MECA482 CSU Chico - Furuta Pendulum Model - Aaron Taylor Angel Sanchez Ingrid Tisell Michele Fragasso Joe Karam

This is the model from the Student Workbook by Quanser developed by Jacob Apkarian, Paul Karam and Michel Lévis. This script starts from the definition of the non-linear model. It solves the ODE numerically to quickly verify its correctness. It reduced the system of 2 seocnd order equations to 4 first equation in order to linearize it about the unstable equilibrium position to get the matrixes A,B and C. All the results are exported in Matlab using the library CodeGeneration.

```
> restart;
> with(plots):
  with(CodeGeneration):
  with(VectorCalculus):
  with(LinearAlgebra):
> merge := proc(x,y)
    [op(x), op(y)]
  end proc;
```

merge := **proc**(x, y) [op(x), op(y)] **end proc** **(1.1)**

```
> linearize := proc(eqs, lin_point)
  local var, vardot, f, var_sub_eqs,var_sub,lin_point_sub,f_sub,
  J,deltax,f_subx0,f_lin_sub,var_sub_eqs_inv,f_lin,
  eqs_space_state;

      var := [seq(lhs(lin_point[i]),i=1..numelems(lin_point))];
      vardot := diff(var,t);
      f:= map(rhs,map(op,solve(eqs,vardot)));
      var_sub_eqs := [seq(var[i]=cat(x,__,i), i = 1..numelems
   (var))];
      var_sub := [seq(rhs(var_sub_eqs[i]), i=1..numelems
   (var_sub_eqs))];
      lin_point_sub := subs(var_sub_eqs, lin_point);
      f_sub := subs(var_sub_eqs, f);
      J := Matrix(evalf(subs(lin_point_sub,Jacobian(f_sub,
  var_sub))));
      deltax:= Transpose(Matrix([seq(var_sub[i] - rhs
   (lin_point_sub[i]), i=1..numelems(var_sub))]));
      f_subx0 := Transpose(Matrix(evalf(subs(lin_point_sub,
   (f_sub)))));
      f_lin_sub := simplify(f_subx0 + J.deltax);
      var_sub_eqs_inv := [seq(rhs(var_sub_eqs[i]) = lhs
   (var_sub_eqs[i]), i =1..numelems(var_sub))];
      f_lin := subs(var_sub_eqs_inv,f_lin_sub);
      eqs_space_state := [seq(vardot[i] = f_lin[i,1],i=1..
  numelems(var))];

  end proc;
```

linearize := **proc**(eqs, lin_point) **(1.2)**

local var, vardot, f, var_sub_eqs, var_sub, lin_point_sub, f_sub, J, deltax, f_subx0, f_lin_sub, var_sub_eqs_inv, f_lin, eqs_space_state;

var := [seq(lhs(lin_point[i]), i = 1 ..numelems(lin_point))];

vardot := VectorCalculus:-diff(var, t);

```
f := map(rhs, map(op, solve(eqs, vardot)));
var_sub_eqs := [seq(var[i] = cat(x, __, i), i = 1 ..numelems(var))];
var_sub := [seq(rhs(var_sub_eqs[i]), i = 1 ..numelems(var_sub_eqs))];
lin_point_sub := subs(var_sub_eqs, lin_point);
f_sub := subs(var_sub_eqs, f);
J := Matrix(evalf(subs(lin_point_sub, VectorCalculus:-Jacobian(f_sub, var_sub))));
deltax := LinearAlgebra:-Transpose(Matrix([seq(VectorCalculus:-`+`(var_sub[i],
VectorCalculus:-`-`(rhs(lin_point_sub[i]))), i = 1 ..numelems(var_sub))]));
f_subx0 := LinearAlgebra:-Transpose(Matrix(evalf(subs(lin_point_sub, f_sub))));
f_lin_sub := simplify(VectorCalculus:-`+`(f_subx0, VectorCalculus:-`.`(J, deltax)));
var_sub_eqs_inv := [seq(rhs(var_sub_eqs[i]) = lhs(var_sub_eqs[i]), i = 1
..numelems(var_sub))];
f_lin := subs(var_sub_eqs_inv, f_lin_sub);
eqs_space_state := [seq(vardot[i] = f_lin[i, 1], i = 1 ..numelems(var))]
end proc
```

