Trabalho a realizar

1. Simulação do movimento livre de uma viatura

Este primeiro exercício proporciona uma introdução simples ao SIMULINK.

Considere a viatura da Figura 1-1 que se desloca em regime livre ao longo do eixo *y*, sem que lhe seja aplicada nenhuma força motora.

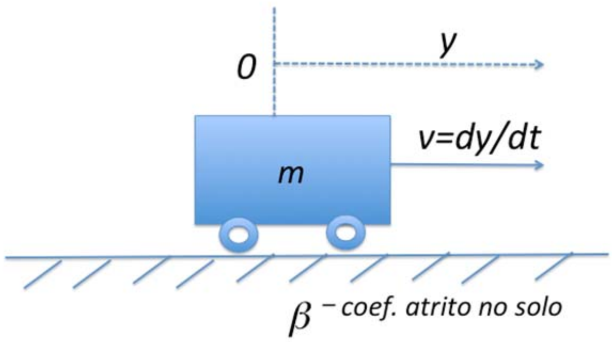

Fig. 1-1. Movimento de uma viatura ao longo do eixo *y*

1.1. (Em casa) Mostre que a evolução no tempo da velocidade *v=dy/dt* é modelada pela equação diferencial

$$m\frac{d}{dt}v(t) = -\beta v(t) \qquad v(0) = v_0$$

onde β é o coeficiente de atrito dinâmico no solo e *m* é a massa do veículo.

1.2. (Em casa) *Sem* resolver explicitamente a equação diferencial, trace qualitativamente o andamento no tempo da sua solução. Para tal, use os sinais da primeira e segundas derivadas de v em ordem ao tempo. Considere valores positivos e negativos para a condição inicial.

1.3. (Em casa) Resolva agora a equação diferencial. Para tal, observe que esta é do tipo "equação de variáveis separáveis" e que é equivalente a

$$\int_{v(0)}^{v(t)} \frac{1}{v}\,dv = -\frac{\beta}{m}\int_0^t dt$$

Calcule os integrais e resolva em ordem a *v(t).* Recorde que a primitiva em ordem a *v de 1/v* é $\ln(v)$.

1.4. (Em casa) Escreva a equação diferencial que rege a posição horizontal do veículo, y, e com base no resultado anterior determine a solução com condição inicial $y(0) = y_0$.

1.5. Crie um diagrama de blocos com o SIMULINK que represente a equação diferencial da alínea 1.1 e que permita simulá-la (ver abaixo instruções adicionais sobre como fazer isto). Em seguida modifique-o para incorporar o cálculo da posição e crie um *script* MATLAB que permita sucessivamente:

- Definir os parâmetros da simulação (m, β, v_0, y_0, tempo total da simulação, escalas de gráficos).
- Realizar a simulação. Se o nome do bloco SIMULINK for *carro.mdl*, o comando *sim('carro')* executa a simulação.
- Representar os gráficos necessários.

Usando este *script* e o diagrama SIMULINK que desenvolveu, represente graficamente os resultados da simulação e compare-os com a análise que fez nos pontos 1.3 e 1.4. Comece por considerar $m = 30\ Kg$ e $\beta = 5\ Nms^{-1}$, ou seja, uma constante de tempo $\frac{m}{\beta} = 6\ s$. Trace curvas da velocidade e da posição com dois outros valores da constante de tempo e para $v_0 = \pm 3\ ms^{-1}$, $y_0 = 5\ m$. Apresente os resultados comparativos de uma forma compacta mas clara, por exemplo, sobrepondo gráficos com *hold on/all* ou, preferencialmente, especificando múltiplas curvas em simultâneo no comando *plot*. Familiarize-se com as opções do comando *plot* para especificação do tipo de linha, cor e espessura, e também com comandos como *xlabel/ylabel*, *legend*, *axis/xlim/ylim*.

Nota: Criação de diagramas de blocos no SIMULINK

O SIMULINK é uma interface gráfica do MATLAB associada a programas para a integração numérica de equações diferenciais representadas por diagramas de blocos ligados entre si. Para além das informações dadas a seguir, sugere-se que consulte o manual do SIMULINK disponibilizado na documentação auxiliar.

