9

I used [1] in making this assignment when finished I compared initial values with Prajish Kumar (4743873).	**Rick Staa, #4511328** **HW set 2 due 06/03/2018** **ME41060**

1 Statement of entregity

my homework is completely in accordance with the Academic Integrity

2 Problem Statement

In this assignment, we were given the same model as in assignment 1 (see Figure 1). This model consists of a double pendulum attached to the ground. For this pendulum, we were asked to compute the equations of motion and possible constraint equations. Following we were asked to calculate the full state vector out of the known angles and angular velocities (inputs of our mode). Contrary to assignment 1 in this assignment we were asked to use the systematic approach explained in [1]. This report shows how to complete the given assignment.

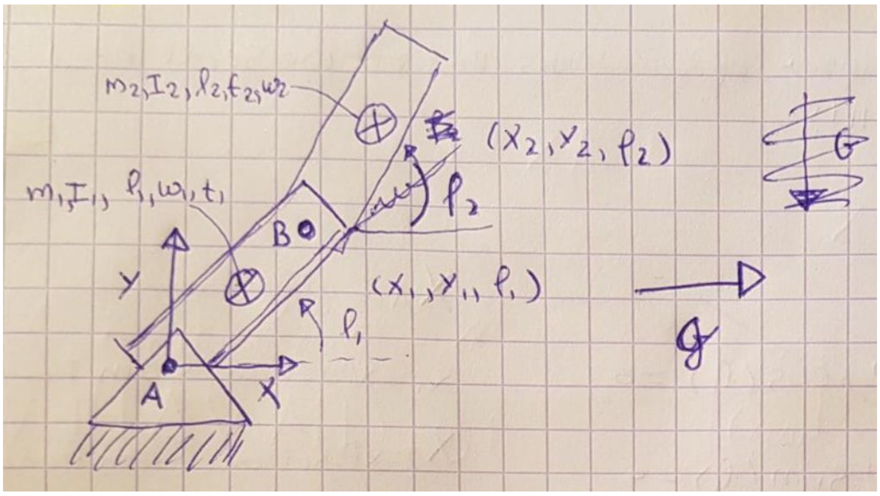

Figure 1: Sketch of a double pendulum with 2 DOF. In this sketch l1,l2 are L2, L1

2.1 Given model Parameters

- $L_1 = L_2 = L = 0.055\, m$
- $p_1 = p_2 = 1180 \frac{kg}{m^3}$
- $w_1 = w_2 = 0.05\, m$
- $t_1 = t_2 = 0.004\, m$
- $g = 9.81 \frac{N}{kg}$

2.2 Calculated parameters

For solving the problem, we need to calculate the mass and the moments of inertia of both rigid bodies. The moment of inertias is taken relative to the CM.

- $m_1 = p_2 * w_1 * t_1 * h_1 = 0.1298\, kg$
- $m_2 = p_2 * w_2 * t_2 * h_2 = 0.1298\, kg$

- $I_1 = I_2 = \int_{-\frac{L}{2}}^{\frac{L}{2}} r^2 dm = \int_{-\frac{L}{2}}^{\frac{L}{2}} \frac{x^2 M}{L} dx = \frac{M}{L}\int_{-\frac{L}{2}}^{\frac{L}{2}} x^2 dx = \frac{M}{L}\left[\frac{x^3}{3}\right]_{-\frac{L}{2}}^{\frac{L}{2}} = \frac{ML^2}{12}$

2.3 Conventions

Our model has the angles and angular velocities of both segments (displayed in radians) as its inputs.

$$u_0 = [\phi_1\ \Phi_2\ \dot{\phi}_1\ \dot{\phi}_2]$$

In the systematic approach, the model has the second derivative of the state $\ddot{x}$ and the Lagrange multipliers of the constraint equations as its outputs. The nature of these Lagrance multipliers will be discussed in section 0.

$$X = \begin{bmatrix} \ddot{x} \\ \lambda_k \end{bmatrix}$$

Where k = 1 … m, with m the number of constraints.

$$F_c = (H_A\ V_A\ H_B\ V_B)^T$$

$$\ddot{x} = \left(\ddot{x}_1\ \ddot{y}_1\ \ddot{\phi}_1\ \ddot{x}_2\ \ddot{y}_2\ \ddot{\phi}_2\right)^T$$

In our output matrix X the linear velocities are given in m/s^2 and the angular velocities are in rad/s^2. The Lagrange multipliers are given in $Newtons$ or $Newton \cdot meters$, depending on what kind of constraint is used [1].