Equations of Motion with and without parameters

```
> eq1 := (m__p*L__r^2+1/4*m__p*L__p^2-1/4*m__p*L__p^2*cos(alpha(t))^2+J__r)*diff(theta(t),t,t) -(1/2*m__p*L__p*L__r*cos(alpha(t)))*diff(alpha(t),t,t) +(1/2*m__p*L__p^2*sin(alpha(t)*cos(alpha(t))))*diff(alpha(t),t)*diff(theta(t),t)+(1/2*m__p*L__p*L__r*sin(alpha(t)))*diff(alpha(t),t)^2 = eta__g*k__g*eta__m*k__t*((V__m - k__g*k__m*diff(theta(t), t))/r__m) - B__r*diff(theta(t),t);
```

$$eq1 := \left(m_p L_r^2 + \frac{m_p L_p^2}{4} - \frac{m_p L_p^2 \cos(\alpha(t))^2}{4} + J_r\right)\left(\frac{d^2}{dt^2}\,\theta(t)\right) - \frac{m_p L_p L_r \cos(\alpha(t))\left(\frac{d^2}{dt^2}\,\alpha(t)\right)}{2} + \frac{m_p L_p^2 \sin(\alpha(t)\cos(\alpha(t)))\left(\frac{d}{dt}\,\alpha(t)\right)\left(\frac{d}{dt}\,\theta(t)\right)}{2} + \frac{m_p L_p L_r \sin(\alpha(t))\left(\frac{d}{dt}\,\alpha(t)\right)^2}{2} = \frac{\eta_g k_g \eta_m k_t \left(V_m - k_g k_m \left(\frac{d}{dt}\,\theta(t)\right)\right)}{r_m} - B_r\left(\frac{d}{dt}\,\theta(t)\right) \quad \textbf{(1.1.1)}$$

```
> eq2 := -1/2*m__p*L__p*L__r*cos(alpha(t))*diff(theta(t),t,t)+(J__p+1/4*m__p*L__p^2)*diff(alpha(t),t,t)-1/4*m__p*L__p^2*cos(alpha(t)) * sin(alpha(t)) * diff(theta(t),t)^2 - 1/2*m__p*L__p*g*sin(alpha(t)) = -B__p*diff(alpha(t),t)
```

(1.1.2)

$$eq2 := -\frac{m_p L_p L_r \cos(\alpha(t)) \left(\frac{d^2}{dt^2}\theta(t)\right)}{2} + \left(J_p + \frac{m_p L_p^2}{4}\right)\left(\frac{d^2}{dt^2}\alpha(t)\right) - \frac{m_p L_p^2 \cos(\alpha(t))\sin(\alpha(t))\left(\frac{d}{dt}\theta(t)\right)^2}{4} - \frac{m_p L_p g \sin(\alpha(t))}{2} = -B_p\left(\frac{d}{dt}\alpha(t)\right) \quad \textbf{(1.1.2)}$$

```
> data_electrical := [eta__g = 0.85, eta__m = 0.87, k__g = 70,
  k__m = 0.0076 , k__t=0.0076, r__m=2.6, V__m = 0];
```

$$data\_electrical := [\eta_g = 0.85, \eta_m = 0.87, k_g = 70, k_m = 0.0076, k_t = 0.0076, r_m = 2.6, V_m = 0] \quad \textbf{(1.1.3)}$$

```
> data_mechanical := [J__p = 0.0023, m__p = 0.125, L__r =
  0.215, m__r = 0, L__p = 0.335, J__r = 0.0023, B__p = 0.000,
  tau__1 = 0, g=9.81, tau__2 = 0, B__r = 0.000];
```

$$data\_mechanical := [J_p = 0.0023, m_p = 0.125, L_r = 0.215, m_r = 0, L_p = 0.335, J_r = 0.0023, B_p = 0., \tau_1 = 0, g = 9.81, \tau_2 = 0, B_r = 0.] \quad \textbf{(1.1.4)}$$

```
> Matlab(data_electrical, resultname="data_e");
eta__g = 0.85e0;
eta__m = 0.87e0;
k__g = 70;
k__m = 0.76e-2;
k__t = 0.76e-2;
r__m = 0.26e1;
V__m = 0;
> Matlab(data_mechanical, resultname="data_m");
J__p = 0.23e-2;
m__p = 0.125e0;
L__r = 0.215e0;
m__r = 0;
L__p = 0.335e0;
J__r = 0.23e-2;
B__p = 0.0e0;
tau__1 = 0;
g = 0.981e1;
tau__2 = 0;
B__r = 0.0e0;
> data := merge(data_mechanical, data_electrical);
```

$$data := [J_p = 0.0023, m_p = 0.125, L_r = 0.215, m_r = 0, L_p = 0.335, J_r = 0.0023, B_p = 0., \tau_1 = 0, g = 9.81, \tau_2 = 0, B_r = 0., \eta_g = 0.85, \eta_m = 0.87, k_g = 70, k_m = 0.0076, k_t = 0.0076, r_m = 2.6, V_m = 0] \quad \textbf{(1.1.5)}$$

```
> subs(data, eq1);
  subs(data, eq2);
```

$$\left(0.01158515625 - 0.003507031250 \cos(\alpha(t))^2\right)\left(\frac{d^2}{dt^2}\theta(t)\right)$$

