máquina.

Verificou-se que esta é uma limitação do “SimMechanics”, porque se se quiser obter resultados precisos não se pode visualizar a animação (perdendo-se demasiado tempo) e vice-versa. Na maior parte das demos inseridas no Matlab, as matrizes de Inércia vêm como uma Matriz diagonal de uns ([1 0 0; 0 1 0; 0 0 1]). Se se alterarem as matrizes de Inércia no modelo do pêndulo por matrizes diagonais de uns, a visualização da simulação já decorre a velocidade normal. Assim, para se obterem resultados precisos, teve de se alterar as opções da simulação. Em vez desta correr em modo “Normal”, passa a correr em modo “Rapid Accelerator”. O problema deste modo da simulação é que não permite a visualização da animação virtual. Deste modo, foi possível a obtenção dos dados necessários para se retirarem as conclusões precisas sobre a performance do modelo virtual do pêndulo invertido.

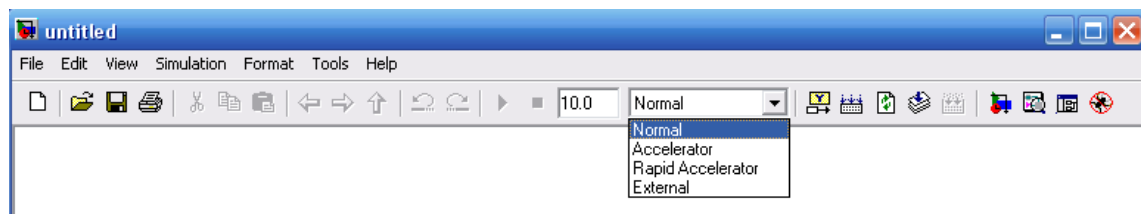


Figura 5-43 – Barra de ferramentas do modelo em Simulink com destaque para a opção da velocidade de simulação

5.2.5 Controlo do Pêndulo Invertido com modelo linear e controlador PID

O esquema de controlo do pêndulo invertido com controlador PID e o modelo linear em espaço de estados (Figura 5-42) é representado na Figura 5-44.

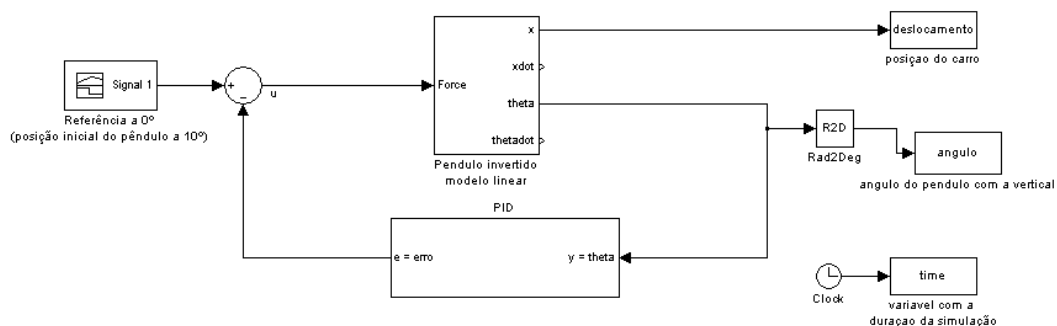


Figura 5-44 – Sistema de controlo do pêndulo invertido com controlador PID (modelo linear)

Os dados para os parâmetros do controlador PID (mesmo bloco utilizado na Figura 5-15) foram retirados de [14].

$$K_p = -60,4;$$

$$K_i = -22,6;$$

$$K_d = -60,4;$$

O bloco *Referência a 0°* é um pequeno impulso de amplitude 100, que corresponde a um deslocamento inicial de, aproximadamente, 10° da posição inicial do pêndulo (ver Figura 5-45).

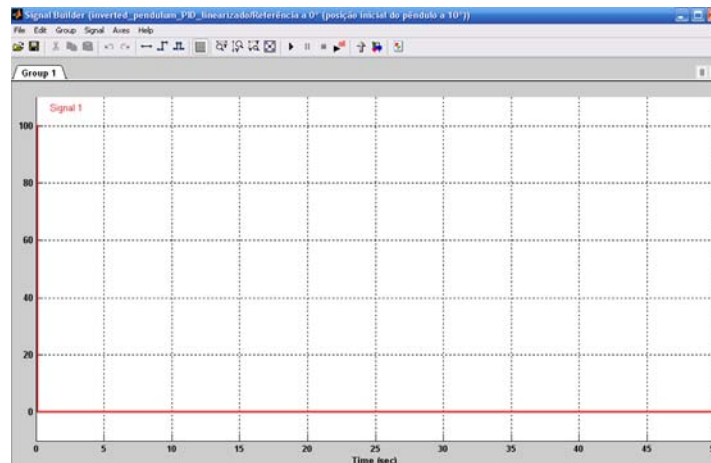


Figura 5-45

Executou-se a simulação da malha fechada de controlo na Figura 5-44 ao longo de 50 segundos e os resultados encontram-se esquematizados no gráfico em baixo.

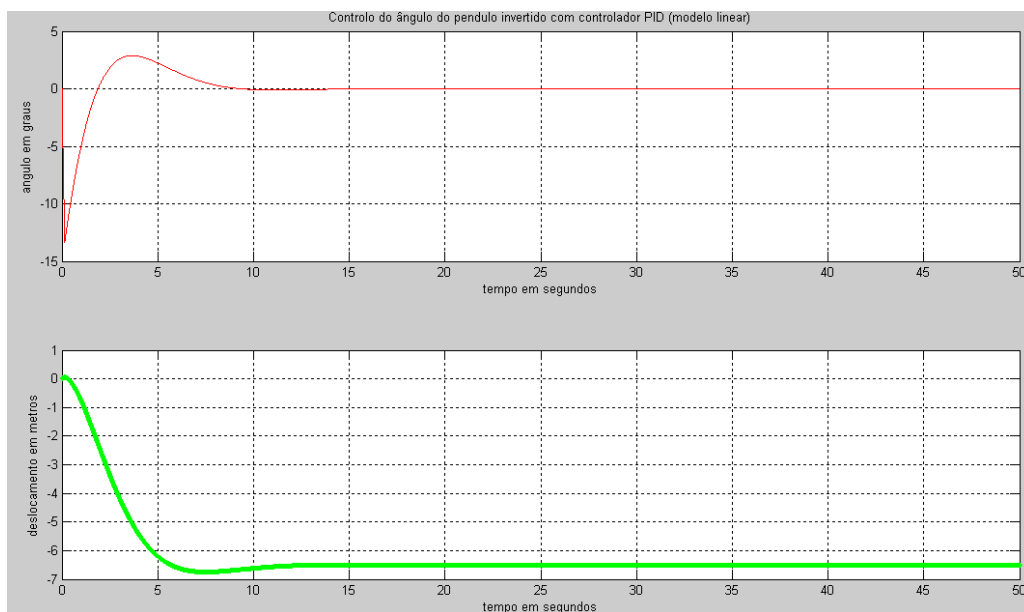


Figura 5-46

Como se pode verificar pelo gráfico, o controlador PID consegue manter o ângulo a rondar os 0°, mas a posição do carro sofre um desvio de, aproximadamente, sete metros, tendendo a estabilizar após 10 segundos de simulação.

5.2.6 Controlo do Pêndulo Invertido no Laboratório Virtual

Controlador PID

Para o modelo virtual, utilizou-se o mesmo diagrama de blocos da Figura 5-44 mas substituiu-se o bloco do modelo linear do pêndulo invertido pelo bloco realizado em Laboratório Virtual (Figura 5-30). Os valores dos parâmetros do bloco *PID* mantiveram-se os mesmos que os usados no modelo linear. O bloco da referência é apenas uma constante nula, visto que o bloco do modelo virtual já possui um bloco que coloca a posição inicial do pêndulo deslocado de 10° da vertical. Retirou-se também o bloco *Rad2Deg*, porque o modelo em Laboratório Virtual já fornece o ângulo em graus. Assim, o esquema em Simulink encontra-se na Figura 5-47.

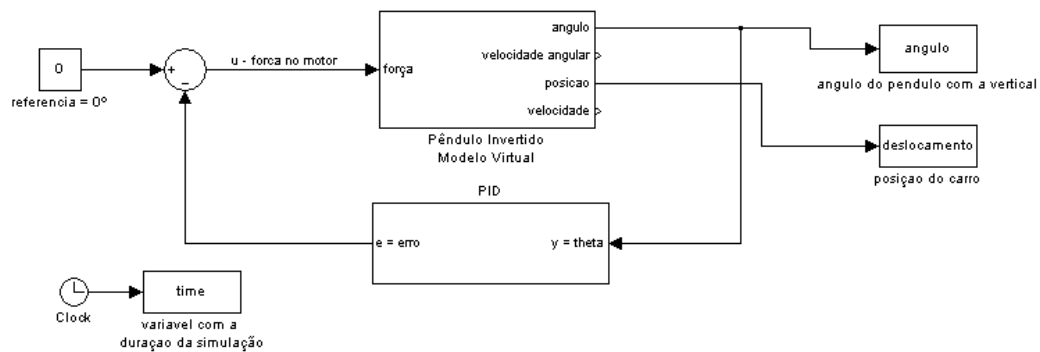


Figura 5-47 - Diagrama de blocos do modelo virtual do controlo do pêndulo invertido com controlador PID

Executou-se a simulação ao longo de 50 segundos e os resultados obtidos encontram-se esquematizados na Figura 5-48 juntamente com os resultados da simulação anterior com o modelo linear.

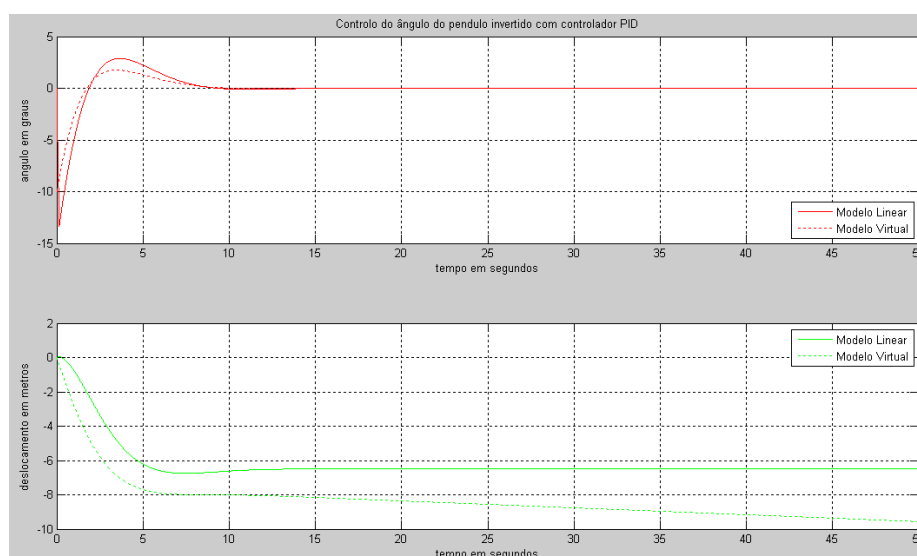


Figura 5-48

Pode-se verificar que os resultados são semelhantes aos obtidos com o modelo linear o que leva à conclusão que o modelo virtual realizado representa adequadamente a dinâmica do pêndulo invertido.

Apesar do controlo do pêndulo ser efectuado dentro das especificações desejadas, o carro mover-se indefinidamente não é propício para um modelo real, visto a calha onde este se move não ser infinita.

Assim, conclui-se que o controlador PID não serve para controlar o sistema mecânico do pêndulo invertido. Optou-se, então, pelo controlo através da realimentação por variáveis de estado.

Controlador por realimentação das variáveis de estado

Para este tipo de controlo é necessário calcular a matriz de ganho K . O método para o cálculo de K utilizado foi o método via implantação de pólos. Este método encontra-se descrito em [11].

Para o cálculo desta matriz é necessário verificar uma série de condições acerca do sistema a controlar. Estas condições foram verificadas no Anexo 2 do Matlab.

Após a verificação das condições, procedeu-se ao cálculo da matriz de ganho K . Através das condições requeridas, obtiveram-se dois pólos dominantes em $-4 \pm 3.9j$. Como K é uma matriz de ordem 4, os outros dois pólos não dominantes escolhidos foram um pólo duplo em -20 (ver [11]).

Através da fórmula de Ackermann, calculou-se a matriz K com os seguintes valores (ver [11]):

$$K = [-290.4485 \quad -103.6213 \quad 297.3826 \quad 53.4510]$$

NOTA: Todos estes cálculos se encontram no Anexo 2.

Após o cálculo da matriz de ganho K criou-se um bloco em Simulink com os ganhos de K (Figura 5-49).

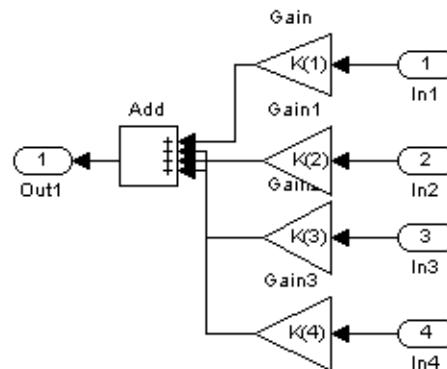


Figura 5-49 - Diagrama de blocos do controlador por realimentação das variáveis de estado (matriz de ganho K)

Este bloco K foi inserido na malha de controlo com o modelo virtual do pêndulo invertido (Figura 5-50).

A referência necessária para o controlo por variáveis de estado deixa de ser 0° , visto que não há apenas a realimentação de uma variável, neste caso, o ângulo. Deixa de se poder comparar o ângulo com zero visto haver realimentação de todas as variáveis de estado.

Assim, foi necessário determinar uma nova referência. Calculou-se uma nova constante como referência, diferente de zero, pelo método definido no site [8] que se encontra em Anexo 2.

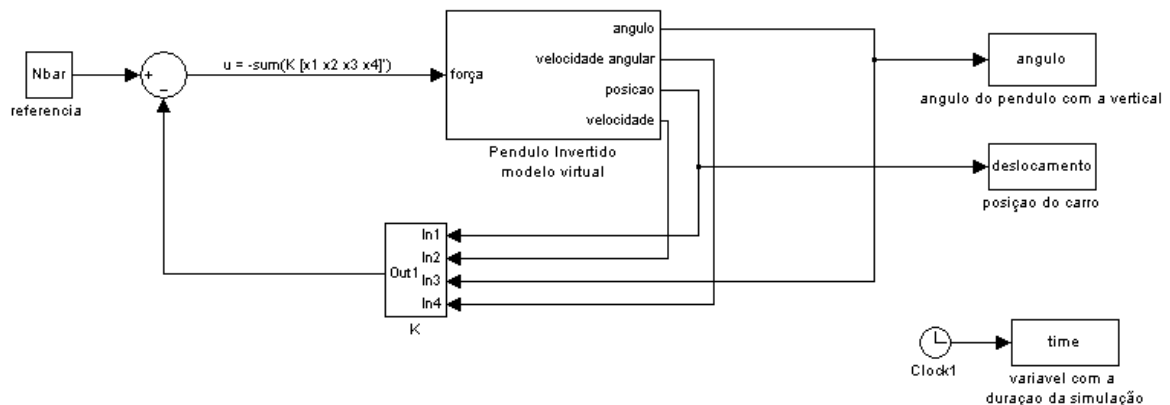


Figura 5-50 - Diagrama de blocos da malha de controlo do modelo virtual do Pêndulo Invertido com realimentação de variáveis de estado

Depois de se executar a simulação durante 500 segundos, os resultados estão esquematizados na Figura 5-51. De notar que a escala do ângulo foi reduzida entre -2° e 2° para melhor visualização de resultados, mas o valor inicial do ângulo é 10° devido ao bloco *Joint Initial Condition* presente no modelo virtual.