Repare que (por exemplo) a equação diferencial

$$\frac{dv}{dt} = -5v$$

pode ser representada com um integrador e um ganho através do diagrama de blocos da Figura 1-2.

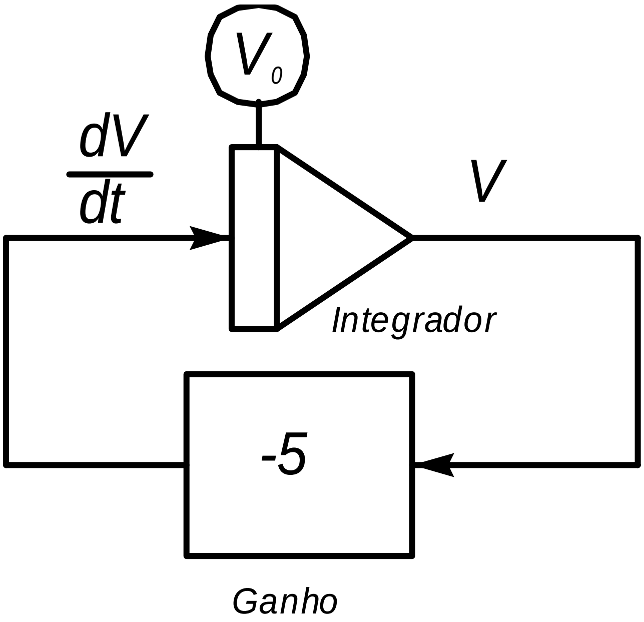

Fig. 1-2. Diagrama de blocos que representa uma equação diferencial linear.

O integrador tem à entrada a derivada e, por conseguinte, à saída o valor de v. Ligando a saída do ganho à entrada do integrador, força-se a que a derivada venha dada por $-5v$, o que corresponde à equação diferencial. A condição inicial é imposta através do valor inicial do integrador, definido na janela de propriedades deste bloco no SIMULINK.

De modo a usar o SIMULINK para definir este diagrama de blocos, comece por correr esta aplicação a partir do MATLAB. Para tal pode:

Simulink

- Dar o comando *simulink* ou, alternativamente
- Clicar no ícone do SIMULINK na régua de botões do MATLAB.

Aparece uma janela correspondente à biblioteca do SIMULINK (“SIMULINK library browser”), com o aspecto que se mostra na Figura 1-3.

Para criar um novo diagrama de blocos clique no ícone em cima à esquerda que representa uma folha de papel em branco. Surge uma janela em branco para onde poderá arrastar os blocos da biblioteca do SIMULINK.

Por exemplo, para obter um integrador, clique no sinal + de “continuous”, surgindo um conjunto de blocos, entre os quais o integrador. Clique no bloco integrador e, mantendo o botão esquerdo do rato premido, arraste o bloco para a página em branco em que está a desenvolver o diagrama de blocos. É criada uma cópia do bloco.

