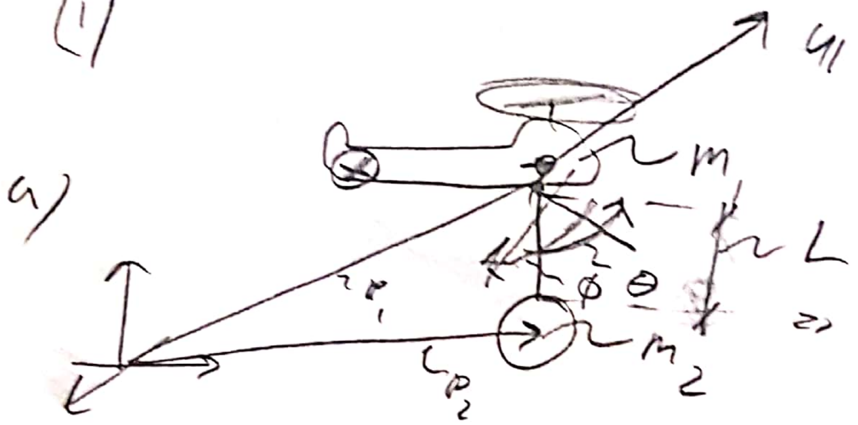
TTK 4130 MODELING AND SIMULATION / ASSIGNMENT 5

ODIN A. SEVERINSEN / MTTK

1

a)



$$q = \begin{bmatrix} p_1 \\ \theta \\ \phi \end{bmatrix}$$

$\Rightarrow$

$$\dot{q} = v$$

$$M(q)\dot{v} = k(q, \dot{q}, u)$$

The position of m_2 is given by

$$p_2 = p_1 + \begin{bmatrix} L\sin(\theta)\cos(\phi) \\ L\sin(\theta)\sin(\phi) \\ L\cos\theta \end{bmatrix}$$

The generalized forces are $Q = \underbrace{\begin{bmatrix} u \\ 0 \\ 0 \end{bmatrix}}_{\text{External forces}}$

The kinetic and potential energies are:

$$T = \frac{1}{2} m_1 \dot{p}_1^T \dot{p}_1 + \frac{1}{2} m_2 \dot{p}_2^T \dot{p}_2$$

$$V = m_1 g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T p_1 + m_2 g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T p_2$$

$\Rightarrow$ $\underline{\mathcal{L} = T - V}$

b) The kinetic energy can now be written as

$$T = \frac{1}{2} m_1 \dot{p}_1^T \dot{p}_2 + \frac{1}{2} m_2 \dot{p}_2^T \dot{p}_2$$

$$= \frac{1}{2} \dot{q}^T \underbrace{\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}}_{W} \dot{q}$$

Potential energy: $m_1 g [0 \; 0 \; 1] p_1 + m_2 g [0 \; 0 \; 1] p_2$

$$= \underbrace{[0 \;\; 0 \;\; m_1 g \;\; 0 \;\; 0 \;\; m_2 g]}_{G^T} q$$

$\Rightarrow$

$$\mathcal{L} = T - V = \frac{1}{2} \dot{q}^T W \dot{q} - G^T q - z \cdot c \sim \frac{1}{2}(e^T e - L^2)$$

$$e = p_1 - p_2 = \underbrace{[I_3 \;\; -I_3]}_{H} q_3$$

$$\Rightarrow \frac{1}{2}(q^T H^T H q - L^2)$$

$$\Rightarrow \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} - \underbrace{z \nabla_q c(q)}_{z \cdot H^T H q}$$

$$= \frac{d}{dt}(W \dot{q}) - G$$

$$= W \ddot{q} - G = Q \Rightarrow W \ddot{q} = Q + G$$

$$\begin{bmatrix} I_3 & -I_3 \\ -I_3 & I_3 \end{bmatrix}$$

$$\Rightarrow M = W, \quad b = Q + G + z$$

$$\mathbb{M}\ddot{q} = \mathbb{b}$$

$$\mathbb{M} = \mathbb{W} = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}$$

$$\mathbb{b} = Q - \cancel{\frac{\partial}{\partial q}\left(\mathbb{W}(q)\dot{q}\right)\dot{q}}^{\,0} + \cancel{\nabla_q T}^{\,0} - \nabla_q V - z\nabla_q c$$

$$= Q - \nabla_q V - z\nabla_q c$$

$\Rightarrow$ $\mathbb{M}_{\text{impl}}\,\ddot{x} = \mathbb{C}$, $\Rightarrow$ $\mathbb{M}_{\text{impl}} = \begin{bmatrix} \mathbb{W}(q) & \nabla_q c \\ \nabla_q \mathbb{C}^T & 0 \end{bmatrix}$ $\mathbb{}^{\text{S}}_{q}$,

$x = \begin{bmatrix} \ddot{q} \\ z \end{bmatrix}$, $\mathbb{C} = \begin{bmatrix} Q - \nabla_q \mathcal{V} \\ -\frac{\partial}{\partial q}\left(\frac{\partial c}{\partial q}\dot{q}\right)\dot{q} \end{bmatrix}$.