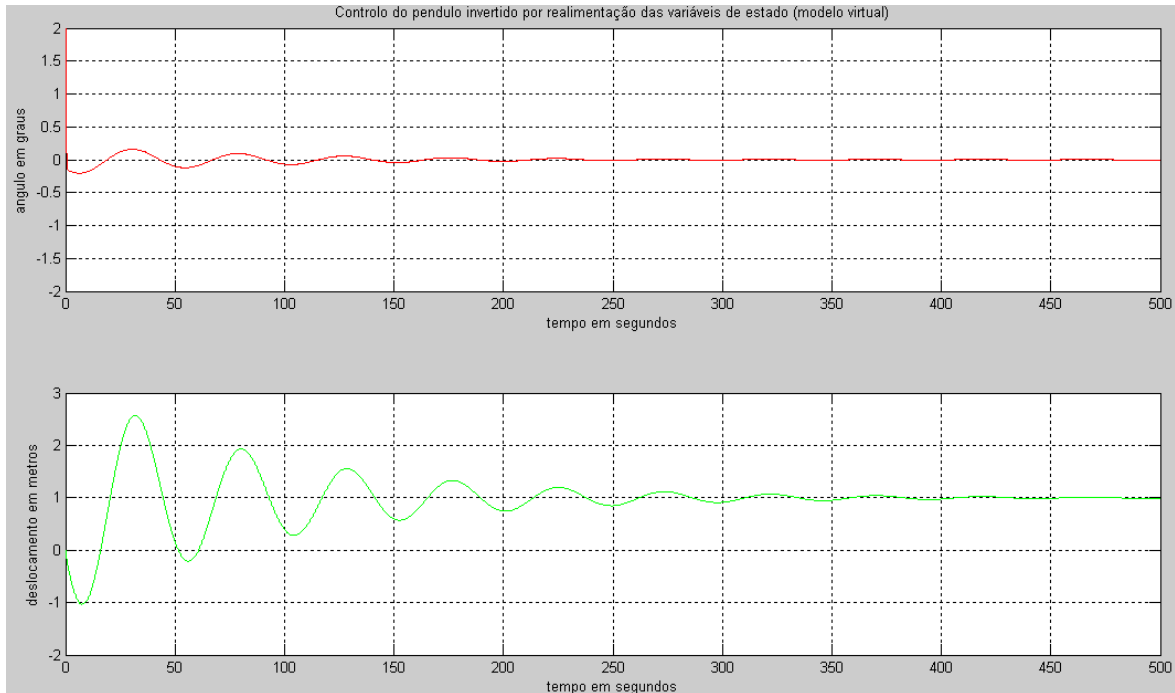


Figura 5-51

Como se pode verificar pelo gráfico em cima já se conseguiu controlar o ângulo e a posição do carro já não aumenta indefinidamente. Verifica-se que as oscilações correspondem com um comportamento esperado de um pêndulo invertido e que o modelo virtual se adequa à dinâmica do mesmo.

As especificações, com a realimentação através das variáveis de estado, foram todas cumpridas.

Para verificar o desempenho do modelo virtual com a introdução de perturbações optou-se por usar uma malha de controlo idêntica à da Figura 5-50. A diferença introduzida foi no bloco *pendulo Invertido modelo virtual* onde se acrescentou um bloco *Body Actuator* para aplicar as perturbações no pêndulo. O bloco *Body Actuator* actua através de forças e através de um vector de três elementos (cada um correspondendo a um eixo). Como o que se pretende é actuar apenas no eixo dos x, o sinal com a perturbação é inserido na primeira posição do vector, tendo as outras coordenadas o valor zero. O modelo virtual com as perturbações encontra-se esquematizado na Figura 5-52.

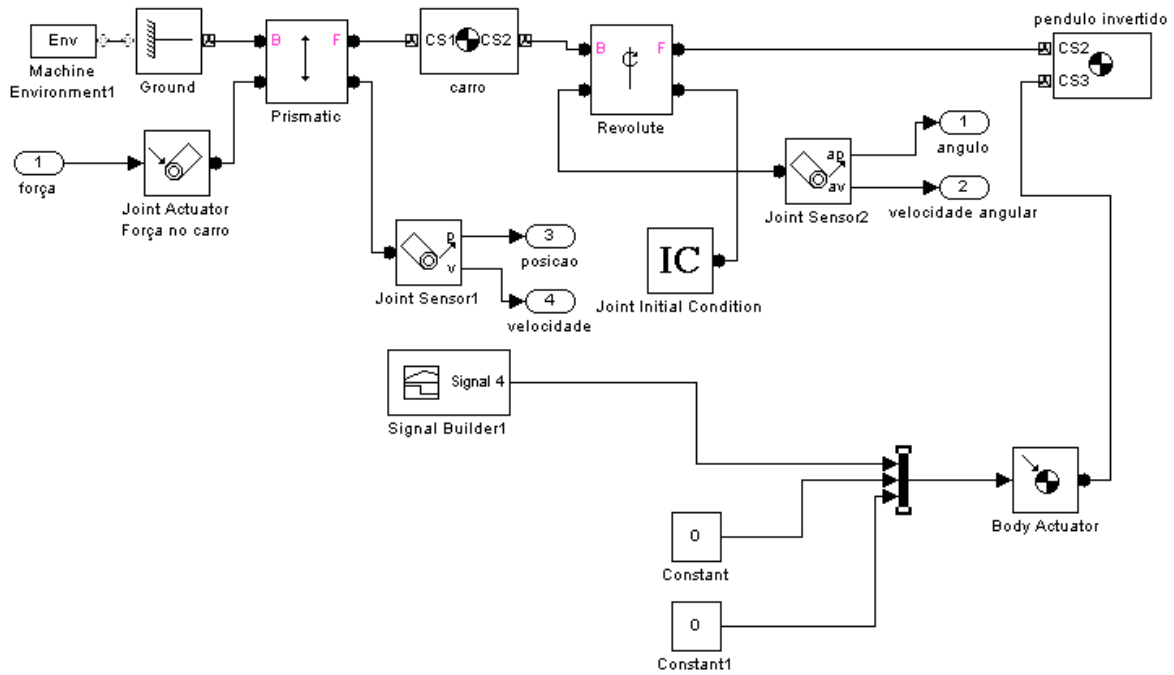


Figura 5-52 - Modelo virtual com perturbações

O bloco *Signal Builder1* tem como diagrama a Figura 5-53. É constituído por um impulso com valor negativo. O valor do impulso corresponde a uma perturbação de aproximadamente -8° na posição do ângulo.

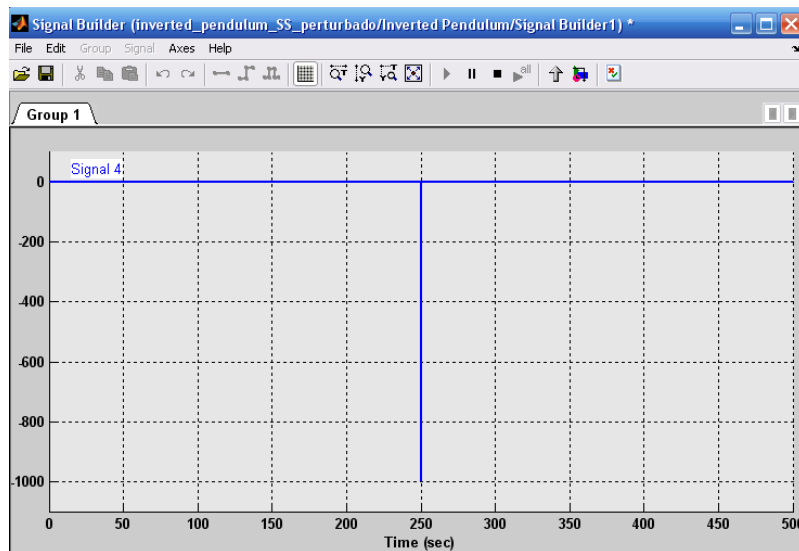


Figura 5-53

O gráfico seguinte foi o obtido após realização da simulação de 500 segundos. De referir, mais uma vez, que a escala do ângulo foi reduzida entre -2° e 2° para melhor visualização dos resultados, e, mais uma vez, a posição inicial do pêndulo é 10° .

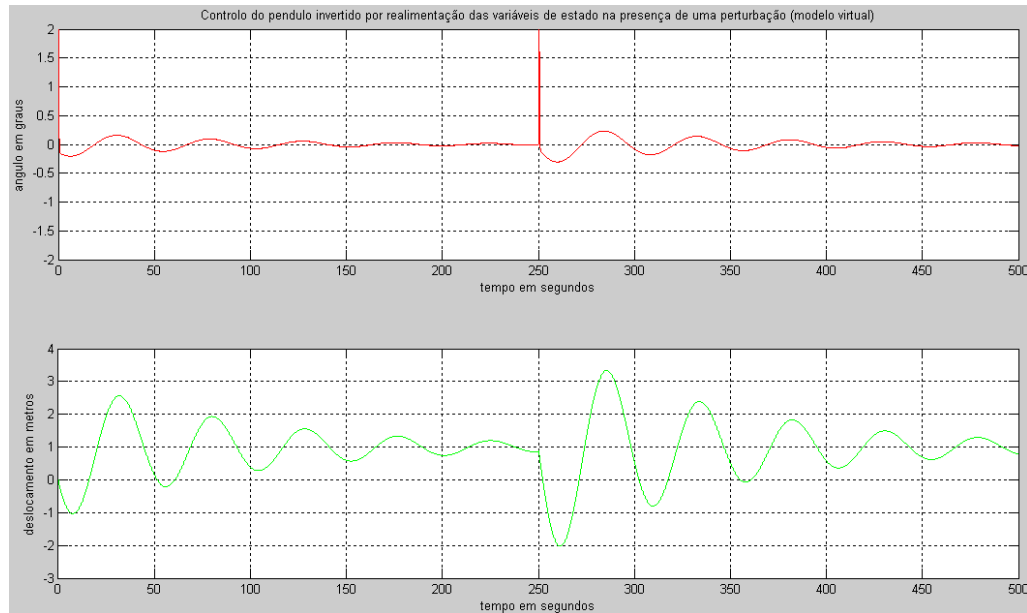


Figura 5-54

Verifica-se que o controlador por realimentação das variáveis de estado consegue estabilizar o pêndulo invertido na presença de uma perturbação em direcção diferente da posição inicial do pêndulo. Assim se conclui que o modelo virtual realizado representa bem o sistema real e pode ser utilizado em testes de vários algoritmos de controlo.

5.3 Controlo da suspensão de um autocarro

5.3.1 Introdução

O desenho de um controlador para uma suspensão automática de um autocarro é um problema interessante e desafiador. O sistema de suspensão para cada roda é individual por isso considera-se o sistema de um quarto do autocarro + suspensão + uma roda. Este sistema encontra-se esquematizado na Figura 5-55.

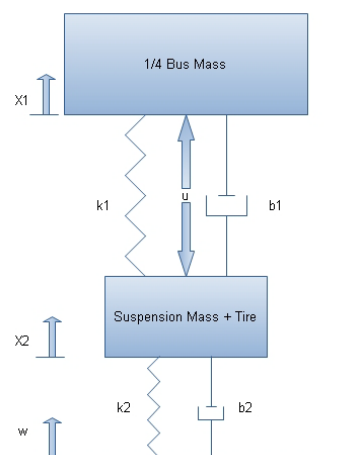


Figura 5-55

Um bom sistema deve permitir que o autocarro ande sobre deformações na estrada e os passageiros praticamente não sintam estas perturbações (diminuindo a amplitude dessas oscilações e dissipando-as o mais rápido possível). Como a deformação do pneu (X_2-W) é muito pequena, a distância que interessa medir será X_1-X_2 (*posição*). Os dados referentes a este problema (massa, comprimento, altura e largura) se encontram no Anexo 3 com a seguinte estrutura:

```
R=0.4; %Raio do roda (m)
m=320; %Massa do conjunto suspensão+roda (Kg)
Ir=[(2*m*R^2)/5 0 0; 0 (2*m*R^2)/5 0; 0 0 (2*m*R^2)/5];
%matriz de Inércia da roda+suspensao (correspondente a uma esfera)
M=2500; %Massa do bloco que representa um quarto de autocarro (Kg)
a=8/2; %Comprimento do bloco que representa um quarto de autocarro (m)
b=3/2; %Largura do bloco que representa um quarto de autocarro (m)
c=3.5; %Altura do bloco que representa um quarto de autocarro (m)
It=[(m/12)*(b^2+c^2) 0 0; 0 (m/12)*(a^2+c^2) 0; 0 0 (m/12)*(a^2+b^2)];
%matriz de Inércia do bloco que representa um quarto de autocarro
k1=80000; % Constante de elasticidade da mola da suspensão (N/m)
k2=500000; % Constante de elasticidade da mola do pneu (N/m)
b1 = 350; % Constante de atrito do amortecedor da suspensão (N.s/m)
b2 = 15020;% Constante de atrito do amortecedor do pneu (N.s/m)
```

Os dados das massas e constantes de elasticidade e atrito foram retirados do site [2].

Os dados do comprimento, largura e altura do bloco de autocarro, e o raio da roda e pneu, foram arbitrados com valores razoáveis de um modelo real desta simulação.

Os valores das Matrizes de Inércia foram retirados do livro [16] referentes à Matriz de Inércia de um paralelepípedo e de uma esfera.

5.3.2 Modelo virtual da suspensão do autocarro

A Figura 5-56 representa o modelo virtual em SimMechanics do conjunto mecânico do sistema de roda – suspensão – $\frac{1}{4}$ de autocarro.

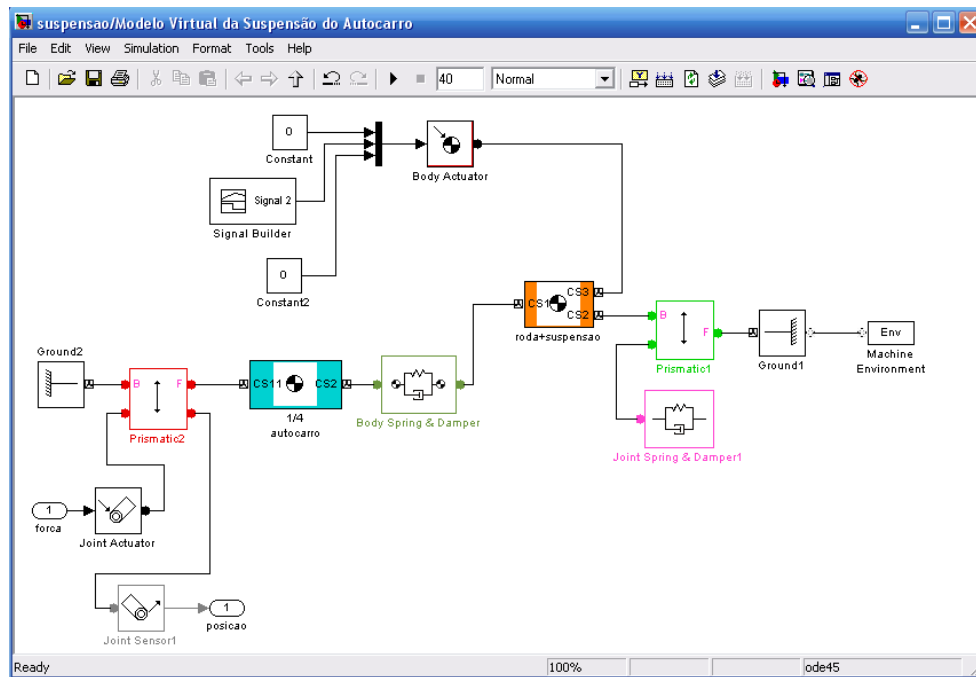


Figura 5-56

Na Figura 5-57 é representado o bloco do respectivo modelo virtual representado na Figura 5-56.

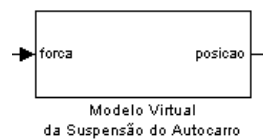


Figura 5-57

Pela Figura 5-56 verifica-se que o *Modelo Virtual da Suspensão do Autocarro* possui dois blocos *Body*. O bloco *roda+suspensão* encontra-se ligado a uma junta *Prismatic1*, a um bloco *Body Actuator* e a um bloco *Body Spring & Damper*. O outro bloco *Body* (*1/4 autocarro*) encontra-se ligado ao mesmo bloco *Body Spring & Damper* anterior e a uma junta *Prismatic2*.

O bloco *roda+suspensão* inclui o conjunto da estrutura da roda, do pneu e da suspensão em si. Este bloco tem as suas propriedades definidas na Figura 5-58. Verifica-se que a massa m e a matriz de Inércia I_r estão definidas em Anexo 2. I_r é uma matriz de Inércia de uma esfera que precisa, apenas, da definição do seu raio (R).

Assim o CG do corpo encontra-se deslocado do AVS através de um vector $[0 \ R \ 0]$ (centro da esfera).

Tem-se o ponto $CS3$ que é onde é aplicada a perturbação através do bloco *Body Actuator*. (a perturbação é por baixo da roda o que faz com que $CS3$ seja no ponto $[0 \ 0 \ 0]$ do AVS).

CS2 é o ponto onde se encontra ligada a junta *Prismatic1* que permite um movimento translacional. Esta junta simula o movimento do pneu em si e encontra-se, também, no ponto $[0\ 0\ 0]$ do AVS.

Por fim, o ponto *CS1* é onde se encontra conectado o bloco *Body Spring & Damper*. Este ponto (onde se aplicam as forças de elasticidade da mola e atrito do amortecedor da suspensão) fica situado no cimo do conjunto pneu+suspensão, encontrando-se deslocado de $[0\ 2R\ 0]$ do AVS.