3 A systematic approach for deriving the DAE

As explained in [1] in the systematic approach we make use of the concept of virtual power to derive the DAE of motion of the double pendulum.

$$\delta P = \sum F_i \delta \dot{\boldsymbol{x}}_i \tag{1}$$

The concept of virtual work states that "A mechanical system is in equilibrium if the virtual power is zero for all virtual velocities that satisfy the constraints". Since in this course we are not interested in static equilibrium we need to use d'Alembert forces to use this concept with dynamic equilibrium. With these forces we can rewrite equation **1** to:

$$\delta P = \left(F_i - M_{ij}\dot{\boldsymbol{x}}_j\right) \cdot \delta \dot{\boldsymbol{x}}_i \tag{2}$$

To make sure that all the virtual velocities we use to satisfy the constraints we need make use of the constraint equations C(x) derived in assignment 1.

$$C(\boldsymbol{x}) = \begin{bmatrix} x_1 - \frac{1}{2}Lcos(\phi_1) \\ y_1 - \frac{1}{2}Lsin(\phi_1) \\ x_2 - x_1 - \frac{1}{2}Lcos(\phi_1) - \frac{1}{2}Lcos(\phi_2) \\ y_2 - y_1 - \frac{1}{2}Lsin(\phi_1) - \frac{1}{2}Lsin(\phi_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

We need to differentiate these constraints with respect to time to determine the velocities that satisfy the constraints. When doing this, we get the following result:

$$\dot{C}_k = C_{k,i} \cdot \dot{\boldsymbol{x}}_i = \begin{bmatrix} 1 & 0 & \frac{1}{2}Lsin(\phi_1) & 0 & 0 & 0 \\ 0 & 1 & -\frac{1}{2}Lcos(\phi_1) & 0 & 0 & 0 \\ -1 & 0 & \frac{1}{2}Lsin(\phi_1) & 1 & 0 & \frac{1}{2}Lsin(\phi_2) \\ 0 & -1 & -\frac{1}{2}Lcos(\phi_1) & 0 & 1 & -\frac{1}{2}Lcos(\phi_2) \end{bmatrix} \cdot \dot{x}_i = 0 \tag{4}$$

In which $C_{k,i}$ is called the jacobian of the matrix $C(x)$. Following we replace the real velocities $\dot{x}_i$ with the virtual velocities and include these equality constraint equations in the virtual power expression by using lagrange multipliers. Finaly by taking into account that the virtual power relationship has to hold for all virtual velocities we get, the following equation as the end result.

$$F - M\ddot{\boldsymbol{x}} - C_{k,i}\lambda_k = \boldsymbol{0} \tag{5}$$

We now have the constrained equations of motion, which describe the dynamic equilibrium of the system, expressed in the unknown centre of mass accelerations $\dot{\boldsymbol{x}}_j$ and the unknown Lagrange multipliers λ_k. We now need to add the constraint equations which glue the individual bodies back to the system. Since the unknowns are the accelerations similar as we did in assignment 1 we need to differentate these constraints twise with respect to time.

$$\ddot{C}_k = C_{k,j}(\boldsymbol{x}_l) \cdot \ddot{\boldsymbol{x}} \ + \ C_{i,jk}(\boldsymbol{x}_l)\dot{\boldsymbol{x}}_j\dot{\boldsymbol{x}}_k \tag{6}$$

This differentiation was done in MATLAB with the use of the symbolic toolbox. After deriving the second derivative of the constraint, we now have all the ingredients to solve the DEA's. To do this, we need to write the dynamic equilibrium equations (5) with the acceleration constraints (**6**) in a matrix-vector product.

$$\begin{pmatrix} \boldsymbol{M} & \boldsymbol{C}_{,x}^{T} \\ \boldsymbol{C}_{,x} & \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ -\boldsymbol{C}_{,xx}\dot{\boldsymbol{x}}\dot{\boldsymbol{x}} \end{pmatrix} \tag{7}$$

Error! Reference source not found.Error! Reference source not found.555 Following as we did in assignment 1 since we now have 10 unknowns and 10 equations we solve the matrix-vector product defined above to get the output state X at the three in the assignment specified configurations.