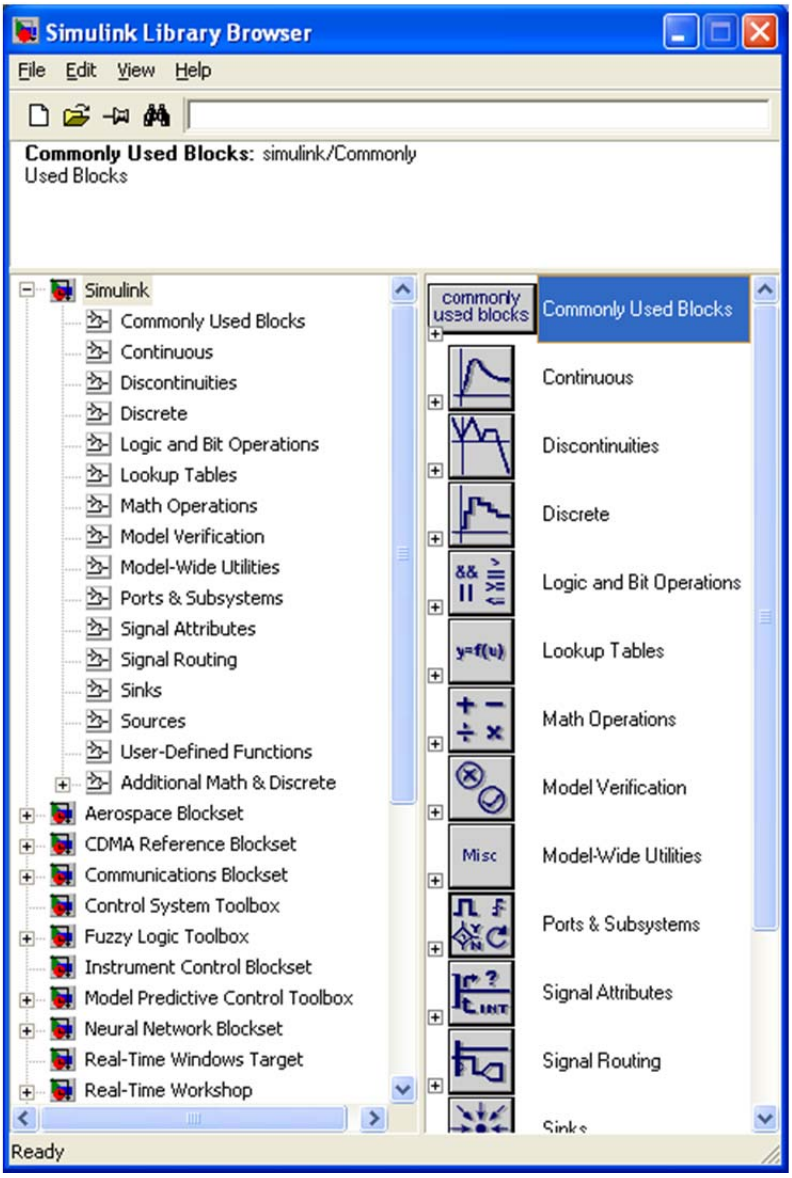

Fig. 1-3 – Janela do SIMULINK, com as bibliotecas de blocos disponíveis.

Clique duas vezes no integrador. Abre-se uma janela que permite a definição das propriedades do integrador. Defina a condição inicial no campo respectivo. Isto pode ser feito escrevendo um número ou o nome de uma variável global definida no espaço de trabalho ("workspace") do MATLAB.

Repita o mesmo procedimento para definir o bloco de ganho. Pode encontrar o bloco de ganho na biblioteca de "Math operations". Para dar uma melhor aparência ao diagrama a entrada do ganho deverá ficar à direita e a saída à esquerda. Para tal, clique no bloco "gain" com o botão direito e escolha sucessivamente "Format" e "Flip block"[1].

Interligue os blocos integrador e ganho de acordo com o esquema da Figura 1-2. Para tal, clique no ponto de saída de um bloco e arraste o cursor com o botão esquerdo do rato premido até ao ponto de entrada onde pretende fazer a ligação[2].

Para passar os sinais gerados para o espaço de trabalho do MATLAB uma possibilidade simples consiste em usar o bloco "To Workspace" do grupo "sinks". Edite este bloco para indicar o nome da variável. Além disso, no campo "Save format" escolha "array" (mais tarde explore as outras possibilidades). Para poder representar graficamente os sinais necessita de gerar um sinal de "tempo"[3]. Isto é feito com o bloco "clock" do grupo "sources". Ligue blocos "To Workspace" à saída do "clock" (tempo t) e à saída do integrador (velocidade v). tout

ligar blocos ctrl

[1] Alternativamente, rode o bloco em passos de 90º com as opções do menu ou CTRL+R.

[2] Para ligar rapidamente a saída de um bloco à entrada de outro seleccione primeiro o bloco fonte, depois prima CTRL e seleccione simultaneamente o bloco de destino.