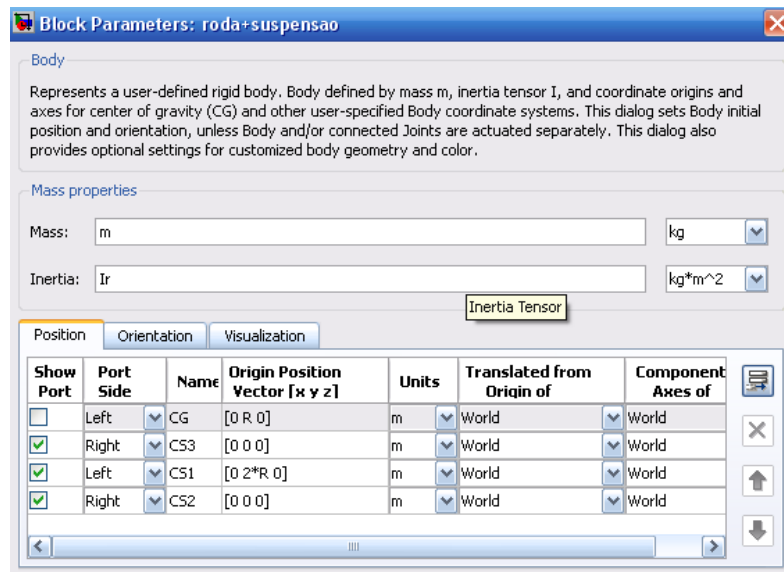


Figura 5-58

O bloco *Ground1* simula a estrada, e por isso encontra-se no ponto $[0\ 0\ 0]$ do AVS. Os parâmetros deste bloco encontram-se na Figura 5-59 (de notar a checkbox activa para o parâmetro “Show Machine Environment port”).



Figura 5-59

O bloco *Machine Environment* possui no separador “Parameters” a gravidade definida no eixo dos *y* (vector $[0\ -9.81\ 0]$). No separador “Visualization” a checkbox do “Visualize Machine” está activa e o campo “Default body geometries” encontra-se seleccionado em “Equivalent ellipsoid from mass properties” (Figura 5-60).

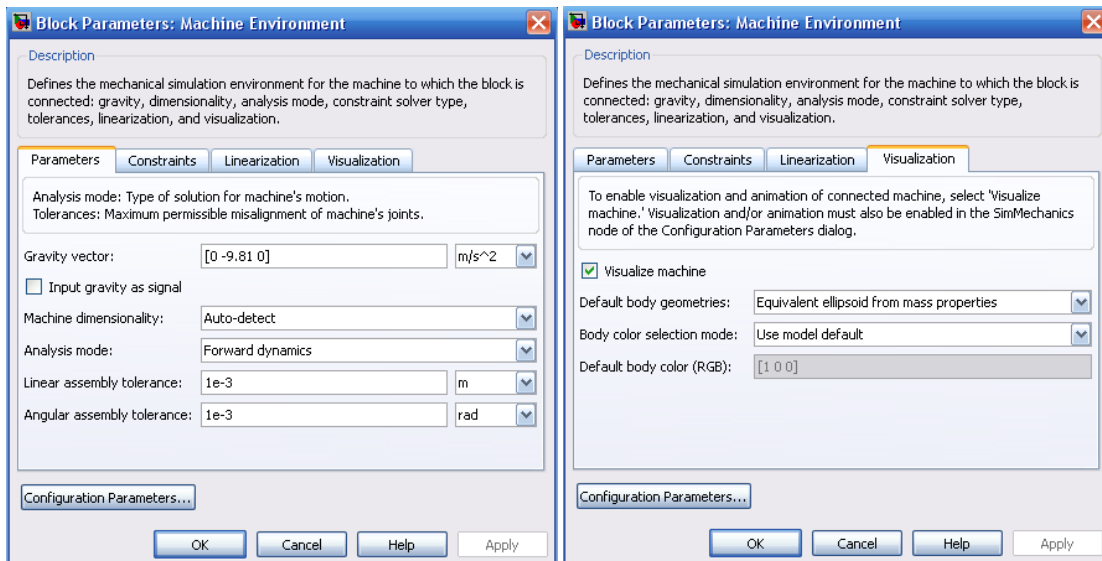


Figura 5-60

Entre o bloco *Ground1* e o bloco *roda+suspensão* está uma junta *Prismatic1*. Esta junta serve para simular o movimento de elasticidade da mola e atrito do amortecedor do pneu. Assim esta junta permite um movimento com um grau de liberdade (segundo eixo dos y) (Figura 5-61).

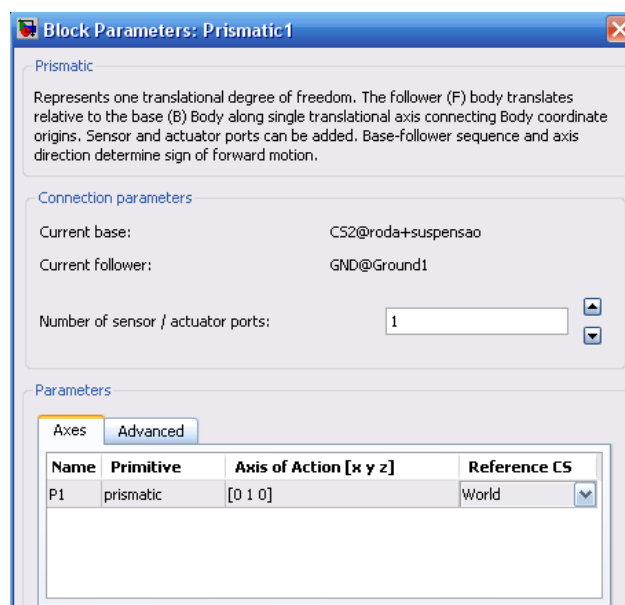


Figura 5-61

A esta junta encontra-se ligado um actuador *Joint Spring & Damper1*. É este bloco que vai actuar na junta, introduzindo o movimento de elasticidade da mola e atrito do amortecedor simulando o pneu do autocarro. Como se pode ver pelos parâmetros do bloco (Figura 5-62), as variáveis introduzidas são $k2$ e $b2$ e um offset de zero metros ($x0=0$).

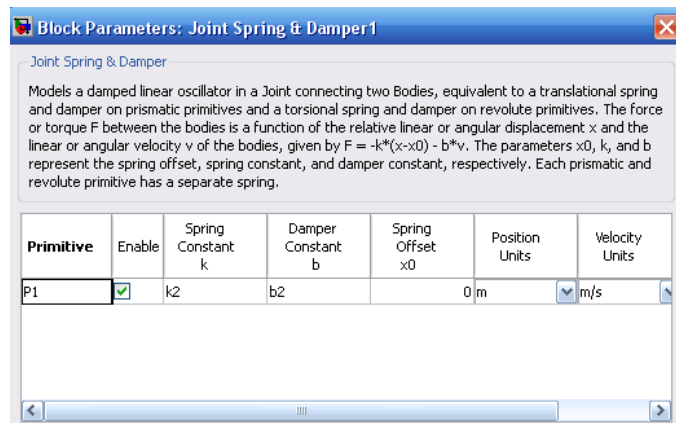


Figura 5-62

No ponto *CS3* do bloco *roda+suspensão* encontra-se aplicado um bloco *Body Actuator*. É através deste bloco que vão ser introduzidas as perturbações que ocorrem “na estrada”. Este bloco vai actuar no bloco *roda+suspensão* através de uma força modelada pelo sinal introduzido pelo bloco *Signal Builder*. As propriedades estão na Figura 5-63.

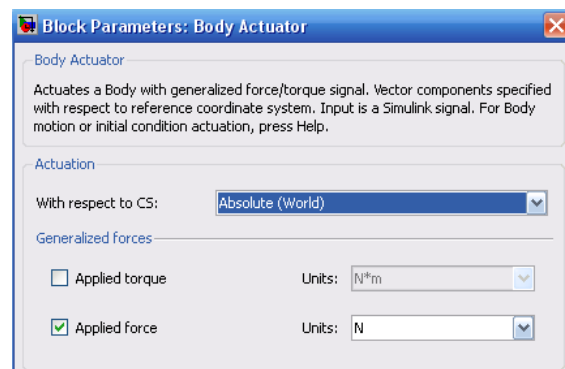


Figura 5-63

As propriedades do bloco *Signal Builder* sofrem várias alterações, conforme as simulações pretendidas. Essas alterações serão explicadas quando se executar cada uma das simulações.

O bloco *Body Spring & Damper* simula as forças devido ao movimento de elasticidade da mola e atrito do amortecedor da suspensão. Encontra-se conectado a ambos os blocos *Body* do modelo virtual. Assim, o movimento destes blocos está condicionado pelas forças que o bloco *Body Spring & Damper* introduz. As propriedades deste bloco encontram-se na Figura 5-64. As constantes k_1 , b_1 e r_0 são, respectivamente, a constante de elasticidade da mola da suspensão, a constante de atrito do amortecedor da suspensão e o tamanho natural da mola. Esta última toma o valor zero já que se considera que o tamanho da suspensão já é tido em conta no bloco *roda+suspensão*.

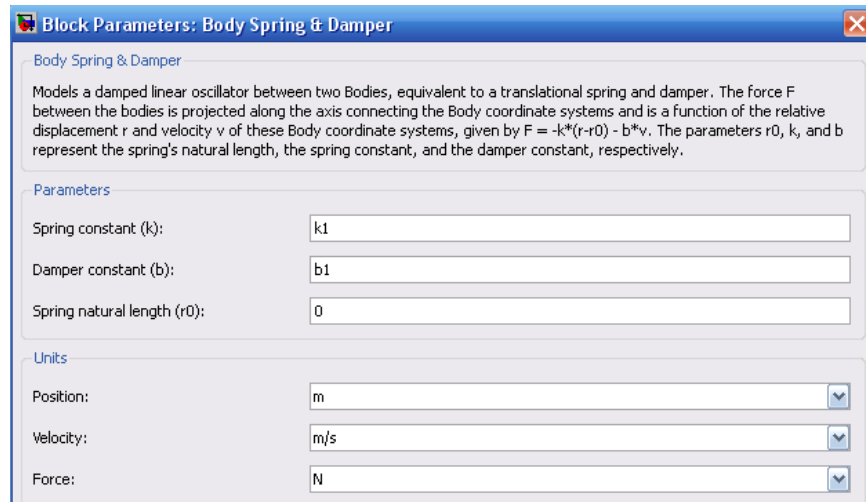


Figura 5-64

O bloco $\frac{1}{4}$ do autocarro tem os parâmetros definidos na Figura 5-65. As variáveis M e I são, respectivamente, a massa e a matriz de Inércia de $\frac{1}{4}$ do autocarro.

O ponto CG encontra-se no centro deste bloco de $\frac{1}{4}$ de autocarro. Assim, a sua localização dentro do AVS é definida segundo um vector de deslocamento $[0 \ c/2+R+0.04 \ 0]$. Como, para este ponto, a única coordenada que varia é a altura a que o bloco se encontra, o deslocamento é efectuado no eixo dos y . Como a altura do autocarro é definida pela variável c , o ponto CG se encontra no meio do autocarro e, como se tem de ter em conta altura do conjunto roda e suspensão (R), o deslocamento é efectuado da origem do AVS segundo uma soma $c/2+R$. O outro factor na soma (0.04) foi calculado através de várias tentativas para estabilizar o modelo virtual, visto que devido à força da gravidade e a dinâmica introduzida pelos blocos *Body Spring & Damper* e *Joint Spring & Damper* o modelo tinha umas oscilações iniciais sem ser aplicada qualquer perturbação. Este factor atenua estas perturbações iniciais para valores praticamente desprezáveis.

Os pontos $CS2$ a $CS6$ e $CS11$ estão definidos a partir das coordenadas do ponto CG . Os pontos $CS7$ a $CS10$ encontram-se deslocados dos pontos $CS3$ a $CS6$. Estes pontos (de $CS3$ a $CS10$) servem para definir os pontos do paralelepípedo que representa o bloco de massa de parte do autocarro.

Os pontos $CS3$ a $CS6$ (definidos a partir do CG) formam a área rectangular da parte de baixo do bloco do autocarro. Por isso encontram-se deslocados de $-c/2$ no eixo dos y . As resto das coordenadas são definidas segundo a largura (b) e o comprimento (a) do autocarro. Os pontos $CS7$ a $CS10$ encontram-se deslocados apenas na altura (c).

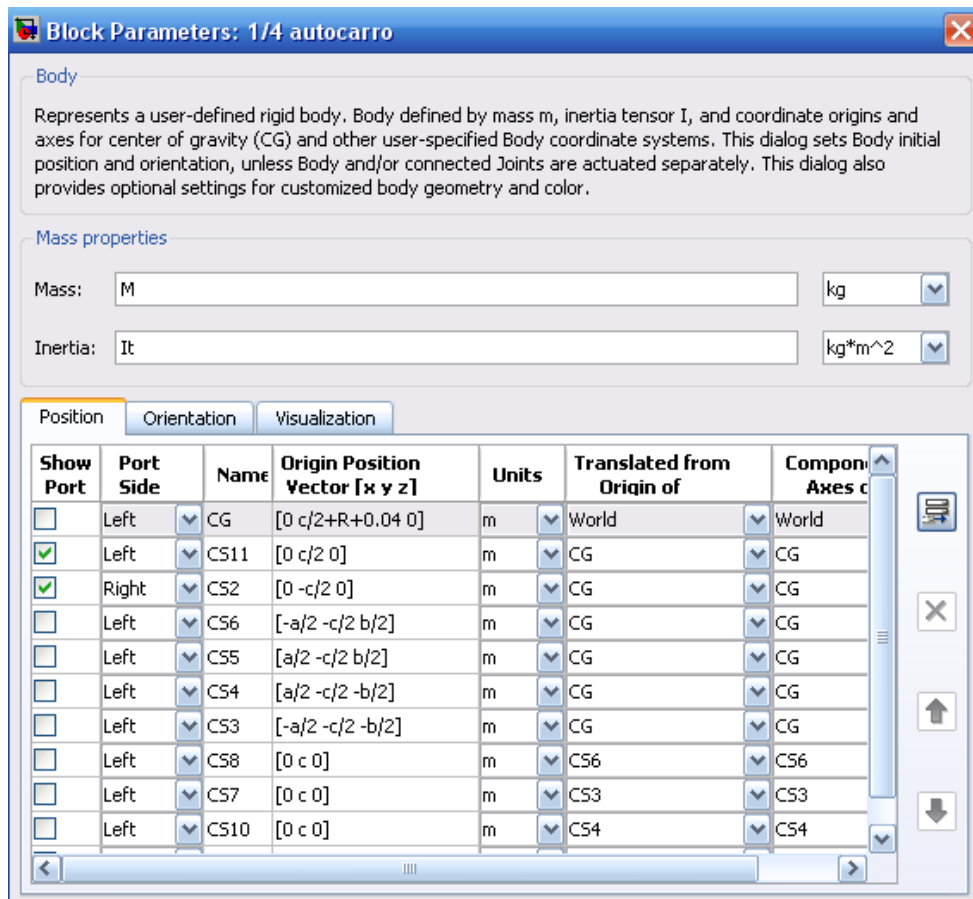


Figura 5-65

No ponto *CS11* do bloco anterior encontra-se conectada uma junta *Prismatic2*. Esta junta permite o movimento vertical no eixo dos y . Encontra-se, também, conectada a um bloco *Ground2*. Possui um actuador (*Joint Actuator*) e um sensor (*Joint Sensor1*) nas duas portas extras. Os parâmetros da junta *Prismatic2* encontram-se na Figura 5-66.

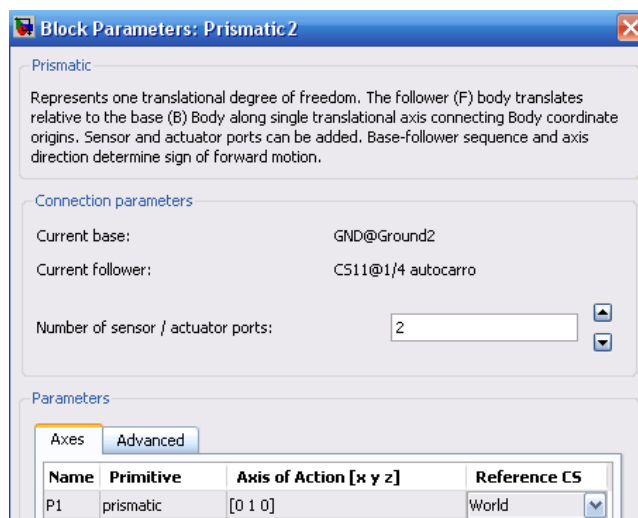


Figura 5-66

O bloco *Ground2* encontra-se no cimo e no centro do autocarro. Assim, basta somar ao ponto *CG* do bloco $\frac{1}{4}$ autocarro *c/2* no eixo dos *y* e encontra-se o vector de deslocamento $[0 \ c+R+0.04 \ 0]$ (Figura 5-67). *Ground2* não necessita de um bloco *Machine Environment*, visto que o bloco *Ground* já possui um.

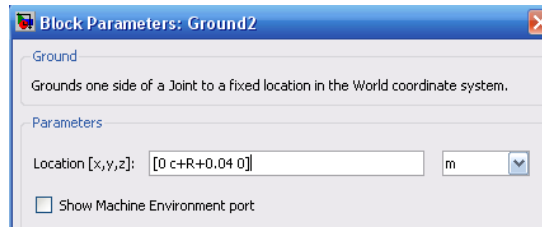


Figura 5-67

O bloco *Joint Actuator* actua na junta *Prismatic2* através da força (em N) que é a entrada deste sistema. Os parâmetros de configuração deste bloco encontram-se na Figura 5-68.