$\boxed{2}$ b)

The $\mathbb{M}$ and $\mathbb{b}$ in 1a), are nasty, but $\mathbb{M}$ and $\mathbb{b}$ in 1b) are much cleaner.

b) $\begin{bmatrix} \mathbb{W}(q) & \nabla_q \mathbb{C} \\ \nabla_q \mathbb{C}^T & 0 \end{bmatrix}$ is not too bad, but

$\begin{bmatrix} W & \nabla_q c \\ \nabla \mathbb{C}_q^T & 0 \end{bmatrix}^{-1}$ is really nasty-looking

2

a)

$$\mathfrak{A}(q) = \nabla_q C$$

$$\mathbb{C} = \begin{bmatrix} Q - \nabla_q \mathcal{L} \\ -\frac{\partial}{\partial q}\left(\frac{\partial c}{\partial q}\dot{q}\right)\dot{q} \end{bmatrix}$$

3

a) Only using $\alpha_1, \alpha_2, \alpha_3$ would result in two different points for nacelle. Additionally, it is computationally demanding as it includes three pivots and three double rods.

b) $V = mg[0 \quad 0 \quad 1]P$

$T = \frac{1}{2} J \|\dot{\alpha}_i\|^2 + \frac{1}{2} m \|\dot{P}\|^2$

$\mathcal{L} = T - V - z \cdot C(q)$, $\Rightarrow \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = Q$, $Q = \begin{bmatrix} u \\ 0 \end{bmatrix}$

$C(q) = 0 \Rightarrow$

$$\|P - P_1\|^2 - l^2 = 0$$
$$\|P - P_2\|^2 - l^2 = 0$$
$$\|P - P_3\|^2 - l^2 = 0$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_f} = W\ddot{q}_f, \quad T = \frac{1}{2}\dot{q}_f^{T} W \dot{q}_f \Rightarrow W = \begin{bmatrix} J & & & & & \\ & J & & & 0 & \\ & & J & & & \\ & 0 & & m & & \\ & & & & m & \\ & & & & & m \end{bmatrix}$$

$$\frac{\partial \mathcal{L}}{\partial q_f} = \begin{bmatrix} 0_{5\times1} \\ -mg \end{bmatrix}$$

```
clear all
clc

% Parameters
syms m1 m2 L g real
% Force
u = sym('u',[3,1]);

% Position point mass 1
pm1 = sym('p1',[3,1]);
dpm1 = sym('dp1',[3,1]);
ddpm1 = sym('d2p1',[3,1]);
% Angles for point mass 2
a  = sym('a',[2,1]);
da = sym('da',[2,1]);
dda = sym('d2a',[2,1]);
% Generalized coordinates
q  = [pm1;a];
dq = [dpm1;da];
ddq = [ddpm1;dda];

% Position of point mass 2
pm2  = pm1 + L*[sin(a(1))*cos(a(2)); sin(a(1))*sin(a(2)); cos(a(1))];
% Velocity of point mass 2
dpm2 = jacobian(pm2,q)*dq;
% Generalized forces
Q = [u; 0; 0];
% Kinetic energy
T = 0.5*m1*(dpm1'*dpm1) + 0.5*m2*(dpm2'*dpm2);
T = simplify(T);
% Potential energy
V = m1*g*[0 0 1]*pm1 + m2*g*[0 0 1]*pm2;
% Lagrangian
Lag = T - V;

% Derivatives of the Lagrangian
Lag_q = simplify(jacobian(Lag,q)).';
Lag_qdq = simplify(jacobian(Lag_q.',dq));
Lag_dq = simplify(jacobian(Lag,dq)).';
Lag_dqdq = simplify(jacobian(Lag_dq.',dq)); % W

% Matrices for problem 1
M = Lag_dqdq;
b = Q + simplify(Lag_q - Lag_qdq*dq);
```

Published with MATLAB® R2019a

```
clear all
clc

% Parameters
syms m1 m2 L g real
% Force
u = sym('u',[3,1]);

% Positions of point masses
pm1 = sym('pm1',[3,1]);
pm2 = sym('pm2',[3,1]);
dpm1 = sym('dpm1',[3,1]);
dpm2 = sym('dpm2',[3,1]);
ddpm1 = sym('d2pm1',[3,1]);
ddpm2 = sym('d2pm2',[3,1]);
% Generalized coordinates
q  = [pm1;pm2];
dq = [dpm1;dpm2];
ddq = [ddpm1;ddpm2];
% Algebraic variable
z = sym('z');

% Generalized forces
Q = [u; 0; 0; 0];
% Kinetic energy (function of q and dq)
W = [m1*eye(3), zeros(3); zeros(3), m2*eye(3)];
T = 0.5*dq'*W*dq;
% Potential energy
G = [0 0 m1*g 0 0 m2*g]';
V = G'*q;
% Lagrangian (function of q and dq)
Lag = T - V;
% Constraint
dpm  = pm1 - pm2; % difference of positions
C = 0.5 * (dpm'*dpm - L^2);

% Derivatives of constrained Lagrangian
Lag_q = simplify(jacobian(Lag,q)).';
Lag_qdq = simplify(jacobian(Lag_q.',dq));
Lag_dq = simplify(jacobian(Lag,dq)).';
Lag_dqdq = simplify(jacobian(Lag_dq.',dq)); % W
C_q = simplify(jacobian(C,q)).';

% Matrices for problem 1
M = Lag_dqdq;
b = Q - Lag_q - z*C_q;

% Matrices for problem 2
Mimplicit = [Lag_dqdq, C_q; C_q', 0];
c = [Q - Lag_q; -dq'*C_q];
% Mexplicit = simplify(inv(Mimplicit));
rhs = simplify(Mexplicit*c);
```

```
Undefined function or variable 'Mexplicit'.

Error in HoveringMassConstraintTemplate (line 52)
rhs = simplify(Mexplicit*c);
```

Published with MATLAB® R2019a