[3] Habitualmente, o SIMULINK cria automaticamente no workspace do MATLAB uma variável *tout* com os instantes de referência da simulação.

tout

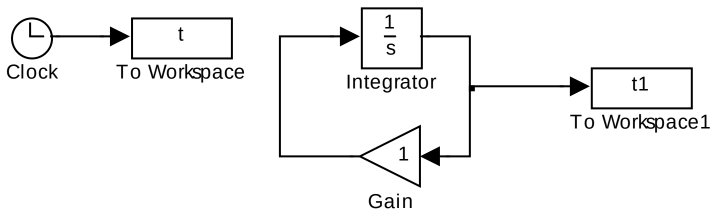

Fig. 1-4. Diagrama de blocos no SIMULINK.

A Figura 1-4 mostra o aspecto do diagrama de blocos na janela do SIMULINK.

No menu (topo da janela de projecto) escolha “Simulation” e depois “Configuration parameters”. Na janela que se abre escreva o tempo de simulação em “stop time” (pode ser uma variável que deverá ser definida no espaço de trabalho).

Grave o diagrama de blocos com um nome à sua escolha no directório de trabalho. A partir daqui, poderá chamar o diagrama de blocos directamente escrevendo o seu nome na linha de comando do MATLAB e fazendo “return”. O diagrama de blocos é aberto mas não é efectuada qualquer simulação.

só com apostrofo

Para efectuar a simulação clique no botão  na régua de comandos da janela de projecto ou dê o comando sim(*‘nome do diagrama’*) na linha de comandos do MATLAB. Se quiser também pode dar o comando sim(“*nome do diagrama”,”tempo de simulação*”). Use o help para ver os argumentos do comando sim (help sim). Habitue-se a fazer isto com outros comandos.

2. Modelo predador-presa

No segundo problema vamos considerar um conjunto de 2 populações em interacção predador-presa, modeladas como um sistema de 2 equações diferenciais não lineares de 1ª ordem (as derivadas são funções não lineares das incógnitas). Este exemplo é aproveitado para introduzir o tema da estimação de parâmetros.

O modelo em questão é dado por

$$\frac{d}{dt}N_1(t) = \delta_1 N_1(t) - \alpha_1 N_1(t) N_2(t)$$
$$\frac{d}{dt}N_2(t) = \delta_2 N_2(t) + \alpha_2 N_1(t) N_2(t)$$

onde N_1 e N_2 quantificam a abundância de cada uma das populações (valores sujeitos a algum tipo de normalização que não é importante aqui), δ_1 e δ_2 representam a diferença entre a taxa de natalidade e mortalidade para cada espécie, isoladamente, e α_1, α_2 definem o impacto que o nível de uma das populações tem na taxa de natalidade ou mortalidade da outra.

2.1. (Em casa) Supondo que os coeficientes α_1 e α_2 são positivos identifique qual das populações corresponde ao predador e qual é a presa. Determine pontos de equilíbrio e trace o andamento qualitativo das soluções em função do tempo. Discuta qualitativamente quais as condições nos sinais algébricos de δ_1 e δ_2 para que se obtenham soluções oscilatórias, por oposição a soluções em que uma das espécies se extingue e/ou a outra cresce indefinidamente.

2.2. Recorrendo ao SIMULINK e ao MATLAB resolva numericamente o sistema de equações com $\alpha_1 = \alpha_2 = 1$ para diferentes valores das condições iniciais e de δ_1, δ_2. Crie um diagrama de blocos e um script para efectuar a simulação. Compare os resultados com os vários tipos de soluções qualitativas que obteve em 2.1.

Sugestão:

Use blocos *fcn* do grupo “User defined functions” para introduzir directamente as expressões das derivadas calculadas a partir de N_1, N_2 e dos restantes parâmetros.

2.3. Represente as soluções oscilatórias que obteve no estudo anterior em espaço de fase (N_1, N_2) com valores fixos à sua escolha para $\delta_{1,2}$ e diferentes valores das condições iniciais para $N_{1,2}$. Confirme o ponto de equilíbrio calculado na alínea 2.1 e indique dois conjuntos de condições iniciais que conduzam a evoluções idênticas do sistema, a menos de um deslocamento temporal.