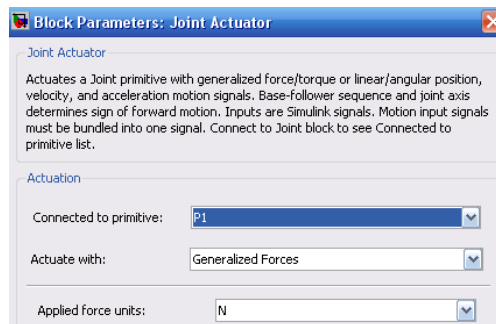


Figura 5-68

Por fim, tem-se o bloco *Joint Sensor1*. Este vai medir o deslocamento na junta *Prismatic1* (em m) do bloco do autocarro em relação ao AVS (“Reaction measured on: Follower; With respect to CS: Absolute(World)”). As configurações deste bloco encontram-se na Figura 5-69. Este bloco permite a saída da variável que se pretende controlar (*posicao*).

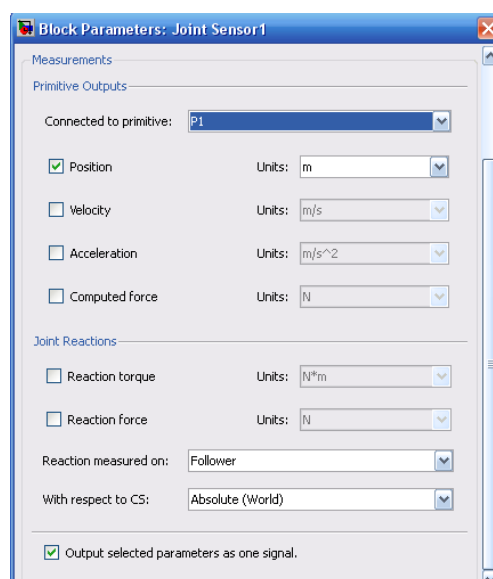


Figura 5-69

O esquema de eixos de actuação no sistema encontra-se na Figura 5-70:
Note-se que o eixo vertical onde actua a gravidade é, neste caso, o eixo y.

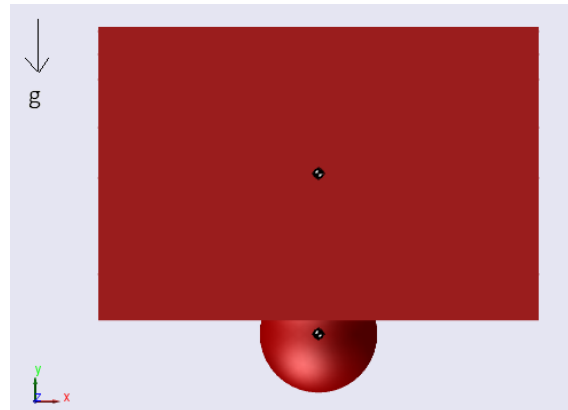


Figura 5-70 - Esquema de eixos

5.3.3 Especificações da dinâmica da suspensão de um autocarro

As especificações para o sistema de um quarto do autocarro mais o conjunto de suspensão e roda, foram retiradas de [2]. Quando o sistema é perturbado com um degrau de 10 cm (uma protuberância na estrada por exemplo) as especificações da dinâmica para o regime transitório são:

- Tempo de subida < 5 segundos;
- Percentagem de sobrelevação < 5%

As especificações para o regime estacionário:

- Tempo de estabelecimento < 5 segundos
- Erro em regime estacionário < 5mm

Para a criação do modelo virtual através do SimMechanics procedeu-se, primeiro, ao cálculo do modelo linear sem controlo para se determinar os valores dos parâmetros dos controladores a aplicar. Neste caso, foram utilizadas técnicas de controlo linear.

Para obter o modelo linear do sistema consultou-se [2]. Através dos métodos referidos no site obteve-se a função de transferência do sistema.

Após o cálculo da função de transferência do sistema aplicou-se um impulso de 10 cm. Todos os comandos para a obtenção da função de transferência encontram-se em Anexo 3. Os resultados encontram-se na Figura 5-71.

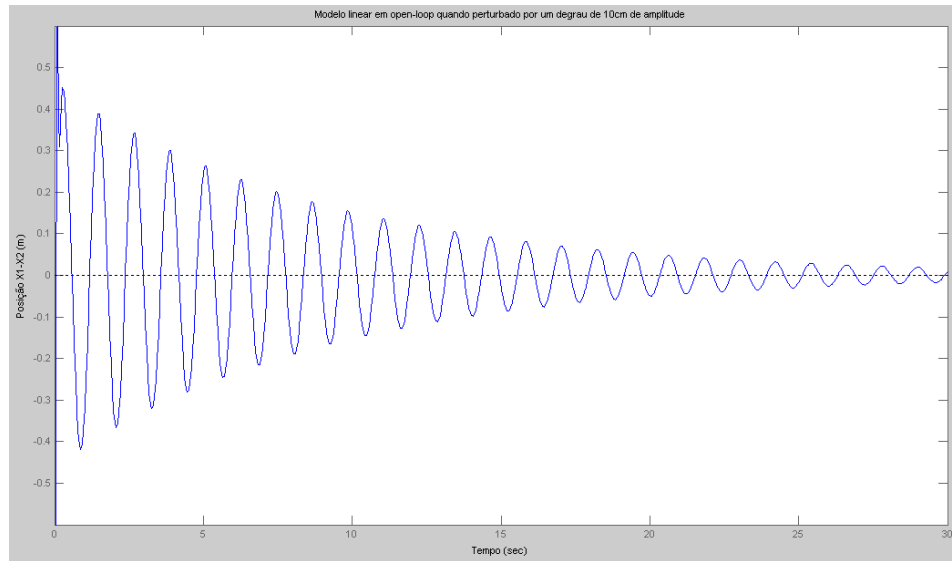


Figura 5-71

Como se pode verificar, o autocarro, ao passar um degrau de 10cm de altura, sofre perturbações de amplitude de quase 10cm.

Todos os comandos necessários em Matlab encontram-se no Anexo 3.

5.3.4 Modelo virtual da suspensão do autocarro na presença de uma perturbação sem controlador

Para verificar se o modelo virtual realizado se comportava de acordo com o modelo linear obtido, introduziu-se uma perturbação neste modelo através do bloco “Body Actuator”. A perturbação é simulada através do bloco “Signal Builder” que tem como características a Figura 5-72.

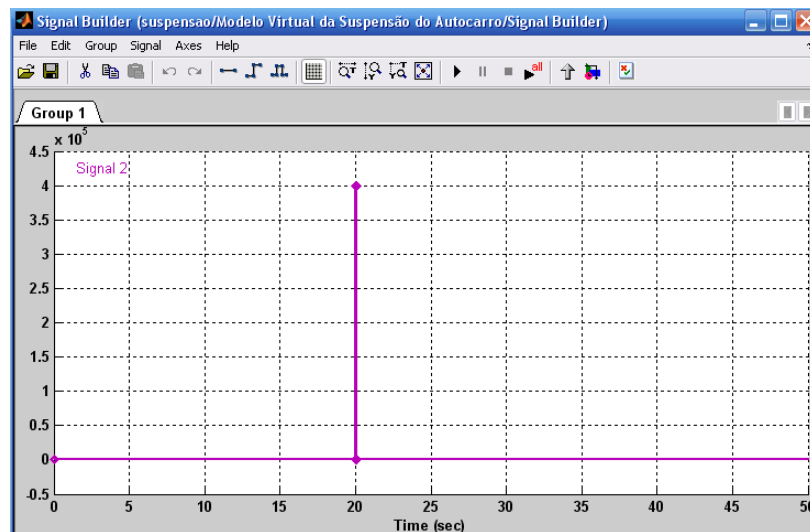


Figura 5-72

O valor utilizado foi obtido após várias tentativas. A perturbação ocorre apenas aos 20 segundos devido às oscilações iniciais inerentes à dinâmica do modelo virtual. Assim, só se pode ter em conta os valores após os 20 segundos iniciais, sendo que os anteriores servem apenas para a estabilização do sistema mecânico.

Este sinal é introduzido no bloco “Body Actuator” e este actua na roda do autocarro simulando o degrau. O bloco *Modelo Virtual da Suspensão do Autocarro* foi inserido no diagrama seguinte:

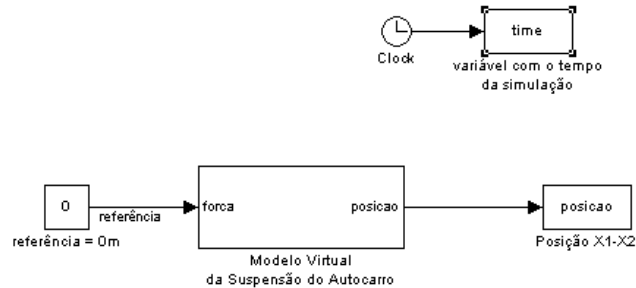


Figura 5-73 – Diagrama de blocos do Modelo Virtual da Suspensão do Autocarro sem controlador

Executou-se a simulação durante 50 segundos e os resultados encontram-se na Figura 5-74.

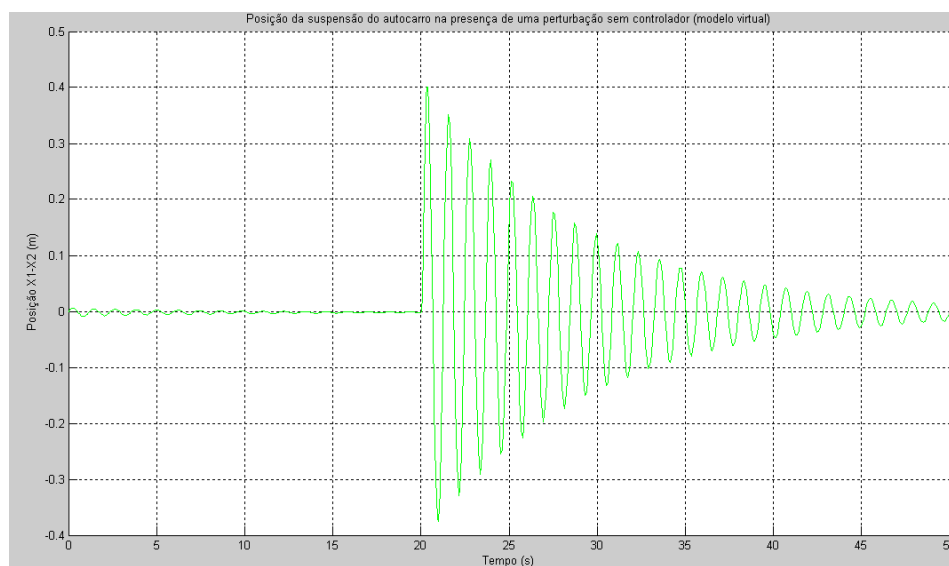


Figura 5-74

Como se pode verificar os valores obtidos na simulação foram idênticos aos valores obtidos para o modelo linear, o que leva à conclusão que o modelo virtual se encontra bem realizado e de acordo com a dinâmica da suspensão do autocarro. De notar que os resultados só começam após 20 segundos sendo que, durante esse tempo inicial, o sistema tende a estabilizar as oscilações iniciais inerentes à dinâmica do sistema.

Quer para o modelo linear da suspensão do autocarro, quer para o modelo virtual, o sistema não cumpre com as especificações porque possui oscilações de amplitude elevada e demora muito a estabilizar na referência (zero). Assim, é necessário realizar um controlador que consiga cumprir com as especificações referidas anteriormente.

5.3.5 Esquema do controlo do modelo virtual da suspensão do autocarro

Controlador PID

A equação dinâmica do controlador PID é a mesma usada no capítulo 5.1.4, assim como o diagrama de blocos em Simulink é o mesmo da Figura 5-15.

Os parâmetros foram retirados do site [3]. Estes valores foram ajustados através de várias tentativas de modo a se adequarem às especificações requeridas e de modo a apresentarem um valor lógico, já que os valores do site são elevados para serem realizáveis na prática.

Os parâmetros do controlador PID tomam os seguintes valores:

$K_p = 16642$;

$K_i = 12482$;

$K_d = 4161$;

A malha de controlo para o modelo virtual da Suspensão do Autocarro encontra-se na Figura 5-75.

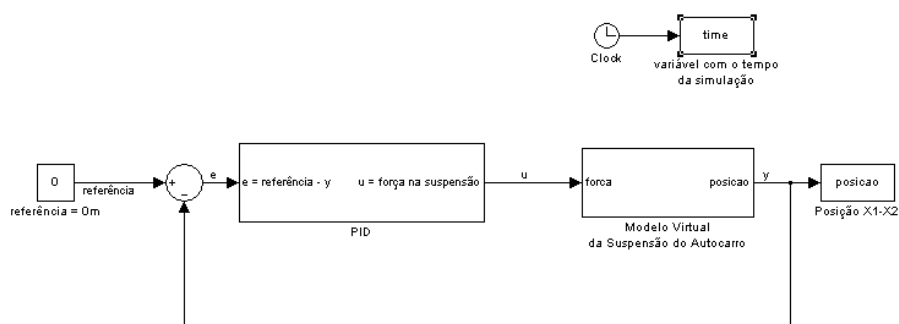


Figura 5-75 – Diagrama de blocos do modelo virtual do controlo da suspensão do autocarro com controlador PID

Executou-se a simulação ao longo de 50 segundos e o gráfico obtido encontra-se esquematizado na Figura 5-76.

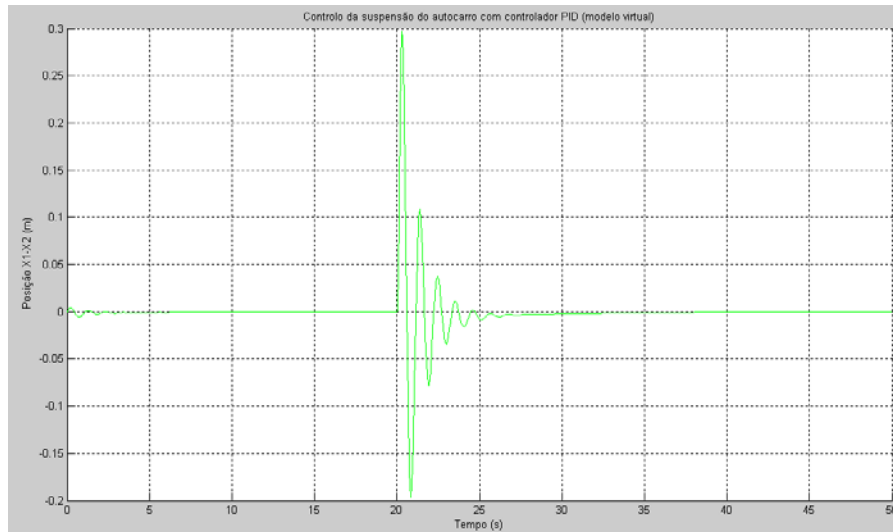


Figura 5-76

Com a mesma perturbação a afectar o sistema, concluiu-se que a sobrelevação não é inferior a 5%. Assim fica por cumprir uma das especificações requeridas para este sistema. Optou-se então por outro controlador: controlador por cancelamento de pólo zero.

Controlador por cancelamento de pólo zero

Para determinar os parâmetros deste controlador utilizou-se o site [4]. Através do site e de uma série de comandos em Matlab que se encontram em Anexo 3, determinou-se a função de transferência correspondente a este tipo de controlador.

Assim a malha de controlo com este tipo de controlador encontra-se na Figura 5-77. Como se pode verificar o bloco do controlador possui os valores determinados no ficheiro no Anexo 3.