$$-0.004501562500\cos(\alpha(t))\left(\frac{d^2}{dt^2}\alpha(t)\right)$$
$$+0.007014062500\sin(\alpha(t)\cos(\alpha(t)))\left(\frac{d}{dt}\alpha(t)\right)\left(\frac{d}{dt}\theta(t)\right)$$
$$+0.004501562500\sin(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right)^2 = -0.08049855692\left(\frac{d}{dt}\theta(t)\right)$$
$$-0.004501562500\cos(\alpha(t))\left(\frac{d^2}{dt^2}\theta(t)\right)+0.005807031250\left(\frac{d^2}{dt^2}\alpha(t)\right) \tag{1.1.6}$$
$$-0.003507031250\cos(\alpha(t))\sin(\alpha(t))\left(\frac{d}{dt}\theta(t)\right)^2-0.2053968750\sin(\alpha(t))$$
$$=-0.$$

```
> eqs := [eq1 ,eq2];
```

$$eqs := \left[\left(m_p L_r^2+\frac{m_p L_p^2}{4}-\frac{m_p L_p^2\cos(\alpha(t))^2}{4}+J_r\right)\left(\frac{d^2}{dt^2}\theta(t)\right) \tag{1.1.7}$$
$$-\frac{m_p L_p L_r\cos(\alpha(t))\left(\frac{d^2}{dt^2}\alpha(t)\right)}{2}$$
$$+\frac{m_p L_p^2\sin(\alpha(t)\cos(\alpha(t)))\left(\frac{d}{dt}\alpha(t)\right)\left(\frac{d}{dt}\theta(t)\right)}{2}$$
$$+\frac{m_p L_p L_r\sin(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right)^2}{2}=\frac{\eta_g k_g \eta_m k_t\left(V_m-k_g k_m\left(\frac{d}{dt}\theta(t)\right)\right)}{r_m}$$
$$-B_r\left(\frac{d}{dt}\theta(t)\right), -\frac{m_p L_p L_r\cos(\alpha(t))\left(\frac{d^2}{dt^2}\theta(t)\right)}{2}+\left(J_p+\frac{m_p L_p^2}{4}\right)\left(\frac{d^2}{dt^2}\right.$$
$$\left.\alpha(t)\right)-\frac{m_p L_p^2\cos(\alpha(t))\sin(\alpha(t))\left(\frac{d}{dt}\theta(t)\right)^2}{4}-\frac{m_p L_p g\sin(\alpha(t))}{2}=$$
$$\left.-B_p\left(\frac{d}{dt}\alpha(t)\right)\right]$$

Simulation

Initial condition --> Vertical Arm at Pi/4

```
> ics1 := [diff(alpha(t), t) = 0, diff(theta(t), t) = 0,
  alpha(t) = Pi/4, theta(t) = 0]
```

$$ics1 := \left[\frac{d}{dt}\,\alpha(t) = 0,\ \frac{d}{dt}\,\theta(t) = 0,\ \alpha(t) = \frac{\pi}{4},\ \theta(t) = 0\right] \tag{1.2.1.1}$$

```
> ics1 := subs(t=0,convert(ics1, D))
```

$$ics1 := \left[D(\alpha)(0) = 0,\ D(\theta)(0) = 0,\ \alpha(0) = \frac{\pi}{4},\ \theta(0) = 0\right] \tag{1.2.1.2}$$