2.4. Carregue no MATLAB o ficheiro *presas.mat,* que fornece valores, ao longo do tempo (com algum ruído adicionado), para uma população de presas caracterizada por $\delta = 3.1$, $\alpha = 1.4$ e $N(0) = 4$. Suponha que os dados no ficheiro foram obtidos monitorizando a população em causa na natureza e que, por serem mais esquivos, não foi possível efectuar estudo semelhante para os predadores. Para estes últimos conhece-se apenas o parâmetro $\delta = -1.5$ com base em estudos realizados em cativeiro, desconhecendo-se a dimensão da população e o coeficiente α que reflecte o efeito das presas na abundância de predadores. Pretende-se inferir as características da população de predadores no modelo para explicar a evolução observada das presas, de acordo com os seguintes passos:

a) Escreva um script em MATLAB que permita sobrepor, no mesmo gráfico, a curva fornecida e a solução gerada pelo SIMULINK correspondente a um par de valores $N(0)$ e α para a equação dos predadores. Por tentativa e erro tente descobrir os valores dos parâmetros $N(0)$ e α dos predadores que melhor explicam a evolução temporal observada das presas.

b) Nesta questão pretende-se estimar os valores de $N(0)$ e α que levam a um ajuste "óptimo" da curva gerada pela solução do sistema de equações diferenciais aos pontos fornecidos. A expressão "óptimo" pode ser entendida em vários sentidos. Neste caso, iremos considerar como medida da qualidade do ajuste o *máximo valor absoluto das diferenças*[4] entre os valores fornecidos e os correspondentes valores calculados, nos mesmos instantes de tempo. Discuta a adequação/unicidade desta estratégia à luz das suas respostas à alínea 2.3.

[4] Habitualmente designado por norma l_∞ do vector de erro entre as previsões e as observações.

Escreva uma função MATLAB que permita calcular o erro em função de $N(0)$ e α (fornecido como vector de 2 componentes na entrada da função). Internamente, o diagrama de SIMULINK é invocado com a função *sim*. Dependendo das versões de MATLAB, as opções de configuração do simulador podem ser estabelecidas com argumentos opcionais de *sim*, ou usando o comando *simset*. Teste a devolução dos resultados de simulação usando blocos *To Workspace*, como anteriormente, ou *OutPorts*.

Suponha em seguida que $N(0)$ e α estão compreendidos entre valores mínimos e máximos conhecidos. Recorrendo à função desenvolvida acima escreva um script para representar graficamente o erro em função da grelha usada para $N(0)$ e α. Use dois ciclos de *for* para gerar essa grelha (os comandos *linspace* ou *meshgrid* poderão ser úteis) nos intervalos estabelecidos para os parâmetros $N(0)$ e α. Para cada "ponto" da grelha (*i. e.*, para cada par de valores de $N(0)$ e α) integre o sistema de equações diferenciais com esses parâmetros e calcule o erro em relação às observações. Existem várias possibilidades para sincronizar no tempo as previsões com as observações: Passagem dos instantes pretendidos à função *sim*; ajuste do parâmetro *sampling time* nos blocos de saída do SIMULINK; escolha de um *solver* de passo fixo; pós-processamento no MATLAB para interpolação dos resultados de simulação com a função *interp1*.

A superfície de erro assim gerada pode ser visualizada com comandos do tipo *mesh, surf, contour,* ou *surfc.* Parametrize os comandos de forma a representar os eixos correspondentes a $N(0)$ e α com os valores numéricos correctos. A execução deste script implica um tempo que começa a ser apreciável (depende do número de pontos da grelha que definiu). Pode ter uma informação do tempo que ainda falta escrevendo o comando *waitbar(n/nmax)* no fim do ciclo *for* externo, em que *n* é o contador do ciclo e *nmax* é o número total de vezes que o ciclo é executado (limite superior de *n*).

Analise a forma da superfície de erro que obteve (pode ser conveniente visualizá-la em escala logarítmica) e discuta a facilidade/dificuldade expectável na estimação de cada um dos dois parâmetros desconhecidos.