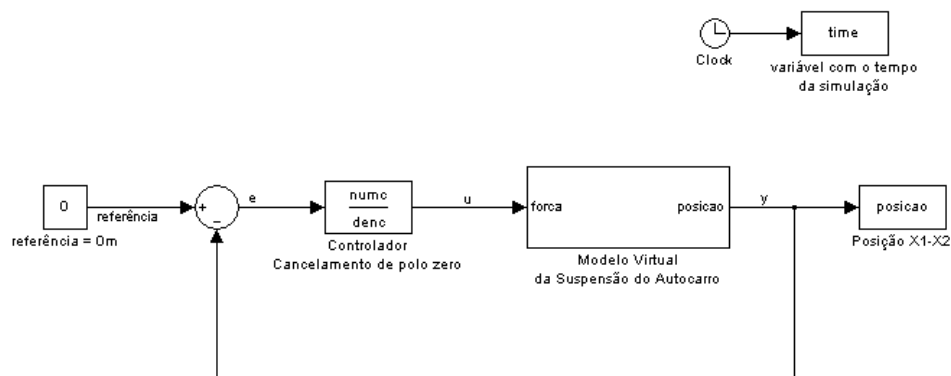


Figura 5-77 – Diagrama de blocos do modelo virtual da suspensão do autocarro com controlador por cancelamento de pólo zero

Após correr a simulação durante 30 segundos, os resultados obtidos foram:

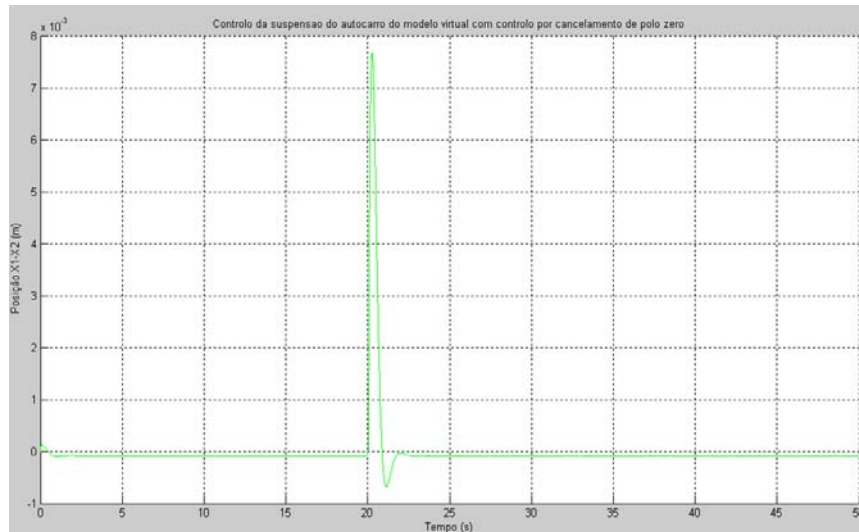


Figura 5-78

Na Figura 5-78 pode-se verificar que o controlador por cancelamento de pólo zero já cumpre com todas as especificações requeridas inicialmente, incluindo o requisito da sobre-elevação.

Conclui-se então que, para este caso, este controlador apresenta melhor desempenho que o controlador PID e que o modelo realizado em laboratório virtual se encontra de acordo com a dinâmica da suspensão de um autocarro.

6 Virtual Reality Toolbox

6.1 Introdução

A “Virtual Reality Toolbox” permite a interacção entre sistemas dinâmicos ao longo do tempo, com modelos de realidade virtual.

Permite a criação de ambientes virtuais a três dimensões usando a linguagem VRML padrão (“Virtual Reality Modeling Language”). Possibilita a interacção entre sinais provenientes do Simulink com o movimento de objectos a três dimensões dentro do ambiente virtual.

Para a criação do modelo virtual utilizou-se o “VRealm Builder 2.0”. Um editor de VRML orientado a objectos que faz parte da “Virtual Reality Toolbox” incluída no Matlab R2008b (exemplo de localização do programa: `c:\Programas\Matlab\R2008b\toolbox\vr\vrealm\program\vrbuild2.exe`).

Nota: Para versões mais recentes do Matlab, “Virtual Reality Toolbox” passou a ser denominada como “Simulink 3D Animation”.

6.2 Objectivos

Criação de animações com os respectivos ambientes virtuais a três dimensões para os modelos criados anteriormente em SimMechanics (controlo da velocidade do automóvel).

6.3 VRealm Builder 2.0

6.3.1 Introdução ao editor VRealm Builder 2.0

A interface gráfica do editor VRealm Builder 2.0 encontra-se na Figura 6-1.

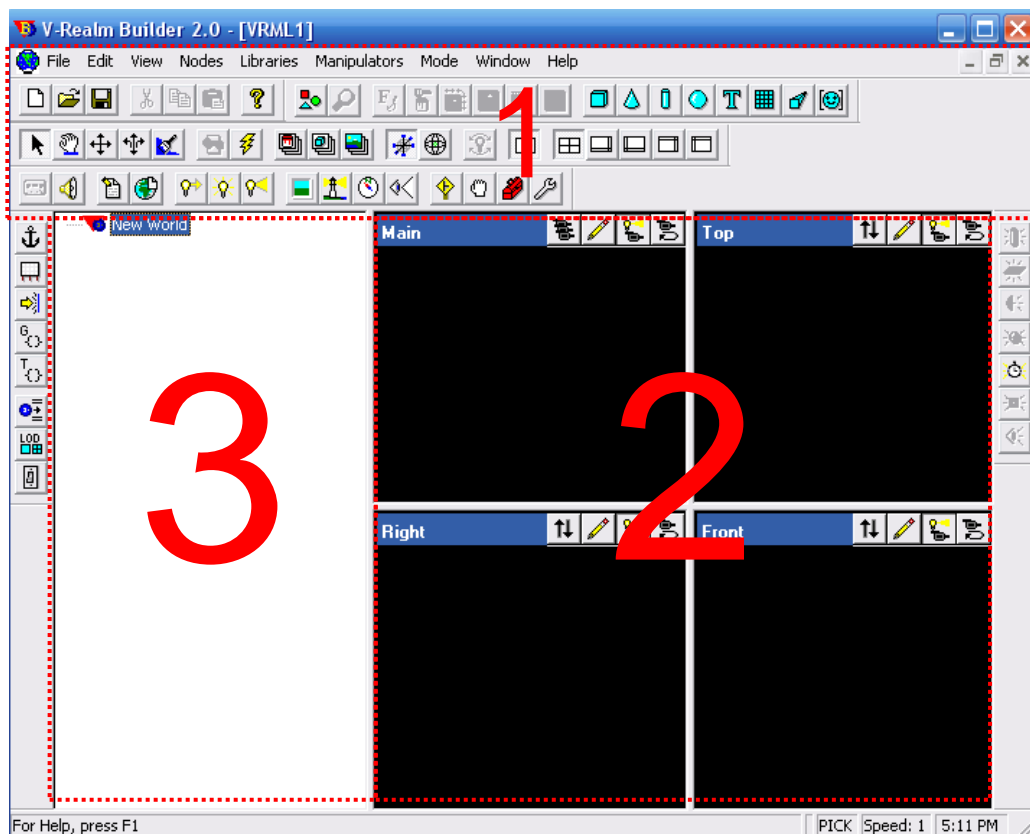


Figura 6-1 – Editor VRealm Builder 2.0

Como se pode verificar têm-se definidas três zonas distintas: em cima, uma barra de ferramentas com múltiplas opções (1); à direita, quatro perspectivas sobre o ambiente virtual que se pretende criar (2); à esquerda, é uma zona onde aparecem todos os objectos (e respectivas propriedades) que se vão acrescentando ao ambiente virtual (3).

Para se inserir um objecto no ambiente virtual pode-se optar por umas poucas formas simples pré-definidas (ver Figura 6-2): um cubo, um cone, um cilindro ou uma esfera (da esquerda para a direita, respectivamente).

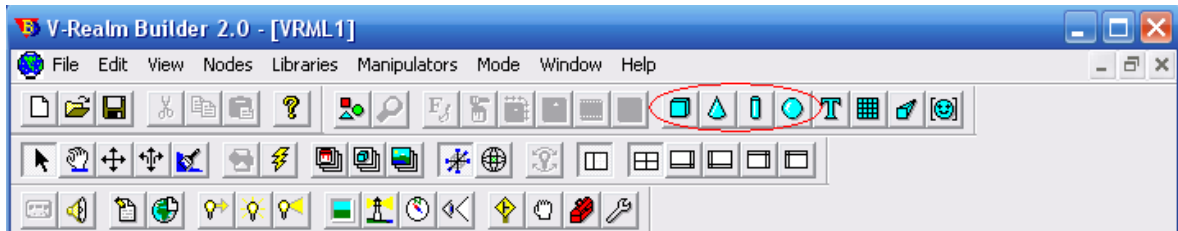


Figura 6-2 – Barra de ferramentas do editor VRealm Builder 2.0

Ao se adicionar o objecto ao ambiente virtual, este vem com definições que podem ser alteradas. Na Figura 6-3 encontram-se algumas das propriedades mais importantes.

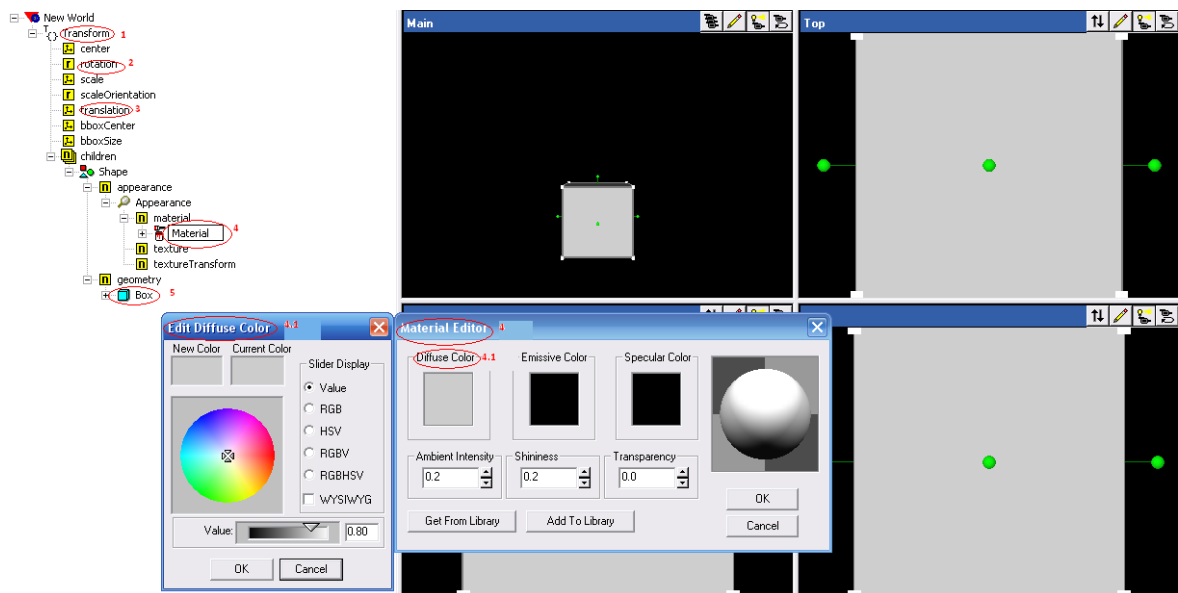


Figura 6-3 – Propriedades e aspecto de um objecto inserido no modelo virtual (um cubo)

Legenda:

1. Permite alterar o nome do objecto;
2. Roda o objecto ao alterar os valores de x,y e z;
3. Desloca o objecto segundo um vector que se define neste campo, reposicionando-o dentro do ambiente virtual;
4. Altera a cor e brilho do objecto (ver também 4.1)
5. Permite alterar as dimensões do próprio objecto (de notar que os valores das dimensões são relativos dentro do ambiente virtual, ou seja, não têm unidades)

Pode-se inserir objectos de livrarias que, em vez de serem formas simples, já têm alguma complexidade, mas que vêm com o próprio editor. Na figura Figura 6-4 tem-se o exemplo de uma árvore que foi inserida a partir das livrarias existentes do editor.

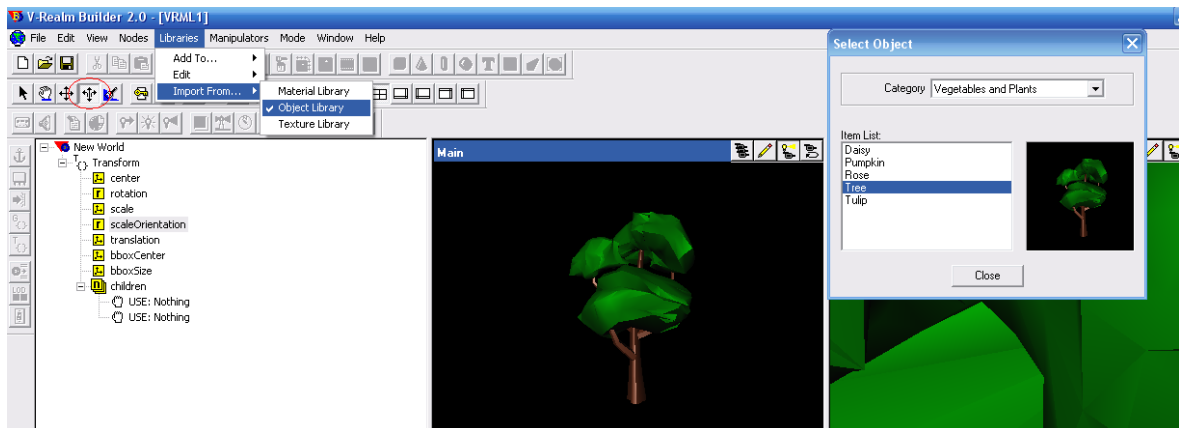


Figura 6-4 – Exemplo de um objecto inserido a partir de uma livraria existente

Note-se que o botão seleccionado e assinalado com um círculo vermelho é o botão de zoom do ambiente virtual.

Para se alterar o fundo pode-se optar por editar um “background” já existente no editor (Figura 6-5). O “background” é adicionado (carregando no botão assinalado pelo círculo a vermelho) na zona de objectos e pode-se alterar cada uma das suas características.

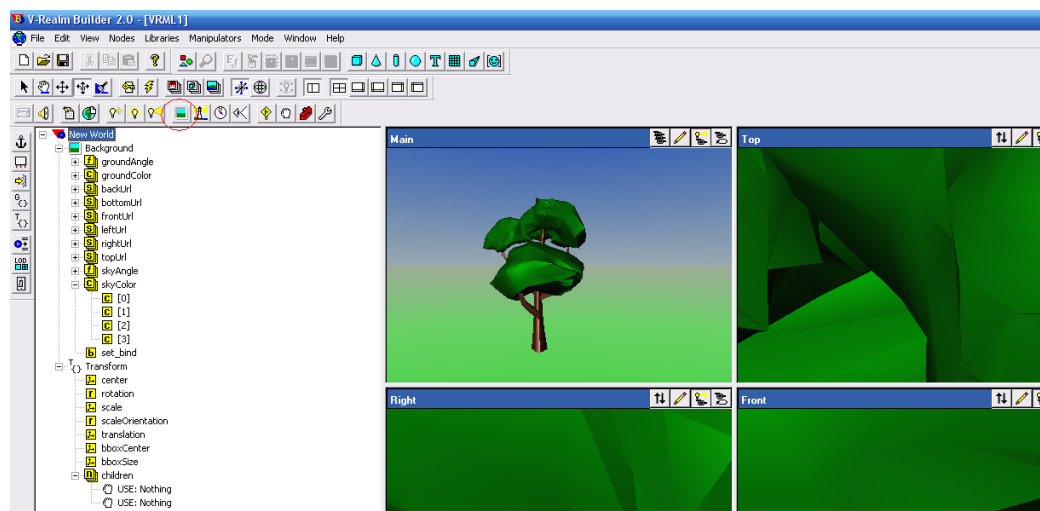


Figura 6-5 – Exemplo da inserção de um background

Por fim, pode-se definir o ponto de visualização para ser a perspectiva que o utilizador tem, quando se correr a simulação. Na Figura 6-6 encontra-se o modo de adicionar o “Viewpoint” (ponto de vista). Ao se inserir um “Viewpoint” no ambiente virtual, este fica definido para a perspectiva actual que o utilizador tem do campo “Main”. Para, posteriormente se alterar o ponto de vista, é necessário alterar o campo assinalado pelo círculo a vermelho (“position”).

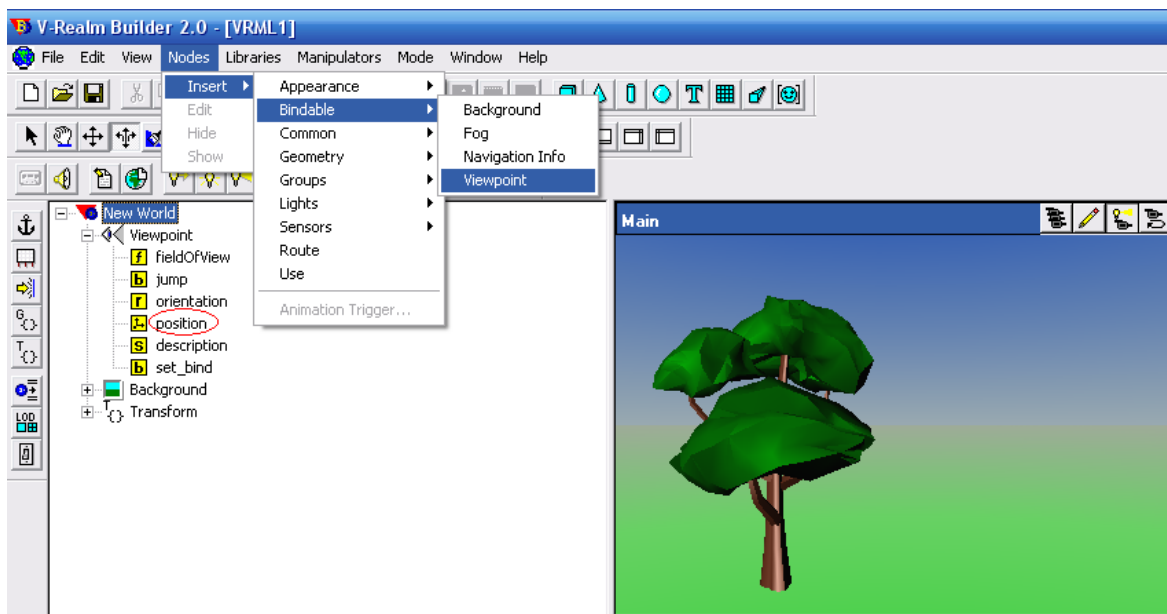


Figura 6-6 – Exemplo da inserção da perspectiva de visualização

6.4 Modelo virtual do automóvel

6.4.1 Ambiente virtual do automóvel criado em VRealm Builder

Começou por inserir-se um objecto (*box*) no modelo em VRML. O objecto teve as suas dimensões ajustadas (alterando o campo *Size*) de modo a se assemelhar a uma estrada (1). De seguida, inseriu-se, a partir das bibliotecas do programa, um carro (2) e três árvores (3). O carro manteve-se na sua posição original, enquanto as árvores foram deslocadas no ambiente virtual (alterando o campo *translation*). Por fim acrescentou-se o fundo (*Insert - Background*) e deixou-se com os valores predefinidos.