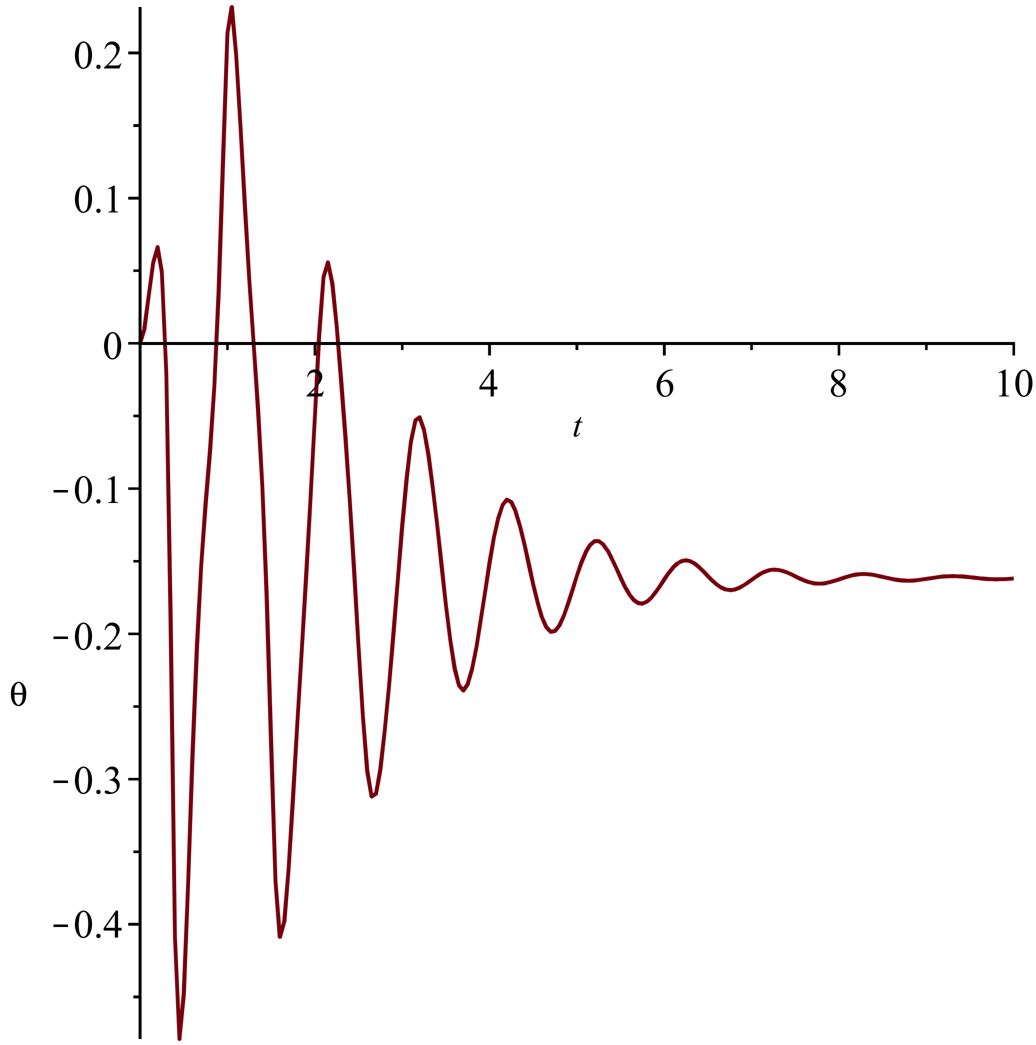
```
> ODEs1 := merge((subs(data, eqs)), ics1);
```

$$\begin{aligned} ODEs1 := \Bigg[& \left(0.01158515625 - 0.003507031250\cos(\alpha(t))^2\right)\left(\frac{d^2}{dt^2}\,\theta(t)\right) \\ & - 0.004501562500\cos(\alpha(t))\left(\frac{d^2}{dt^2}\,\alpha(t)\right) \\ & + 0.007014062500\sin(\alpha(t)\cos(\alpha(t)))\left(\frac{d}{dt}\,\alpha(t)\right)\left(\frac{d}{dt}\,\theta(t)\right) \\ & + 0.004501562500\sin(\alpha(t))\left(\frac{d}{dt}\,\alpha(t)\right)^2 = -0.08049855692\left(\frac{d}{dt}\,\theta(t)\right), \\ & -0.004501562500\cos(\alpha(t))\left(\frac{d^2}{dt^2}\,\theta(t)\right) + 0.005807031250\left(\frac{d^2}{dt^2}\,\alpha(t)\right) \\ & - 0.003507031250\cos(\alpha(t))\sin(\alpha(t))\left(\frac{d}{dt}\,\theta(t)\right)^2 \\ & - 0.2053968750\sin(\alpha(t)) = -0.,\ D(\alpha)(0) = 0,\ D(\theta)(0) = 0,\ \alpha(0) = \frac{\pi}{4}, \\ & \theta(0) = 0\Bigg] \end{aligned} \tag{1.2.1.3}$$