1

$$\begin{bmatrix} \ddot{\boldsymbol{x}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{pmatrix} \boldsymbol{M} & \boldsymbol{C}_{,x}^{T} \\ \boldsymbol{C}_{,x} & \boldsymbol{0} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{F} \\ -\boldsymbol{C}_{,xx}\dot{\boldsymbol{x}}\dot{\boldsymbol{x}} \end{pmatrix}$$

We do this with the help of the initial state which is defined as:

$$x_0 = [\phi_1 \ \phi_2 \ \dot{\phi}_1 \ \dot{\phi}_2]$$

The MATLAB code implementing all above-described operations can be found in Appendix A.

3.1.1 Both bars vertical up and zero speed

With $x_0 = [0.5\,\pi\ 0.5\pi\ 0\ 0]$ we get the following result:

$\ddot{x}_1$	6.31 m/s^2
$\ddot{y}_1$	0.00 m/s^2
$\ddot{\phi}_1$	-22.93 rad/s^2
$\ddot{x}_2$	10.51 m/s^2
$\ddot{y}_2$	0.00 m/s^2
$\ddot{\phi}_2$	7.64 rad/s^2
λ_1	0.36 N
λ_2	0.00 N
λ_3	-0.09 N
λ_4	0.00 N

1/3

These results are the same as the result obtained in assignment 1. However, the Lagrange multipliers are the negatives of the reaction forces at the joints A and B. Meaning that a positive λ_a is equivalent to a Reaction force in the negative direction. The interpretation of the other Lagrange multipliers is similar to this. This can be verified by comparing the equations of motion derived in assignment 1 with the dynamic equilibrium equations derived in this assignment.

Assignment 1 (Equations of motion)	**Assignment 2 (dynamic equilibrium equations)**
$m\ddot{x}_1 + \lambda_1 - \lambda_3 = mg$ $I_{CM}\ddot{\phi}_1 + \frac{1}{2}L\lambda_1 + \frac{1}{2}L\lambda_3 = 0$	$m\ddot{x}_1 - H_A + H_B = mg$ $-I_{CM}\,\ddot{\phi}_1 - \frac{1}{2}LH_A - \frac{1}{2}LH_B = 0$

3.1.2 Both bars horizontal to the right and zero speed

With $x_0 = [0\ 0\ 0\ 0]$ we get the following result:

$\ddot{x}_1$	0.00 m/s^2
$\ddot{y}_1$	0.00 m/s^2
$\ddot{\phi}_1$	0.00 rad/s^2
$\ddot{x}_2$	0.00 m/s^2
$\ddot{y}_2$	0.00 m/s^2
$\ddot{\phi}_2$	0.00 rad/s^2
λ_1	2.55 N
λ_1	0.00 N
λ_1	1.27 N
λ_1	0.00 N

1/3

Again, equal to assignment 1 but with $\lambda_k = -F_{reaction}$.

3.1.3 Both bars horizontal and with an initial angular speed on both bars of 60 rpm.

Since 60 rmp is equal to 2π rad/s we use $x_0 = [0\ 0\ 2\pi\ 2\pi]$ as the initial state we then the following result:

1/3

$\ddot{x}_1$	-10.8566 m/s^2
$\ddot{y}_1$	0.00 m/s^2
$\ddot{\phi}_1$	0.00 rad/s^2
$\ddot{x}_2$	-32.5697 m/s^2
$\ddot{y}_2$	0.00 m/s^2
$\ddot{\phi}_2$	0.00 rad/s^2
λ_1	8.1834 N
λ_2	0.00 N
λ_3	5.5009 N
λ_4	0.00 N

Again, equal to assignment 1 but with $\lambda_k = -F_{reaction}$. The velocities were also derived as was stated in assignment 1. The MATLAB script implementing this can be found in appendix A. The resulting velocities are:

$\dot{x}_1$	0.00 m/s^2
$\dot{y}_1$	1.7279 m/s^2
$\dot{x}_2$	0.00 rad/s^2
$\dot{y}_2$	5.1836 m/s^2

3.2 Adding extra constraint to the system

In this question we were asked to add an extra constrain that constraints the distal end of bar 2 to a vertical line going through the origin (assumed to be in point A). The new set up is depicted in figure 2. Because we are using the systemetic approach we can simply add this equation to our constraint equation vector $C(\boldsymbol{x})$ and repeat the steps explained in section 3. The constraint vector now becomes:

$$C(\boldsymbol{x}) = \begin{bmatrix} x_1 - \frac{1}{2}Lcos(\phi_1) \\ y_1 - \frac{1}{2}Lsin(\phi_1) \\ x_2 - x_1 - \frac{1}{2}Lcos(\phi_1) - \frac{1}{2}Lcos(\phi_2) \\ y_2 - y_1 - \frac{1}{2}Lsin(\phi_1) - \frac{1}{2}Lsin(\phi_2) \\ x_2 + \frac{1}{2}Lcos(\phi_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