Nota: Os valores entre parênteses dizem respeito à Figura 6-7.

No final acrescentou-se uma casa que foi deslocada para a direita do ambiente virtual (mudando o campo *translation*) de modo semelhante ao referido anteriormente.

Para terminar acrescentaram-se dois pontos de visualização (*Insert – Bindable - Viewpoint*) distintos (Figura 6-8 e Figura 6-9).

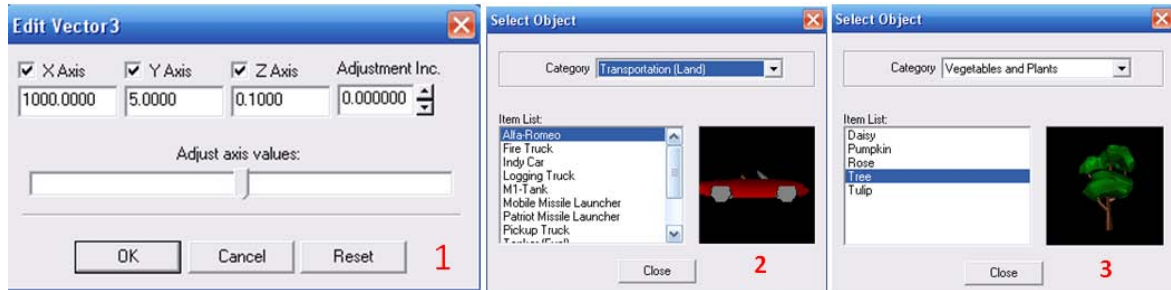


Figura 6-7

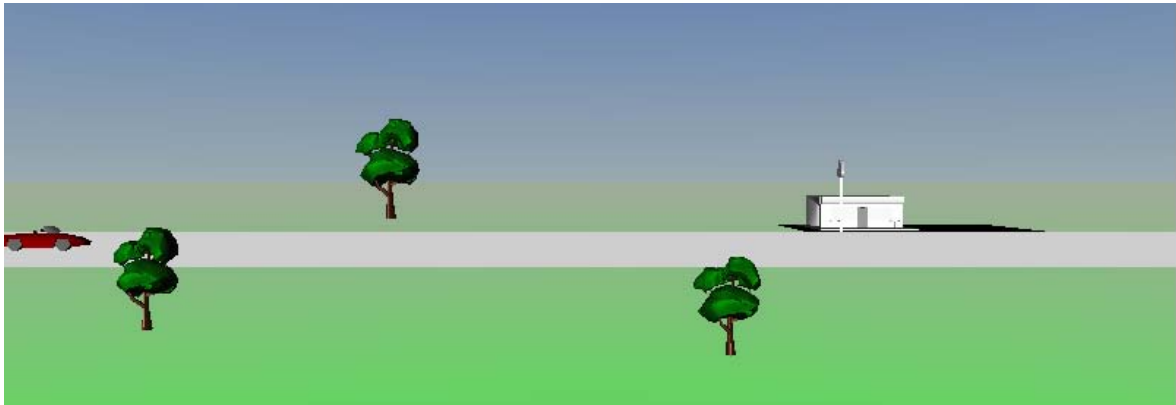


Figura 6-8 – Viewpoint - ‘side view’



Figura 6-9 – Viewpoint – ‘3d view’

Pelas figuras (Figura 6-8 e Figura 6-9) pode-se verificar o aspecto final do ambiente virtual a três dimensões para o modelo do automóvel.

6.4.2 Ligação do modelo criado em VRealm com o modelo em SimMechanics

Após se ter criado o modelo em VMRL, utilizou-se o modelo do automóvel criado em SimMechanics. O bloco *Modelo Virtual do Carro* e o bloco *PID* são os mesmos das Figura 5-3 e Figura 5-15 respectivamente, excepto para o bloco *Modelo Virtual do Carro*, ao qual se acrescentou uma nova saída para se obter a posição do carro. Foram necessários

dois blocos da *Virtual Reality Toolbox*. O bloco de *referencia* tem propriedades definidas na Figura 6-10 e a malha de controlo encontra-se esquematizada na Figura 6-11.

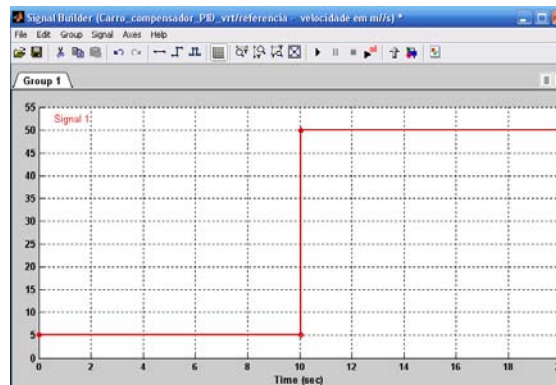


Figura 6-10

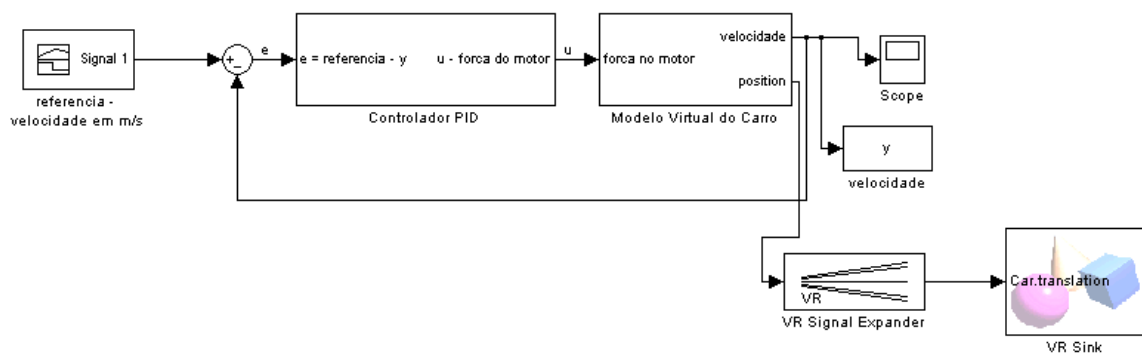


Figura 6-11 – Diagrama de blocos da malha de controlo do modelo virtual do automóvel com controlador PID e animação a três dimensões (Virtual Reality Toolbox)

O bloco *VR Signal Expander* serve para transformar um sinal proveniente do Simulink num sinal compatível com o ambiente VRML. Transforma a saída num vector com três colunas em cada uma delas representa um eixo de acção (Figura 6-12).

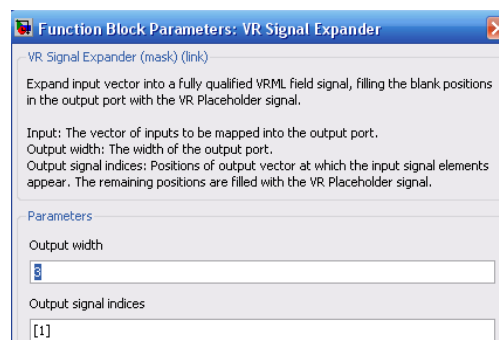


Figura 6-12

Como se pode verificar, este bloco usa a saída em Simulink da posição do carro e cria um sinal com três colunas (*Output width*). Coloca, também, os valores da saída em Simulink da posição do carro na primeira coluna (*Output signal indices*), correspondendo

esta ao eixo dos x. As outras colunas correspondentes aos eixos y e z são colocadas a zero.

O bloco *VR Sink* é o bloco que, efectivamente, liga um modelo em Simulink a um modelo em VRML. As definições deste bloco encontram-se na Figura 6-13.

Neste bloco, são seleccionados os pontos onde os sinais provenientes do Simulink vão actuar dentro do modelo em VRML. Para cada ponto e para cada tipo de movimento diferentes, uma nova entrada é adicionada a este bloco. Neste caso do automóvel, é apenas no carro que se quer actuar com movimento de translação e por isso se seleccionou a checkbox presente na Figura 6-13. Assim, tem-se apenas a entrada *Car.translation* presente na malha de controlo da Figura 6-11.

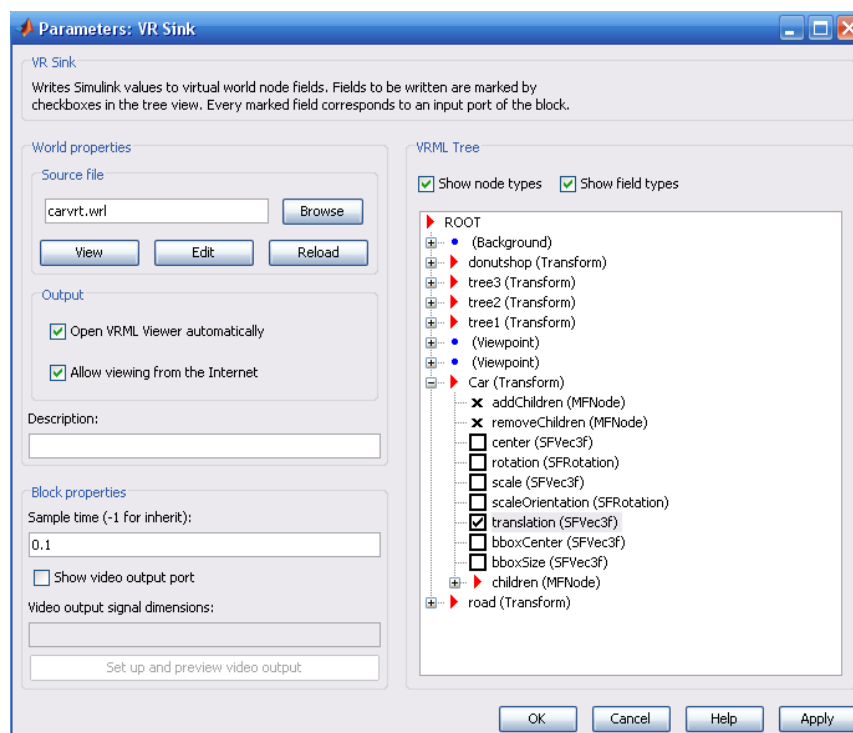


Figura 6-13

Após a execução da simulação ao longo de 15 segundos verificou-se que o carro se movia ao longo do ambiente virtual a três dimensões (tendo a opção de alternar entre os dois pontos de vista criados. Pode-se verificar que o carro acelera até aos 5 m/s e após 10 segundos de simulação aumenta a sua velocidade até estabilizar nos 50 m/s.

6.5 Modelo virtual do pêndulo invertido

6.5.1 Ambiente virtual do pêndulo invertido criado em VRealm Builder

Para a criação do Ambiente virtual para o pêndulo invertido foram criados três blocos das formas pré-definidas do editor.

Primeiro inseriu-se uma *Box* para fazer a calha onde se move o carro. O campo *Size* foi alterado de acordo com os valores da Figura 6-14 (1). De seguida inseriu-se outra *Box* para representar o carro onde se apoia o pêndulo. Mais uma vez, alteraram-se os valores do campo *Size* (Figura 6-14 (2)). Para representar o pêndulo inseriu-se um bloco *Cylinder*. Como o pêndulo está conectado ao carro, este o bloco *Cylinder* não foi acrescentado no ambiente virtual em si, mas inserido como parte integrante do carro (ver Figura 6-15 – zona das propriedades dos objectos). Como se pode verificar, o bloco *pendulo* encontra-se inserida no campo *children* do *carro*.

Os valores da altura e raio do cilindro encontram-se no campo (*Shape – Geometry – Cylinder (height = 8; radius = 0.3)*). A força que actua no movimento do pêndulo é aplicada no ponto onde este se conecta ao carro. Por isso foi necessário alterar o centro de massa do pêndulo para o ponto de conexão com o carro. Se se mantivesse o ponto central do pêndulo como o seu centro de massa, este rodaria sobre ele próprio. Assim, o campo *center* alterou-se (ver Figura 6-14 (3)) deslocando-o para o ponto de conexão com o carro.

Os blocos representando o carro e o pêndulo tiveram as suas cores e texturas alteradas (campo *Material*).

Por fim, inseriu-se um ponto de visualização (*Insert – Blindable - Viewpoint*) para se visualizar o modelo de acordo com a Figura 6-15.

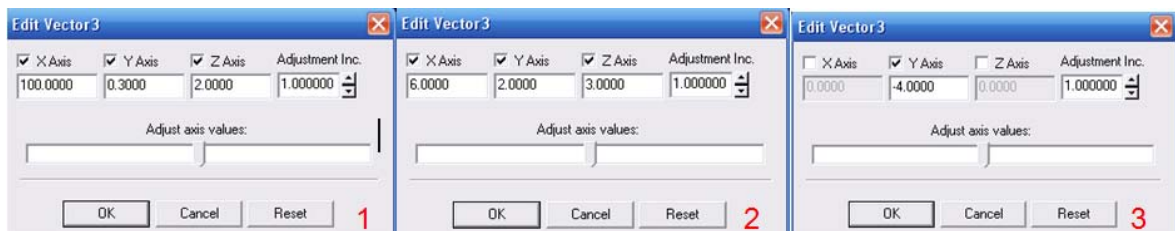


Figura 6-14 – Valores dos parâmetros dos objectos inseridos

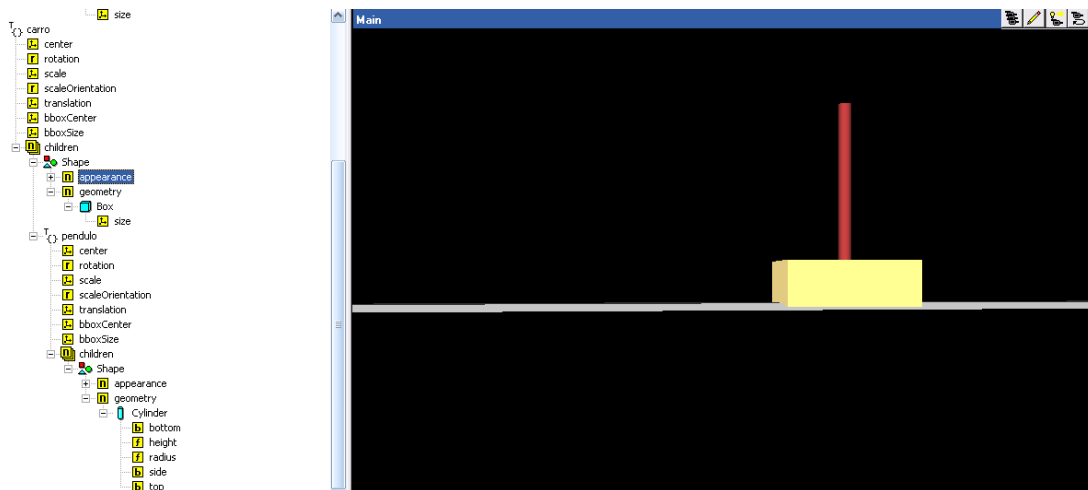


Figura 6-15 – Viewpoint – ‘side view’

6.5.2 Ligação do modelo criado em VRealm com o modelo em SimMechanics

Após se ter criado o modelo em VMRL, utilizou-se o modelo do pêndulo invertido criado em SimMechanics. Usou-se o modelo virtual usando anteriormente representado na Figura 5-52. Para uma questão de melhor visualização de resultados (nomeadamente na oscilação do pêndulo) o controlador usado foi o PID. Alterou-se o valor inicial do pêndulo (bloco Joint Initial Conditions) para um deslocamento de 15° da vertical e, também, o sinal da perturbação (bloco *Signal Builder1*) foi alterado para os valores presentes na Figura 6-16.