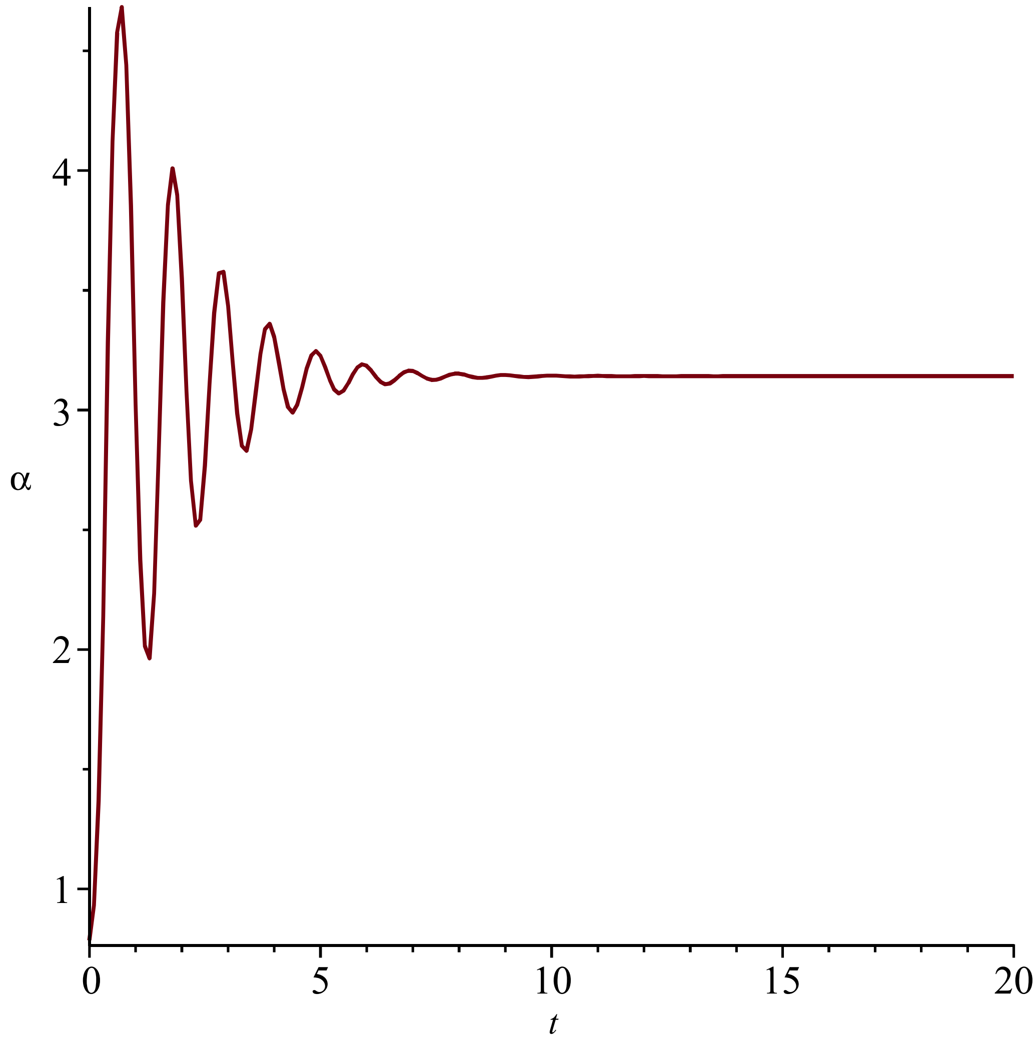
```
> ode_sol1 := dsolve(ODEs1, numeric);
```

$$ode\_sol1 := \mathbf{proc}(x\_rkf45)\ \ldots\ \mathbf{end\ proc} \tag{1.2.1.4}$$

```
> odeplot(ode_sol1, [t, theta(t)], t=0..10);
  odeplot(ode_sol1, [t, alpha(t)], t=0..20);
```



0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
θ
2
4
6
8
10
t



Reducing to first order

```
> add_diff := [diff(theta(t), t) = thetadot(t), diff(alpha(t), t) = alphadot(t)];
```

$$add\_diff := \left[\frac{d}{dt}\, \theta(t) = thetadot(t),\ \frac{d}{dt}\, \alpha(t) = alphadot(t) \right] \quad \textbf{(1.3.1)}$$

```
> eqs_first_order := merge(subs(add_diff, eqs), add_diff);
```

$$eqs\_first\_order := \left[\left(m_p L_r^2 + \frac{m_p L_p^2}{4} - \frac{m_p L_p^2 \cos(\alpha(t))^2}{4} + J_r \right) \left(\frac{d}{dt}\, thetadot(t) \right) \right. \quad \textbf{(1.3.2)}$$

$$- \frac{m_p L_p L_r \cos(\alpha(t)) \left(\frac{d}{dt}\, alphadot(t) \right)}{2}$$

$$+ \frac{m_p L_p^2 \sin(\alpha(t) \cos(\alpha(t)))\, alphadot(t)\, thetadot(t)}{2}$$

$$+\frac{m_p L_p L_r \sin(\alpha(t))\, alphadot(t)^2}{2} = \frac{\eta_g k_g \eta_m k_t \left(V_m - k_g k_m\, thetadot(t)\right)}{r_m}$$

$$- B_r\, thetadot(t),\ -\frac{m_p L_p L_r \cos(\alpha(t)) \left(\frac{d}{dt}\, thetadot(t)\right)}{2} + \left(J_p + \frac{m_p L_p^{\,2}}{4}\right) \left(\frac{d}{dt}\right.$$

$$\left. alphadot(t)\right) - \frac{m_p L_p^{\,2} \cos(\alpha(t)) \sin(\alpha(t))\, thetadot(t)^2}{4} - \frac{m_p L_p g \sin(\alpha(t))}{2} =$$

$$-B_p\, alphadot(t),\ \frac{d}{dt}\,\theta(t) = thetadot(t),\ \frac{d}{dt}\,\alpha(t) = alphadot(t)\Big]$$

```
> var := [theta(t), alpha(t), thetadot(t), alphadot(t)];
```