Através da observação da representação gráfica das curvas de nível, vá restringindo sucessivamente os intervalos de valores que considera para $N(0)$ e α e repetindo o procedimento anterior para determinar aproximadamente o mínimo global da função.

c) Faça agora a estimativa dos parâmetros de evolução da população de predadores usando um método de optimização. Na alínea anterior escreveu uma função que lhe permite calcular o erro para valores específicos de $N(0)$ e α. Use agora as funções *fminunc* ou *fminsearch* da *Optimization Toolbox* do MATLAB para coordenar as chamadas à função que desenvolveu, e assim localizar o mínimo de forma mais eficiente que a busca exaustiva da alínea anterior. Como é que esta estimativa se compara com a que obteve na alínea b)? Tente ilustrar a convergência deste método para outras soluções (mínimos locais), dependendo da escolha dos valores iniciais para $N(0)$ e α, e comente os resultados.

d) Para verificar o resultado (validar o modelo), simule o sistema de equações diferenciais para os valores de $N(0)$ e α correspondentes ao erro mínimo e sobreponha à solução os pontos da curva fornecida para as presas. Marque os pontos fornecidos como círculos e as curvas calculadas com linhas contínuas.

3. Sistema caótico

Neste problema simula-se o pêndulo duplo representado na Figura 1-5, constituído por duas barras homogéneas articuladas. Este exemplo ilustra um comportamento do tipo caótico em sistemas mecânicos, onde pequenas variações nas condições iniciais podem conduzir a evoluções temporais do sistema substancialmente diferentes. Matematicamente, este comportamento *resulta do carácter não linear particular das equações que descrevem a dinâmica do sistema.*

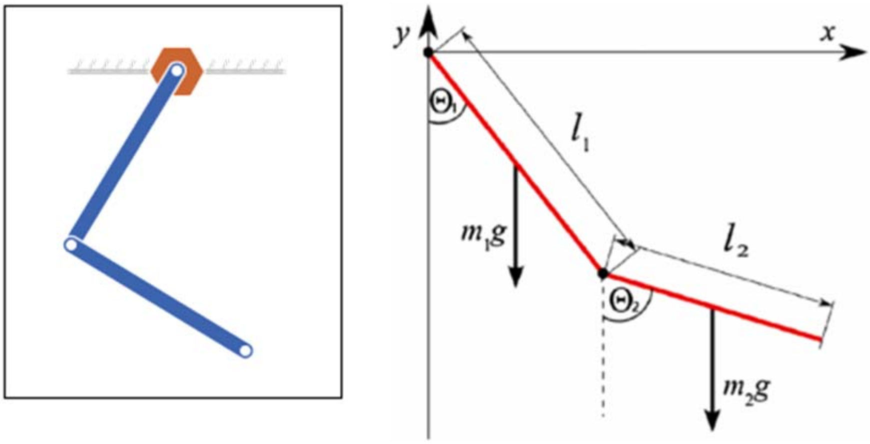

Fig. 1-5. Pêndulo duplo.

Para $m_1 = m_2 = m$ e $l_1 = l_2 = l$ o pêndulo pode ser descrito pelo seguinte sistema de quatro equações diferenciais não lineares de primeira ordem:

$$\dot{\theta}_1 = \frac{6}{ml^2}\frac{2p_1 - 3\cos(\theta_1 - \theta_2)p_2}{16 - 9\cos^2(\theta_1 - \theta_2)}$$

$$\dot{\theta}_2 = \frac{6}{ml^2}\frac{8p_2 - 3\cos(\theta_1 - \theta_2)p_1}{16 - 9\cos^2(\theta_1 - \theta_2)}$$

$$\dot{p}_1 = -\frac{1}{2}ml^2\left[\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + 3\frac{g}{l}\sin\theta_1\right]$$

$$\dot{p}_2 = -\frac{1}{2}ml^2\left[-\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + \frac{g}{l}\sin\theta_2\right]$$

onde g é a aceleração da gravidade e as variáveis p_1, p_2 se relacionam com os ângulos e velocidades angulares através de

$$p_1 = \frac{1}{6}ml^2\left[8\dot{\theta}_1 + 3\dot{\theta}_2\cos(\theta_1 - \theta_2)\right]$$

$$p_2 = \frac{1}{6}ml^2\left[2\dot{\theta}_2 + 3\dot{\theta}_1\cos(\theta_1 - \theta_2)\right]$$