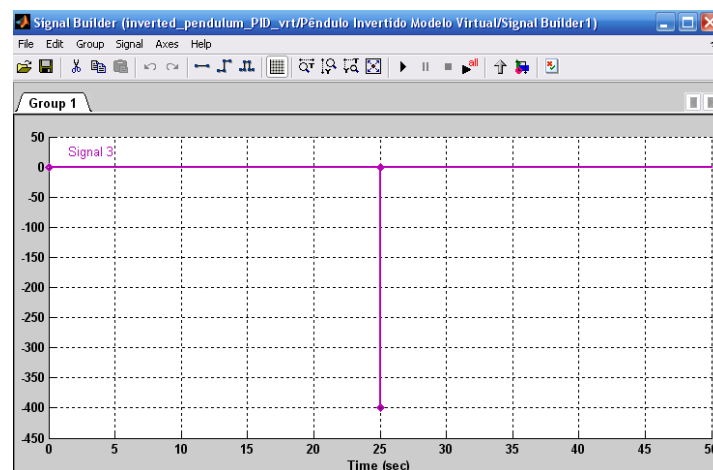


Figura 6-16

A malha de controlo usada encontra-se na Figura 6-17.

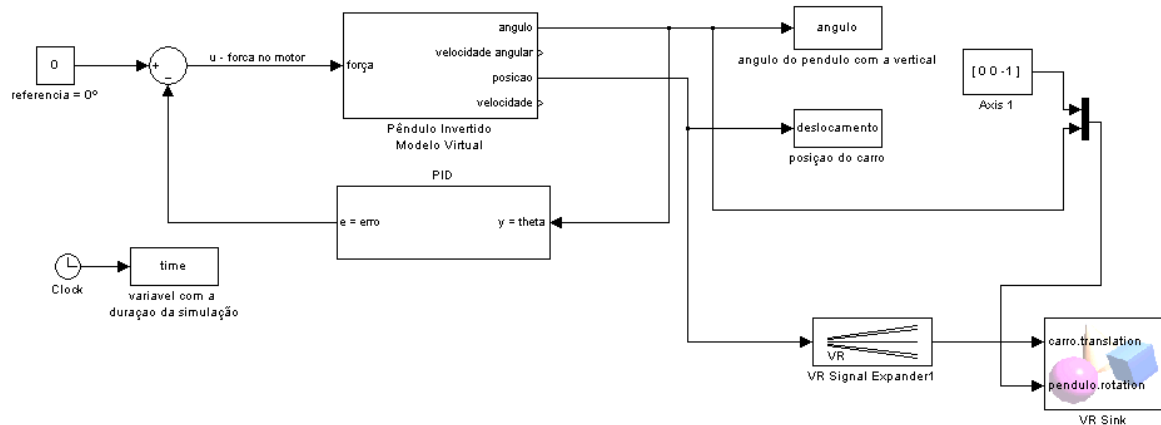


Figura 6-17 - Diagrama de blocos da malha de controlo do modelo virtual do pêndulo invertido com controlador PID e animação a três dimensões (Virtual Reality Toolbox)

Neste caso existem dois sinais a actuar no ambiente a três dimensões criado, a deslocação do carro e as oscilações do pêndulo.

Para o movimento do carro, como se pode verificar na Figura 6-17, tem-se novamente um bloco *VR Signal Expander1*. As suas propriedades encontram-se na Figura 6-18.

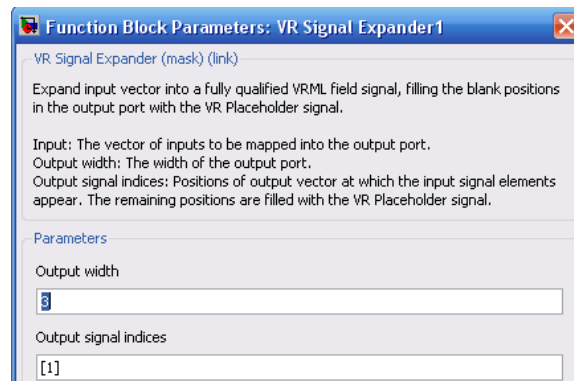


Figura 6-18

Mais uma vez, o sinal que entra no bloco é a posição do carro. A deslocação ocorre no primeiro eixo de acção (eixo dos x) (*Output signal indices*). Nos outros dois eixos não existe movimento necessário à deslocação do carro mas o vector de entrada terá de ser um vector com três colunas, logo o campo *Output width* toma o valor 3 (preenchendo as outras colunas com zero).

Para o movimento rotacional do pêndulo é necessário um vector com 4 colunas. A primeira coluna com os valores da variação do ângulo e as outras três para indicar em que eixos de acção ocorre o movimento. Neste caso a rotação ocorre no eixo dos z e por

isso teve de se inserir um bloco com um vector $[0 \ 0 \ -1]$ para se juntar ao sinal vindo do Simulink.

Por fim, tem-se o bloco *VR Sink*. Como se pode observar na Figura 6-19, as checkboxes seleccionadas correspondem ao movimento de translação do carro e ao movimento de rotação do pêndulo, dando origem às entradas *carro.translation* e *pêndulo.rotation* do bloco *VR Sink*.

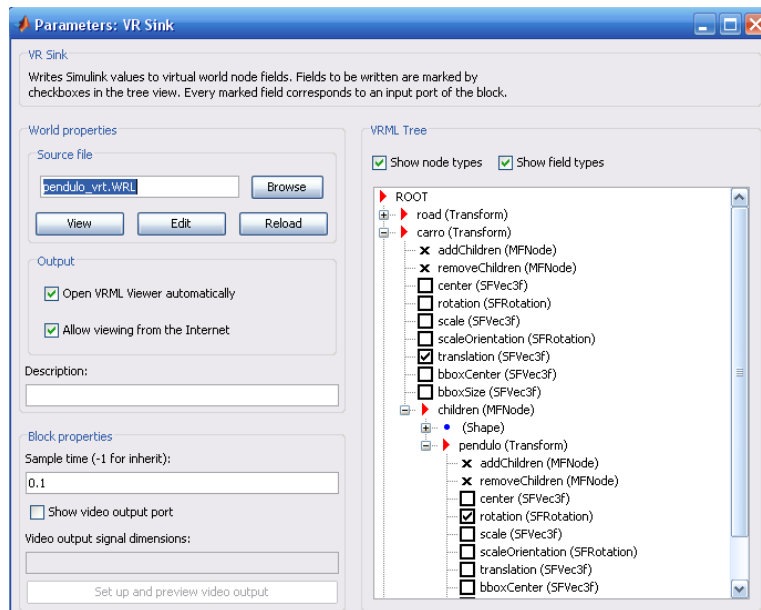


Figura 6-19

Depois de se executar a simulação ao longo de 50 segundos verificou-se que o carro desloca-se ao longo da calha (dentro do ambiente a três dimensões criado e representado na Figura 6-15). Verifica-se, também, que o carro compensa o deslocamento quer da posição inicial de 15º do pêndulo, quer, após 25 segundos, da perturbação inserida, mantendo o pêndulo na vertical.

6.6 Modelo virtual da suspensão do autocarro

6.6.1 Ambiente virtual da suspensão do autocarro criado em VRealm Builder

A criação do ambiente virtual a três dimensões para a suspensão do autocarro foi realizada, como para o caso do pêndulo, com formas pré-definidas do editor. Os passos realizados foram semelhantes aos dois casos anteriores por isso não se entrará em muito detalhe neste caso, referindo apenas que blocos se usou para a criação do ambiente a três dimensões.

Inseriu-se uma *Box* para representar o bloco de $\frac{1}{4}$ do autocarro (com as dimensões alteradas no campo *Size*), um bloco *Cylinder* para representar o pneu e outro bloco *Cylinder* para representar a roda. O bloco que representa a roda foi inserido como parte integrante do bloco que representa o pneu (campo *children* do bloco *tire*).

As cores foram alteradas para cada um dos blocos no campo *Material*.

Por fim acrescentou-se outro bloco *Box* para representar a estrada.

Foi acrescentado um fundo diferente (*Insert - Background*) deixando-o com as definições iniciais que este apresenta.

Inseriu-se um ponto de vista (*Insert - Blindable - Viewpoint*) com o nome *3d view* e cuja representação se encontra na Figura 6-20.

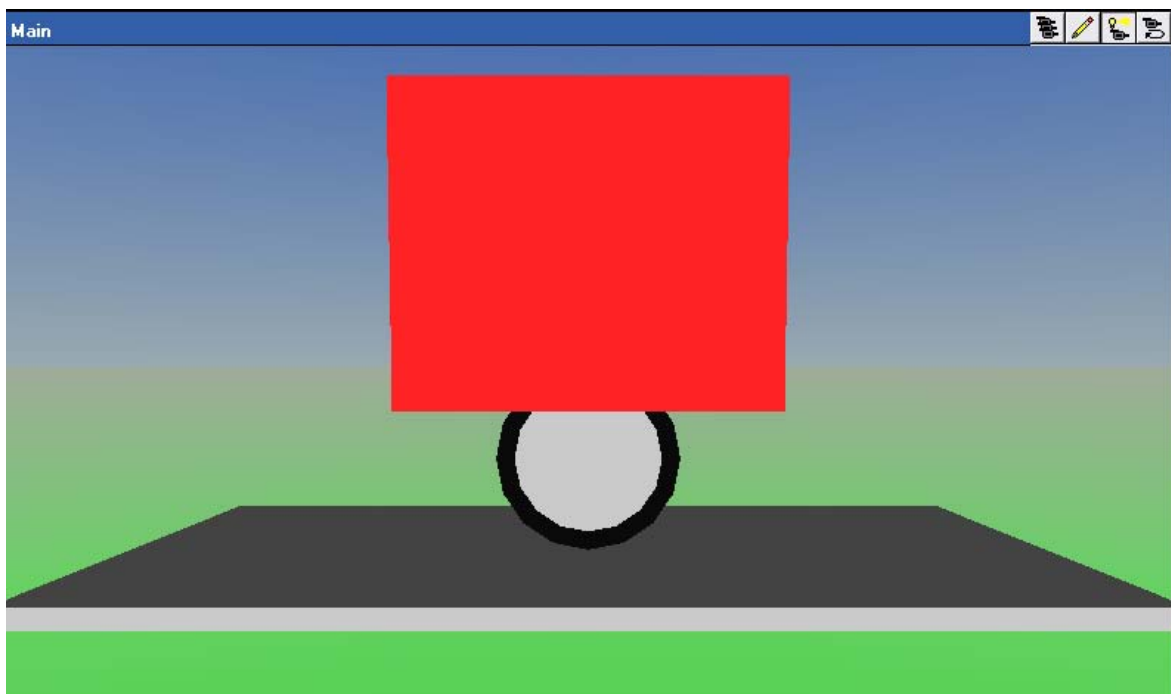


Figura 6-20 – Viewpoint – ‘3d view’

6.6.2 Ligação do modelo criado em VRealm com o modelo em SimMechanics

Para efeitos de melhor visualização da animação, o modelo usado para a implementação do ambiente a três dimensões foi o modelo da suspensão sem controlador. Ou seja, o modelo virtual criado em SimMechanics (Figura 5-56) com uma perturbação (a mesma usada na Figura 5-72) mas sem nenhum controlador.

Ao modelo criado anteriormente, adicionaram-se os blocos necessários a fazer a ligação entre SimMechanics e o modelo criado em VRML (Figura 6-21).

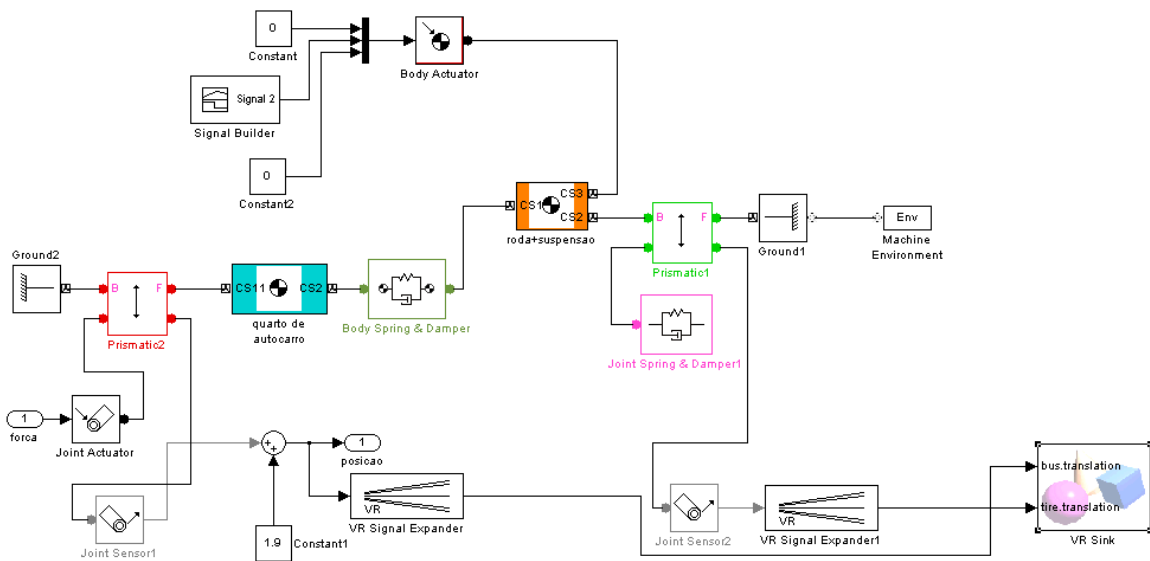


Figura 6-21 - Diagrama de blocos do modelo virtual da suspensão do autocarro sem controlador e com animação a três dimensões (Virtual Reality Toolbox)

A perturbação (com os mesmos valores da Figura 5-72) é inserida pelo bloco *Body Actuator* e ocorre após 20 segundos para o modelo estabilizar as oscilações iniciais que possui inerentes à sua dinâmica (como acontecia anteriormente). Neste caso tem-se dois blocos *VR Signal Expander* mas as suas propriedades (Figura 6-22) são iguais, visto o movimento de translação se realizar segundo o mesmo eixo. Mais uma vez, o movimento ocorre segundo o eixo dos *y* (*Output signal indices*) sendo que os outros dois eixos não possuem movimento (*Output width*).

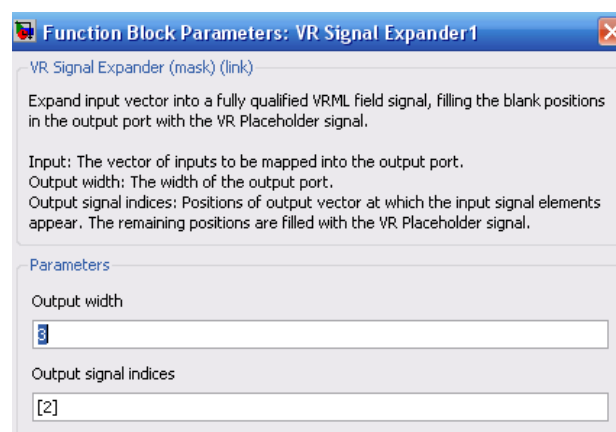


Figura 6-22

Para o deslocamento do bloco do autocarro, é necessário criar um novo zero virtual, visto que os valores da posição calculados pelo modelo em SimMechanics são relativos e não

têm em conta a altura a que se encontra o autocarro. Assim, foi necessário acrescentar ao sinal da posição do autocarro o valor da altura (bloco *Constan1*) a que este se encontra, antes de se interligar o sinal com o modelo criado em VRML.

O bloco *VR Sink* tem as suas propriedades definidas na Figura 6-23. As checkboxes activas são relativas ao movimento translacional do bloco do autocarro e do bloco do pneu (a roda, como se encontra inserida como parte integrante do bloco pneu não necessita de um sinal para se movimentar).

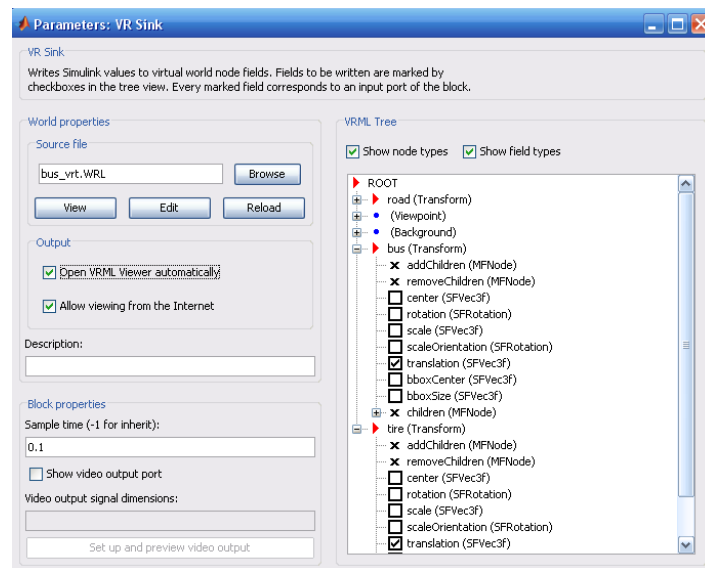


Figura 6-23

Assim, após se executar a simulação durante 50 segundos, pode-se verificar que a animação se move de acordo com os sinais inseridos obtendo um aspecto visual bastante superior ao obtido com a animação integrante no SimMechanics.