This leads to the following Jacobean:

$$C_{k,i} = \begin{bmatrix} 1 & 0 & \frac{1}{2}Lsin(\phi_1) & 0 & 0 & 0 \\ 0 & 1 & -\frac{1}{2}Lcos(\phi_1) & 0 & 0 & 0 \\ -1 & 0 & \frac{1}{2}Lsin(\phi_1) & 1 & 0 & \frac{1}{2}Lsin(\phi_2) \\ 0 & -1 & -\frac{1}{2}Lcos(\phi_1) & 0 & 1 & -\frac{1}{2}Lcos(\phi_2) \\ 0 & 0 & 0 & 1 & 0 & -\frac{1}{2}Lsin(\phi_2) \end{bmatrix} \tag{9}$$

And the following constraint equations:

$$C_{k,jl}\dot{\boldsymbol{x}}_j\dot{\boldsymbol{x}}_l = \begin{bmatrix} \frac{1}{2}Lcos(\phi_1)\dot{\phi_1}^2 \\ \frac{1}{2}Lsin(\phi_1)\dot{\phi_1}^2 \\ \frac{1}{2}Lcos(\phi_1)\dot{\phi_1}^2 + \frac{1}{2}Lcos(\phi_2)\dot{\phi_2}^2 \\ \frac{1}{2}Lsin(\phi_1)\dot{\phi_1}^2 + \frac{1}{2}Lsin(\phi_2)\dot{\phi_2}^2 \\ \frac{1}{2}Lcos(\phi_2)\dot{\phi_2}^2 \end{bmatrix} \tag{10}$$

The MATLAB code for doing this can be found in appendix A.

3.2.1 Results for the case when both bars are vertically up

With $x_0 = [0.5\pi\ 0.5\pi\ 0\ 0]$ we get the following result:

1/2

$\ddot{x}_1$	$7.36\ m/s^2$
$\ddot{y}_1$	$0.00\ m/s^2$
$\ddot{\phi}_1$	$-26.7545\ rad/s^2$
$\ddot{x}_2$	$7.36\ m/s^2$
$\ddot{y}_2$	$0.00\ m/s^2$
$\ddot{\phi}_2$	$26.7545\ rad/s^2$
λ_1	0.3183 N
λ_2	0.00 N
λ_3	0 N
λ_4	0.00 N
λ_5	0.3183 N

As the physical interpretation of the linear acceleration, angular acceleration and reaction forces was already explained in assignment 4 we will only focus on the 5th Lagrange multiplier. Since the new constraint allows movement in the y direction but prohibits movement in the x direction the 5th Lagrange multiplier needs to be a horizontal reaction force keeping point C on the same x position. Since the reaction force is the (real) conjugate of the LaGrange multiplier this reaction force is negative and points in the negative x direction. This is as expected since gravity works on the pendulum in the horizontal direction.

3.2.2 Results for the case when both bars are vertical up and bar 1 has a clockwise initial angular speed of ω = 60 rpm

With $x_0 = [0.5\pi\ 0.5\pi - 2\pi\ \cancel{0}]$ we get the following result: Speed bar 2 depended on speed bar 1

$\ddot{x}_1$	$7.36\ m/s^2$
$\ddot{y}_1$	$-10.86\ m/s^2$
$\ddot{\phi}_1$	$-26.7545\ rad/s^2$
$\ddot{x}_2$	$7.36\ m/s^2$
$\ddot{y}_2$	$\cancel{-21.7131\ m/s^2}$
$\ddot{\phi}_2$	$26.7545\ rad/s^2$

λ_1	0.3183 N
λ_2	~~4.2275 N~~
λ_3	0 N
λ_4	~~2.8184 N~~
λ_5	0.3183 N

All these results can be explained by looking at the constrained equations of motion. Since $\lambda_3 = 0$ from equation (8) we see that $\ddot{x}_2$ becomes equal to $\ddot{x}_1$. Further the vertical accelerations are negative because the vertical reaction forces in A and B are in the negative direction. The opposite signs of the angular accelerations can be explained by looking at the horizontal reaction forces in point A and C. Since λ_1 is positive H_A it creates a negative moment of body 1 around its CM. Contrary H_c creates a positive moment of body 2 around its CM.