$$var := [\theta(t), \alpha(t), thetadot(t), alphadot(t)] \quad \textbf{(1.3.3)}$$

```
> vardot := diff(var,t);
```

$$vardot := \left[\frac{d}{dt}\,\theta(t), \frac{d}{dt}\,\alpha(t), \frac{d}{dt}\, thetadot(t), \frac{d}{dt}\, alphadot(t)\right] \quad \textbf{(1.3.4)}$$

```
> var_sub := [theta(t) = theta, alpha(t) = alpha, thetadot(t) =
  thetadot, alphadot(t) = alphadot]
```

$$var\_sub := [\theta(t) = \theta, \alpha(t) = \alpha, thetadot(t) = thetadot, alphadot(t) = alphadot] \quad \textbf{(1.3.5)}$$

```
> solve(eqs_first_order, vardot)[1]:
  map(rhs, %):
  F := simplify(subs(var_sub, %));
```

$$F := \Bigg[thetadot, alphadot, \Big(2\, m_p r_m\, alphadot\, thetadot\, L_p^{\,2} \left(m_p L_p^{\,2} + 4 J_p\right) \sin(\alpha \cos(\alpha)) \quad \textbf{(1.3.6)}$$

$$- 2 L_p^{\,3} L_r \cos(\alpha)^2 \sin(\alpha)\, thetadot^2 m_p^{\,2} r_m - 4\, m_p r_m L_p L_r \big(g\, m_p \sin(\alpha)\, L_p$$

$$- 2\, alphadot\, B_p\big) \cos(\alpha) + 4 \left(m_p L_p^{\,2} + 4 J_p\right) \Bigg(\frac{m_p r_m\, alphadot^2 \sin(\alpha)\, L_p L_r}{2}$$

$$+ r_m\, thetadot\, B_r + k_t \eta_g \eta_m k_g \left(k_g k_m\, thetadot - V_m\right)\Bigg)\Big) \Big/ \Big(r_m \Big(L_p^{\,2} m_p \Big(\Big(L_p^{\,2}$$

$$+4 L_r^2\right) m_p + 4 J_p\right) \cos(\alpha)^2 - \left(m_p L_p^2 + 4 J_p\right) \left(\left(L_p^2 + 4 L_r^2\right) m_p + 4 J_r\right)\right)\right),$$

$$\left(4 L_p^3 L_r \sin(\alpha \cos(\alpha))\ alphadot \cos(\alpha)\ thetadot\ m_p^2 r_m\right.$$

$$+ L_p^4 \cos(\alpha)^3 \sin(\alpha)\ thetadot^2 m_p^2 r_m + 2 m_p r_m L_p^2 \left(g\, m_p \sin(\alpha) L_p\right.$$

$$\left.- 2\, alphadot\, B_p\right) \cos(\alpha)^2$$

$$+ 8 L_p \left(\frac{1}{2} \left(L_p r_m \left(\left(\left(-\frac{thetadot^2 L_p^2}{4} + L_r^2 \left(alphadot^2 - thetadot^2\right)\right) m_p\right.\right.\right.\right.$$

$$\left.\left.\left.- thetadot^2 J_r\right) \sin(\alpha)\right) + L_r \left(r_m\, thetadot\, B_r + k_t \eta_g \eta_m k_g \left(k_g k_m\, thetadot - V_m\right)\right)\right)$$

$$\left. m_p \cos(\alpha) - 2 r_m \left(\left(L_p^2 + 4 L_r^2\right) m_p + 4 J_r\right) \left(g\, m_p \sin(\alpha) L_p - 2\, alphadot\, B_p\right)\right)$$

$$\Big/ \left(r_m \left(L_p^2 m_p \left(\left(L_p^2 + 4 L_r^2\right) m_p + 4 J_p\right) \cos(\alpha)^2 - \left(m_p L_p^2 + 4 J_p\right) \left(\left(L_p^2\right.\right.\right.\right.$$