6.7 Conclusões

Após se terem realizado os ambientes virtuais em VRML para os três casos de estudo, verificou-se que as animações produzidas em linguagem VRML são visualmente mais atractivas que as animações inseridas no SimMechanics.

As animações do SimMechanics possuem formas rudimentares e com poucas opções em termos de visualização, enquanto que em VRML temos inúmeras opções em termos de criação de blocos, de mudanças do background e, inclusive, do ponto de vista que se quer ter da animação criada.

7 Bibliografia

- [1] Apontamentos das Aulas Práticas de Sistemas e Controlo II (2007-2008)
- [2] Carnegie Mellon – University of Michigan - Control Tutorials for Matlab – Bus Suspension - <http://www.engin.umich.edu/group/ctm/examples/susp/susp.html> (acedido em 02/04/2009)
- [3] Carnegie Mellon – University of Michigan - Control Tutorials for Matlab – Bus Suspension (PID Controller) - <http://www.engin.umich.edu/group/ctm/examples/susp/PID1.html> (acedido em 02/04/2009)
- [4] Carnegie Mellon – University of Michigan - Control Tutorials for Matlab – Bus Suspension (Root Locus Design) - <http://www.engin.umich.edu/group/ctm/examples/susp/rlocus1.html> (acedido em 02/04/2009)
- [5] Carnegie Mellon – University of Michigan - Control Tutorials for Matlab – Cruise Control - <http://www.engin.umich.edu/group/ctm/examples/cruise/cc.html> (acedido em 23/02/2009)
- [6] Carnegie Mellon – University of Michigan - Control Tutorials for Matlab – Cruise Control (PID Controller) - <http://www.engin.umich.edu/group/ctm/examples/cruise/ccPID.html> (acedido em 23/02/2009)
- [7] Carnegie Mellon – University of Michigan - Control Tutorials for Matlab – Inverted Pendulum - <http://www.engin.umich.edu/group/ctm/examples/pend/invpen.html> (acedido em 14/03/2009)
- [8] Carnegie Mellon – University of Michigan - Control Tutorials for Matlab – Inverted Pendulum (State-Space) - <http://www.engin.umich.edu/group/ctm/examples/pend/invSS.html> (acedido em 27/03/2009)
- [9] Hanselman, Duane C.. MATLAB tools for control system analysis and design / Duane C. Hanselman and Benjamin C. Kuo Englewood Cliffs (NJ) : Prentice Hall, 1995
- [10] Leonard, Naomi Ehrich. Using MATLAB to analyse and design control systems / Naomi Ehrich Leonard and William S. Levine Menlo Park (CA). Addison-Wesley, 1995
- [11] Melo, A. - Sistemas e Controlo Cap. 11 (2008)

- [12] Melo, A. - Sistemas e Controlo Cap. 3 (2008)
- [13] Physicall Modelling for Matlab – Demos – Modeling a Piston - <http://www.mathworks.com/products/demos/simmechanics/piston/index.html> (acedido em 27/11/2008)
- [14] Salgado, Norberto E. G. - Dissertação: Ajuste de Controladores por Optimização Não Linear Multiobjectivo- 2008
- [15] Simulink 3-D Animation - Getting Started with Simulink 3D Animation - <http://www.mathworks.com/products/3d-animation/demos.html> (acedido em 12/08/2008)
- [16] The MathWorks. SimMechanicsTM3 Getting Started Guide (2008) Cap. I
- [17] The MathWorks. SimMechanicsTM3 Getting Started Guide (2008) Cap. II
- [18] The MathWorks. SimMechanicsTM3 User's Guide (2008) Cap. I
- [19] The MathWorks. Simulink 3-D Animation 5 User's Guide (2009) Cap. I
- [20] VCLab - Laboratório Virtual de Sistemas de Controlo Online - <http://www.esr.ruhr-uni-bochum.de/VCLab/> (acedido em 18/10/2008)
- [21] VL-CSD- Laboratório Virtual de Sistemas de Controlo Online - <http://www.virtual-laboratories.com> (acedido em 25/10/2008)

8 Anexos

8.1 Anexo 1

Controlo da velocidade do carro

```

clear all;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Variáveis%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a=3; %comprimento do carro (m)
b=2; %largura do carro (m)
c=1.5; %altura do carro (m)
M=1000; %massa do carro (Kg)
f=50; %força de atrito (N)
ref=10; %velocidade a atingir - referencia (m/s)
It=[(M/12)*(b^2+c^2) 0 0; 0 (M/12)*(a^2+c^2) 0; 0 0 (M/12)*(a^2+b^2)]; %Matriz de Inércia
Kp=800; %constante de proporcionalidade do controlador PID
Ki=40; %constante de integração do controlador PID
Kd=100; %constante de derivação do controlador PID
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Resultados%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlador Atraso/Avanço%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MODELO LINEAR
%Controlador Avanço
K=800; %constante de ganho do controlador
A=0.1; %alfa
L=10; %lambda
sim('Carro_compensador_avanco_atraso_linearizado');
plot(tout,y,'r*');
xlabel('tempo');
ylabel('velocidade');
hold on;
grid on;
%Controlador Atraso
A=1; %alfa
L=0.1; %lambda
sim('Carro_compensador_avanco_atraso_linearizado');
plot(tout,y,'g. ');
xlabel('tempo');
ylabel('velocidade');
hold on;

```

```

grid on;
legend('Compensador Avanço','Compensador Atraso');
title('Controlo da velocidade do automóvel com controlador Atraso/Avanço (modelo linear)');
%MODELO VIRTUAL
%Controlador Avanço
A=0.1; %alfa
L=10; %lambda
sim('Carro_compensador_avanco_atraso');
plot(tout,y,'bx');
xlabel('tempo');
ylabel('velocidade');
hold on;
grid on;
%Controlador Atraso
A=1; %alfa
L=0.1; %lambda
sim('Carro_compensador_avanco_atraso');
plot(tout,y,'c-');
hold on;
grid on;

legend('Controlador Avanço no modelo linear','Controlador Atraso no modelo linear','Controlador
Avanço no modelo virtual','Controlador Atraso no modelo virtual','Location','SouthEast');
title('Controlo da velocidade do automóvel com controladores Atraso/Avanço');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlador PID%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
%MODELO VIRTUAL
sim('Carro_compensador_PID');
plot(tout,y,'kx');
xlabel('tempo');
ylabel('velocidade');
hold on;
grid on;
%MODELO LINEAR
sim('Carro_compensador_PID_linearizado');
plot(tout,y,'g');
xlabel('tempo');
ylabel('velocidade');
hold on;

```

```

grid on;
title('Controlo da velocidade do automóvel com controlador PID');
legend('Modelo virtual','Modelo linear','Location','SouthEast');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PID com perturbações%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
%MODELO VIRTUAL
sim('Carro_compensador_PID_perturbado');
plot(tout,y,'g'); %sistema mecanico
xlabel('tempo');
ylabel('velocidade');
hold on;
grid on;
%MODELO LINEAR
sim('Carro_compensador_PID_linearizado_perturbado');
plot(tout,y,'rx'); %sistema com função de transferencia
xlabel('tempo');
ylabel('velocidade');
hold on;
grid on;
legend('sistema virtual','sistema linear','Location','SouthEast');
title('Controlo da velocidade do automóvel com controlador PID com a variação da referência');

```

8.2 Anexo 2

Controlo do pêndulo invertido

```

clear all;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Variáveis%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=0.5; %massa do carro
m=0.2; %massa do pendulo
%I=0.006; %momento de inercia do pendulo
a=0.6; %comprimento do carro
b=0.2; %largura do carro
c=0.3; %altura do carro
ap=0.1; %comprimento do pendulo
bp=0.2; %largura do pendulo
cp=0.6; %altura do pendulo
g=9.8; %gravidade

```

```

l=cp/2; %distancia do pendulo ao seu centro de massa
It=[(M/12)*(b^2+c^2) 0 0; 0 (M/12)*(a^2+c^2) 0; 0 0 (M/12)*(a^2+b^2)];
%matriz de inércia do carro
Itp=[(m/12)*(bp^2+cp^2) 0 0; 0 (m/12)*(ap^2+cp^2) 0; 0 0 (m/12)*(ap^2+bp^2)];
%matriz de inércia do pêndulo
I=max(max(Itp)); %valor da Inércia do pêndulo necessário para o modelo linear
Kp=-60.4; %constante de proporcionalidade do controlador PID
Ki=-22.6; %constante de integração do controlador PID
Kd=-60.4; %constante de derivação do controlador PID
%Espaço de estados (referencia a Carnegie Mellon)
p = l*(M+m)+M*m*l^2; %denominador para a matriz A e B do espaço de estados
A = [0 1 0 0;
      0 -(l+m*l^2)*b/p (m^2*g*l^2)/p 0;
      0 0 0 1;
      0 -(m*l*b)/p m*g*l*(M+m)/p 0];
B = [ 0;
      (l+m*l^2)/p;
      0;
      m*l/p];
D = [1 0 0 0;
      0 0 1 0];
E = [0;
      0];
%Condição para uso da fórmula de Ackermann: (referencia a livro de A. Melo Sistemas e Controlo
Cap. 11)
%Verificar se o sistema é observável ou controlável
S=ctrb(A,B);cont=length(A)-rank(S)%completamente controlavel se =0
Q=obsv(A,D);obsev=length(A)-rank(Q)%completamente observavel se =0
%Condições pretendidas para o comportamento do sistema
tss=5;po=.04;%comportamento desejado
qsi=sqrt(log(po)^2/(pi^2+log(po)^2));wn=4/(tss*qsi);
[num,den]=ord2(wn,qsi);gd=tf(num,den);%step(gd);
raizes=roots(den);%raizes desejadas
P=[raizes(1,1),raizes(2,1),-20,-20];%pólos desejados
K=acker(A,B,P)%matriz de realimentação
%Cálculo da referência para o modelo com controlo por realimentação das
%variáveis de estado (referencia a Carnegie Mellon)
Cn=[1 0 0 0];
Nbar=rscale(A, B, Cn, 0, K);

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlador PID%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MODELO LINEAR
sim('inverted_pendulum_PID_linearizado');
subplot(2,1,1);
plot(time,angulo,'r-');
xlabel('tempo em segundos');
ylabel('angulo em graus')
grid on;
title('Controlo do ângulo do pendulo invertido com controlador PID (modelo linear)');
subplot(2,1,2)
sim('inverted_pendulum_PID_linearizado');
plot(time,deslocamento,'g-');
grid on;
xlabel('tempo em segundos');
ylabel('deslocamento em metros')
%MODELO VIRTUAL
sim('inverted_pendulum_PID_linearizado');
subplot(2,1,1);
plot(time,angulo,'r-');
xlabel('tempo em segundos');
ylabel('angulo em graus')
grid on;
hold on;
sim('inverted_pendulum_PID');
plot(time,angulo,'r:');
legend('Modelo Linear','Modelo Virtual','Location','SouthEast')
title('Controlo do ângulo do pendulo invertido com controlador PID');
subplot(2,1,2)
sim('inverted_pendulum_PID_linearizado');
plot(time,deslocamento,'g-');
grid on;
xlabel('tempo em segundos');
ylabel('deslocamento em metros')
hold on;
sim('inverted_pendulum_PID');
plot(time,deslocamento,'g:');
legend('Modelo Linear','Modelo Virtual')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlador por realimentação das variáveis de estado%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MODELO VIRTUAL
```



```

sim('inverted_pendulum_SS');
figure
subplot(2,1,1);
plot(time,angulo,'r-');
grid on;
axis([0 500 -2 2]);
xlabel('tempo em segundos');
ylabel('angulo em graus')
title('Controlo do pendulo invertido por realimentação das variáveis de estado (modelo virtual)');
subplot(2,1,2)
plot(time,deslocamento,'g-');
grid on;
xlabel('tempo em segundos');
ylabel('deslocamento em metros')
%MODELO VIRTUAL (na presença de perturbação)
sim('inverted_pendulum_SS_perturbado');
figure
subplot(2,1,1);
plot(time,angulo,'r-');
grid on;
axis([0 500 -2 2]);
xlabel('tempo em segundos');
ylabel('angulo em graus')
title('Controlo do pendulo invertido por realimentação das variáveis de estado na presença de uma
perturbação (modelo virtual)');
subplot(2,1,2)
plot(time,deslocamento,'g-');
grid on;
xlabel('tempo em segundos');
ylabel('deslocamento em metros')

```

8.3 Anexo 3

Controlo da suspensão de um autocarro

```

clear all;
close all;
%%%%%%%%%%%%Variáveis%%%%%%%%%%%%
R=0.4; %Raio do pneu+roda (m)
m=320; %Massa do conjunto suspensão+roda+pneu (Kg)

```

```

m2=320 %Massa do conjunto suspensão+roda+pneu (Kg)
Ir=[(2*m*R^2)/5 0 0; 0 (2*m*R^2)/5 0; 0 0 (2*m*R^2)/5];
%matriz de Inércia da roda+suspensao (correspondente a uma esfera)
M=2500; %Massa do bloco que representa um quarto de autocarro (Kg)
m1=2500;%Massa do bloco que representa um quarto de autocarro (Kg)
a=6/2; %Comprimento do bloco que representa um quarto de autocarro (m)
b=3/2; %Largura do bloco que representa um quarto de autocarro (m)
c=2; %Altura do bloco que representa um quarto de autocarro (m)
It=[(m/12)*(b^2+c^2) 0 0; 0 (m/12)*(a^2+c^2) 0; 0 0 (m/12)*(a^2+b^2)];
%matriz de Inércia do bloco que representa um quarto de autocarr
k1=80000; % Constante de elasticidade da mola da suspensão (N/m)
k2=500000; % Constante de elasticidade da mola do pneu (N/m)
b1 = 350; % Constante de atrito do amortecedor da suspensão (N.s/m)
b2 = 15020;% Constante de atrito do amortecedor do pneu (N.s/m)

var=50 %valor de tuning dos parâmetros do controlador PID
Kd=208025/var; %constante de derivação do controlador PID
Kp=832100/var; %constante de proporcionalidade do controlador PID
Ki=624075/var; %constante de integração do controlador PID
%MODELO LINEAR(referência a Carnegie Mellon)
%Determinação da função de transferência do sistema
nump=[(m1+m2) b2 k2]
denp=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+(b1*b2) (b1*k2)+(b2*k1) k1*k2]

num1=[-(m1*b2) -(m1*k2) 0 0]
den1=[(m1*m2) (m1*(b1+b2))+(m2*b1) (m1*(k1+k2))+(m2*k1)+(b1*b2) (b1*k2)+(b2*k1) k1*k2]

numf=num1;
denf=denp;
%resposta da função de transferencia a um impulso de 10cm sem controlador
impulse(0.1*num1,den1);
xlabel('Tempo');
ylabel('Posição X1-X2 (m)')
title('Posição da suspensão do autocarro na presença de uma perturbação sem controlador
(modelo linear)');
axis([0 30 -0.6 0.6]);

%MODELO VIRTUAL

```

```

%resposta do modelo virtual a um impulso de 10cm sem controlador
sim('suspensao.mdl');
figure
hold on;
plot(time, posicao, 'g');
grid on;
%axis([20 70 -0.1 0.1]);
xlabel('Tempo (s)');
ylabel('Posição X1-X2 (m)')
title('Posição da suspensão do autocarro na presença de uma perturbação sem controlador
(modelo virtual)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlador PID%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MODELO VIRTUAL
sim('suspensao_PID.mdl');
figure
hold on;
plot(time, posicao, 'g');
grid on;
xlabel('Tempo (s)');
ylabel('Posição X1-X2 (m)')
title('Controlo da suspensão do autocarro com controlador PID (modelo virtual)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Controlo por Cancelamento de polo zero%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MODELO VIRTUAL
%(referencia a Carnegie Mellon)
%rlocus com notch filter
z1=3+3.5i;
z2=3-3.5i;
p1=30;
p2=60;

numc=conv([1 z1],[1 z2]); %numerador do controlador
denc=conv([1 p1],[1 p2]); %denominador do controlador

%rlocus(conv(numc,numc),conv(denc,denc))
%axis([-70 70 -70 70]);
%devolve o lugar de raizes do sistema sem controlador
%calcular ganho do controlador
%[k,poles]=rlocfind(conv(numc,numc),conv(denc,denc))

```

```
% selected_point =
% -2.8555 - 3.4472i
%
% k = 1.3197e+008
%
% poles =
% 1.0e+002 *
% -0.6326 + 6.8371i
% -0.6326 - 6.8371i
% -0.0289 + 0.1306i
% -0.0289 - 0.1306i
% -0.0294 + 0.0364i
% -0.0294 - 0.0364i
k=1.3197e+008; %ganho do controlador
numc=k*numc; %numerador do controlador (com ganho k)
%MODELO VIRTUAL
%Controlador com cancelamento de polo zero
sim('suspensao_cancelamentopolo.mdl');
figure
hold on;
plot(time, posicao, 'g');
grid on;
xlabel('Tempo (s)');
ylabel('Posição X1-X2 (m)')
title('Controlo da suspensao do autocarro do modelo virtual com controlo por cancelamento de
polo zero');
```