3.1. Crie o modelo do sistema em SIMULINK. É aconselhável a utilização de um bloco *fcn* para gerar de modo compacto cada uma das 4 derivadas a partir das variáveis de estado[5] θ_1, θ_2, p_1, p_2. Este esquema pode ser implementando compactamente de forma semelhante à Figura 1-4 usando

[5] Pode também incluir nas entradas de um bloco as saídas de outros.

ligações vectoriais[6]. Tome $m = 1\,Kg$, $l = 0.5\,m$, $g = 9.8\,ms^{-2}$, e considere repouso inicial no cálculo das condições iniciais $p_1(0)$, $p_2(0)$.

Exporte para o MATLAB os ângulos calculados. Para confirmar a plausibilidade física do movimento simulado visualize uma animação simples do movimento das barras[7]. Não há necessidade de a incluir no relatório do trabalho, mas esta é muito conveniente como ferramenta para detectar possíveis erros na implementação do sistema dinâmico do pêndulo duplo. Use o sistema de coordenadas da Figura 1-5.

3.2. No regime de fraca amplitude ($\theta_1(0)$ e $\theta_2(0)$ pequenos) represente a trajectória do sistema no plano (θ_1, θ_2), verificando que esta descreve uma curva de Lissajous[8]. Aumente a amplitude da deflexão inicial do pêndulo e verifique que esta figura se torna gradualmente menos regular.

3.3. (Em casa) Dado um ponto (x, y) tal que $\sqrt{x^2 + y^2} \leq 2l$ obtenha expressões para os ângulos θ_1 e θ_2 de forma a que a ponta do pêndulo se situe nessa posição. Note que na maioria dos casos existem duas soluções para o problema que correspondem a configurações simétricas da articulação do pêndulo. Opte pela que apresentar maior energia potencial gravítica.

3.4. Considere uma grelha de pontos no plano correspondentes às posições iniciais da ponta do pêndulo. Para cada uma dessas posições, e tomando $\dot{\theta}_1(0) = 0°\ s^{-1}$, $\dot{\theta}_2(0) = -30°\,s^{-1}$, obtenha as condições iniciais de θ_i, p_i e simule o sistema até $t = 250\,s$. Determine o tempo que decorre até uma das barras "fazer um *looping*" (na maioria dos casos pode-se testar se o respectivo ângulo passa de $-\pi$ a π ou vice-versa) e represente como uma imagem os logaritmos desses tempos em função de x e y (usando, por exemplo, o comando *pcolor*). Associe o valor NaN aos pontos da grelha

[6] Pode combinar várias ligações escalares numa ligação vectorial usando blocos *mux*, e realizar a operação inversa usando blocos *demux*.

[7] Numa versão simples implemente um ciclo percorrendo os resultados de simulação, e em cada iteração use os comandos *plot*, *line*, *axis* e *axis('square')* para desenhar o pêndulo. Use *pause* para temporizar adequadamente a passagem de um instante de simulação para o seguinte.

[8] http://en.wikipedia.org/wiki/Lissajous_curve

inacessíveis ao pêndulo duplo ou onde não ocorre nenhum *looping* no horizonte considerado. Discuta os resultados obtidos.

Seleccione 3 configurações representativas em que os tempos de *looping* estejam contidos nos intervalos $[0\ 30]\ s$, $[30\ 100]\ s$ e $[100\ 250]\ s$ e examine detalhadamente a evolução do pêndulo no tempo, confirmando o cálculo automático dos tempos de *looping*. Apresente os gráficos com a evolução temporal de θ_1 ou θ_2, conforme apropriado, assinalando o instante em que ocorre o *looping*.