3.2.3 Results for the case when both bars are horizontal and have zero speeds, but an additional force of Fy = 10 N is applied in B

Following we were asked to introduce a force to end of bar 1 this can be done by introducing a moment about the CM of bar 1. As a result our F vector changes to:

$$F = \left[m1g \ 0 \ 10\frac{L}{2} \ m2g \ 0 \ 0\right]^T \tag{11}$$

The MATLAB code for implementing this can be found in Appendix A. With $x_0 = [0\ 0\ 0\ 0]$ we get the following result:

Warning: Matrix is singular to working precision.

$\ddot{x}_1$	NaN m/s^2
$\ddot{y}_1$	NaN m/s^2
$\ddot{\phi}_1$	$NaN\ rad/s^2$
$\ddot{x}_2$	$NaN\ m/s^2$
$\ddot{y}_2$	$NaN\ m/s^2$
$\ddot{\phi}_2$	$NaN\ rad/s^2$
λ_1	NaN N
λ_2	NaN N
λ_3	NaN N
λ_4	NaN N
λ_5	inf N

½

As seen from the output MATLAB cannot calculate a unique solution to the problem we gave it. This is because with the given initial state the Jacobian of the constraint equations gets a rank of 4 while its dimension is 5. Meaning that it is a singular matrix and the rows and or columns are linear independent. As a result, we only have four independent variables infinite possibilities are possible. Further as the rank of the null space is 2 it means that in this configuration the system has 2 DOF. These effects are caused by the fact that both the C joint and the A joint are aligned together in the origin and a rotation of the two pendulums can be described in both the C joint as the A joint.

Missed the possible movement

4 References

[1] A. L. Schwab, “Virtual power and Lagrance multipliers,” in *Multibody Dynamics*, Delft, The Netherlands: TU Delft, 2018.