$$\left.\left.\left.\left.+ 4 L_r^2\right) m_p + 4 J_r\right)\right)\right)\right]$$

```
> Matlab(F[1], resultname="dtheta_dt");
dtheta_dt = thetadot;
> Matlab(F[2], resultname="dalpha_dt");
dalpha_dt = alphadot;
> Matlab(F[3], resultname="d2theta_dt2");
d2theta_dt2 = (0.2e1 * m__p * r__m * alphadot * thetadot *
(L__p ^ 2) * (m__p * L__p ^ 2 + 4 * J__p) * sin(alpha * cos
(alpha)) - 0.2e1 * (L__p ^ 3) * L__r * cos(alpha) ^ 2 * sin
(alpha) * thetadot ^ 2 * (m__p ^ 2) * r__m - 0.4e1 * m__p *
r__m * L__p * L__r * (g * m__p * sin(alpha) * L__p - 0.2e1 *
alphadot * B__p) * cos(alpha) + 0.4e1 * (m__p * L__p ^ 2 + 4
* J__p) * (m__p * r__m * alphadot ^ 2 * sin(alpha) * L__p *
L__r / 0.2e1 + r__m * thetadot * B__r + k__t * eta__g *
eta__m * k__g * (k__g * k__m * thetadot - V__m))) / r__m / (
```

```
(L__p ^ 2) * m__p * (((L__p ^ 2) + 0.4e1 * L__r ^ 2) * m__p
+ (4 * J__p)) * cos(alpha) ^ 2 - (m__p * L__p ^ 2 + 4 *
J__p) * (((L__p ^ 2) + 0.4e1 * L__r ^ 2) * m__p + (4 * J__r)
));
```

```
> Matlab(F[4], resultname="d2alpha_dt2");
```

```
d2alpha_dt2 = (0.4e1 * L__p ^ 3 * L__r * sin(alpha * cos
(alpha)) * alphadot * cos(alpha) * thetadot * m__p ^ 2 *
r__m + L__p ^ 4 * cos(alpha) ^ 3 * sin(alpha) * thetadot ^ 2
* m__p ^ 2 * r__m + 0.2e1 * m__p * r__m * L__p ^ 2 * (g *
m__p * sin(alpha) * L__p - 0.2e1 * alphadot * B__p) * cos
(alpha) ^ 2 + 0.8e1 * L__p * (L__p * r__m * ((-thetadot ^ 2
* L__p ^ 2 / 0.4e1 + L__r ^ 2 * (alphadot ^ 2 - thetadot ^
2)) * m__p - thetadot ^ 2 * J__r) * sin(alpha) / 0.2e1 +
L__r * (r__m * thetadot * B__r + k__t * eta__g * eta__m *
k__g * (k__g * k__m * thetadot - V__m))) * m__p * cos(alpha)
- 0.2e1 * r__m * ((L__p ^ 2 + 0.4e1 * L__r ^ 2) * m__p +
0.4e1 * J__r) * (g * m__p * sin(alpha) * L__p - 0.2e1 *
alphadot * B__p)) / r__m / (L__p ^ 2 * m__p * ((L__p ^ 2 +
0.4e1 * L__r ^ 2) * m__p + (4 * J__p)) * cos(alpha) ^ 2 -
(m__p * L__p ^ 2 + (4 * J__p)) * ((L__p ^ 2 + 0.4e1 * L__r ^
2) * m__p + 0.4e1 * J__r));
```

Reducing to state space model and exporting to Matlab

```
> eqs_first_order
```

$$\left[\left(m_p L_r^2 + \frac{m_p L_p^2}{4} - \frac{m_p L_p^2 \cos(\alpha(t))^2}{4} + J_r\right)\left(\frac{d}{dt}\, thetadot(t)\right) - \frac{m_p L_p L_r \cos(\alpha(t))\left(\frac{d}{dt}\, alphadot(t)\right)}{2} + \frac{m_p L_p^2 \sin(\alpha(t)\cos(\alpha(t)))\, alphadot(t)\, thetadot(t)}{2} + \frac{m_p L_p L_r \sin(\alpha(t))\, alphadot(t)^2}{2} = \frac{\eta_g k_g \eta_m k_t \left(V_m - k_g k_m\, thetadot(t)\right)}{r_m} - B_r\, thetadot(t), -\frac{m_p L_p L_r \cos(\alpha(t))\left(\frac{d}{dt}\, thetadot(t)\right)}{2} + \left(J_p + \frac{m_p L_p^2}{4}\right)\left(\frac{d}{dt}\, alphadot(t)\right) - \frac{m_p L_p^2 \cos(\alpha(t)) \sin(\alpha(t))\, thetadot(t)^2}{4} - \frac{m_p L_p g \sin(\alpha(t))}{2} = -B_p\, alphadot(t), \frac{d}{dt}\theta(t) = thetadot(t), \frac{d}{dt}\alpha(t) = alphadot(t)\right] \tag{2.1}$$

```
> lin_point:=[theta(t) =2.99,alpha(t)=0,thetadot(t)=0,alphadot(t)
=0];
```

(2.2)

$$lin\_point := [\theta(t) = 2.99, \alpha(t) = 0, thetadot(t) = 0, alphadot(t) = 0] \quad (2.2)$$

```
> eqs_space_state := linearize(eqs_first_order, lin_point);
```

$$eqs\_space\_state := \left[\frac{d}{dt}\theta(t) = 1.\, thetadot(t), \frac{d}{dt}\alpha(t) = 1.\, alphadot(t), \frac{d}{dt} thetadot(t) \right. \quad (2.3)$$

$$= \frac{1}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)} \Big(L_p^2 L_r g m_p^2 \alpha(t) r_m + \Big(\Big(-1.\, B_r\, thetadot(t)\, r_m$$