1

Appendix 1

```
%% MBD_B: Assignment 2 - Double pendulum systemetic approach
%  Rick Staa (4511328)
%  Last edit: 05/03/2018
clear all; close all; clc;

%% Script settings
parms.booleans.ex_constr = 0;                   % Put on 1 if you want to enable the
extra constraint

%% Parameters
% Segment 1
parms.L1      = 0.55;                  % [m]
parms.w1      = 0.05;                  % [m]
parms.t1      = 0.004;                 % [m]
parms.p1      = 1180;                  % [kg/m^3]
parms.m1      = parms.p1 * parms.w1 * parms.t1 * parms.L1;          % [kg]
parms.I1      = (1/12) * parms.m1 * parms.L1^2;                     % [kg*m^2]

% Segment 2
parms.L2      = 0.55;                  % [m]
parms.w2      = 0.05;                  % [m]
parms.t2      = 0.004;                 % [m]
parms.p2      = 1180;                  % [m]
parms.m2      = parms.p2 * parms.w2 * parms.t2 * parms.L2;         % [kg]
parms.I2      = (1/12) * parms.m2 * parms.L2^2;                    % [kg*m^2]

% World parameters
parms.g       = 9.81;                  % [m/s^2]

% Create state space matrices
[parms] = create_state(parms);

%% Question 1: Initial states
% Set Force vector
parms.Fg         = [parms.m1*parms.g;0;0;parms.m2*parms.g;0;0];

% Both bars vertical up and zero speed
x0            = [0.5*pi 0.5*pi 0 0];
x_dp_1        = state_calc(x0,parms);

% Both bars horizontal to the right and zero speed
x0            = [0 0 0 0];
x_dp_2        = state_calc(x0,parms);

% Both bars horizontal and with an initial angular speed on both bars of 60
% rpm
x0            = [0 0 2*pi 2*pi];
x_dp_3        = state_calc(x0,parms);

% Calculate velocities
x1_p          = -(parms.L1/2)*sin(x0(1))*x0(3);
y1_p          =  (parms.L1/2)*cos(x0(1))*x0(3);
x2_p          = x1_p - (parms.L1/2)*sin(x0(1))*x0(3) - (parms.L2/2)*sin(x0(2))*x0(4);
y2_p          = y1_p + (parms.L1/2)*cos(x0(1))*x0(3) + (parms.L2/2)*cos(x0(2))*x0(4);

% Questions e-g Extra constraint
parms.booleans.ex_constr = 1;                   % Put on 1 if you want to enable the
extra constraint
[parms] = create_state(parms);                  % Calculate new state matrixes

% e - Both bars vertical up and zero speed
x0            = [0.5*pi 0.5*pi 0 0];
x_dp_e        = state_calc(x0,parms);

% f - Both bars vertical up and with an initial angular speed of omega = 60
% rpm on bar 1
x0           = [0.5*pi 0.5*pi -2*pi 0];
x_dp_f       = state_calc(x0,parms);
```

```
% f - Both bars vertical up and with an initial angular speed of omega = 60
% rpm on bar 1
parms.Fg     = [parms.m1*parms.g;0;-10*(parms.L1/2);parms.m2*parms.g;0;0];        % Add a
10 N force in the y direction in B
x0           = [0 pi 2*pi 0];
x_dp_g       = state_calc(x0,parms);

% Test rank of Cx matrix
Cx          = double(vpa(subs(parms.Cx,{'phi1','phi2','phid1','phid2' 'm1' 'm2' 'I1' 'I2'
'L1' 'L2'},[x0(1) x0(2) x0(3) x0(4) parms.m1 parms.m2 parms.I1 parms.I2 parms.L1
parms.L2])));
rank_cx     = rank(Cx);
null_cx     = null(Cx);

%% Functions
% -- Create_state space --
% This function calculates the state matrixes for our double pendulum
% simulation

function [parms] = create_state(parms)

% Create symbolic variables
syms x1 y1 phi1 x2 y2 phi2                        % States
syms xd1 yd1 phid1 xd2 yd2 phid2                  % State derivatives (dx/dt)
syms L1 L2 m1 m2 I1 I2 g                          % Parameters

% put the cm cogordinates and their time derivatives in a column vector x
x       = [x1; y1; phi1; x2; y2; phi2];
xd      = [xd1; yd1; phid1; xd2; yd2; phid2];

% Create constraints
ck_x1      = x1-(L1/2)*cos(phi1);                                   % X constraint on
first body
ck_y1      = y1-(L1/2)*sin(phi1);                                   % Y constraint on
second body
ck_x2      =  (x2-L2/2*cos(phi2))-(x1+L1/2*cos(phi1));              % X constraint on
second body
ck_y2      =  (y2-L2/2*sin(phi2))-(y1+L1/2*sin(phi1));              % Y constraint on
second body
if parms.booleans.ex_constr == 1
    ck_x2c     = x2+(L2/2)*cos(phi2);                               % Extra constraint on
the right end of bar 2 (vertical path following at orgin)
end

% Calculate the jacobian (This can be used for the constraint equations of
% motion
if parms.booleans.ex_constr == 1
    C          = [ck_x1;ck_y1;ck_x2;ck_y2;ck_x2c];
else
    C          = [ck_x1;ck_y1;ck_x2;ck_y2];
end
Cx          = jacobian(C,x);                                    % Calculate partial
derivative
Cx          = simplify(Cx);

% Calculate second derivative this is done with the jacobian and hessian
% (Chain rule)
Cd          = Cx*xd;
Cdp         = jacobian(Cd,x)*xd;
Cdp         = simplify(Cdp);

parms.Cx = Cx;
parms.Cdp = Cdp;

end

% -- State Calc --
% This function calculates the second derivative of the double pendulum
% using the initial states.
function [x_dp] = state_calc(x0,parms)

% Now build the system
```

```
M          = diag([parms.m1 parms.m1 parms.I1 parms.m2 parms.m2 parms.I2]);
Fg         = parms.Fg;
A          = [M,parms.Cx';parms.Cx,zeros(size(parms.Cx,1),size(parms.Cx',2))];
b          = [parms.Fg;-parms.Cdp];

% Substitude initial states in derived matrices
A         = double(vpa(subs(A,{'phi1','phi2','phid1','phid2' 'm1' 'm2' 'I1' 'I2' 'L1'
'L2'},[x0(1) x0(2) x0(3) x0(4) parms.m1 parms.m2 parms.I1 parms.I2 parms.L1
parms.L2])));
b         = double(vpa(subs(b,{'phi1','phi2','phid1','phid2' 'm1' 'm2' 'L1' 'L2'
'g'},[x0(1) x0(2) x0(3) x0(4) parms.m1 parms.m2 parms.L1 parms.L2 parms.g])));

% Calculate second derivative
x_dp = A\b;

end
```