$$+ \left(-1.\, thetadot(t)\, k_g^2 k_m k_t + V_m k_g k_t\right) \eta_m \eta_g\Big) L_p^2 - 2.\, B_p L_p L_r\, alphadot(t)\, r_m\Big) m_p + \Big($$

$$-4.\, B_r\, thetadot(t)\, r_m + \left(-4.\, thetadot(t)\, k_g^2 k_m k_t + 4.\, V_m k_g k_t\right) \eta_m \eta_g\Big) J_p\Big), \frac{d}{dt}$$

$$alphadot(t) = \frac{1}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)} \Big(2.\, L_p g m_p^2 r_m \alpha(t) L_r^2 + \Big(\Big(\Big($$

$$-2.\, B_r\, thetadot(t)\, r_m + \left(-2.\, thetadot(t)\, k_g^2 k_m k_t + 2.\, V_m k_g k_t\right) \eta_m \eta_g\Big) L_r$$

$$\left. + 2.\, g r_m \alpha(t) J_r\Big) L_p - 4.\, B_p L_r^2\, alphadot(t)\, r_m\Big) m_p - 4.\, B_p J_r\, alphadot(t)\, r_m\Big) \right]$$

```
> eqs_space_state2 := solve(eqs_space_state, vardot);
  A, RES := GenerateMatrix(map(rhs,eqs_space_state2[1]),var):
```

$$eqs\_space\_state2 := \left[\left[\frac{d}{dt}\theta(t) = thetadot(t), \frac{d}{dt}\alpha(t) = alphadot(t), \frac{d}{dt} thetadot(t) = \right.\right. \quad (2.4)$$

$$-\frac{1}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)} \Big(1.\, \Big(L_p^2\, thetadot(t)\, \eta_g \eta_m k_g^2 k_m k_t m_p$$

$$+ 4.\, J_p\, thetadot(t)\, \eta_g \eta_m k_g^2 k_m k_t - 1.\, L_p^2 L_r g m_p^2 \alpha(t) r_m - 1.\, L_p^2 V_m \eta_g \eta_m k_g k_t m_p$$

$$+ 2.\, B_p L_p L_r\, alphadot(t)\, m_p r_m + B_r L_p^2\, thetadot(t)\, m_p r_m - 4.\, J_p V_m \eta_g \eta_m k_g k_t$$

$$+ 4.\, B_r J_p\, thetadot(t)\, r_m\Big)\Big), \frac{d}{dt} alphadot(t) =$$

$$-\frac{1}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)} \Big(2.\, \Big(L_p L_r\, thetadot(t)\, \eta_g \eta_m k_g^2 k_m k_t m_p$$

$$- 1.\, L_p g m_p^2 r_m \alpha(t) L_r^2 - 1.\, L_p L_r V_m \eta_g \eta_m k_g k_t m_p + 2.\, B_p L_r^2\, alphadot(t)\, m_p r_m$$

$$\left.\left. + B_r L_p L_r\, thetadot(t)\, m_p r_m - 1.\, J_r L_p \alpha(t) g m_p r_m + 2.\, B_p J_r\, alphadot(t)\, r_m\Big)\Big) \right]\right]$$

```
> B, RES := GenerateMatrix(map(rhs, eqs_space_state2[1]),[V__m]):
```

Space State Matrixes

```
> A := Matrix(A);
  A_data := subs(data, A);
```

$$A := \left[\left[0, 0, 1, 0\right],\right.$$

$$\left[0, 0, 0, 1\right],$$

$$\left[0, \frac{1.\, L_p^2 L_r g\, m_p^2}{4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r},\right.$$

$$-\frac{1.\left(L_p^2 \eta_g \eta_m k_g^2 k_m k_t m_p + 4.\, J_p \eta_g \eta_m k_g^2 k_m k_t + B_r L_p^2 m_p r_m + 4.\, B_r J_p r_m\right)}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)},$$

$$\left.-\frac{2.\, B_p L_p L_r m_p}{4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r}\right],$$

$$\left[0, -\frac{2.\left(-L_p L_r^2 g\, m_p^2 r_m - J_r L_p g\, m_p r_m\right)}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)},\right.$$

$$-\frac{2.\left(L_p L_r \eta_g \eta_m k_g^2 k_m k_t m_p + B_r L_p L_r m_p r_m\right)}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)},$$

$$\left.\left.-\frac{2.\left(2.\, B_p L_r^2 m_p r_m + 2.\, B_p J_r r_m\right)}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)}\right]\right]$$

$$A\_data := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 34.69983298 & -17.54334995 & -0. \\ 0 & 62.26939834 & -13.59945949 & -0. \end{bmatrix} \qquad \mathbf{(2.1.1)}$$

```
> B := Matrix(B);
  B_data := subs(data, Matrix(B));
```

$$B := \begin{bmatrix} 0 \\ 0 \\ -\dfrac{1.\left(-L_p^2 \eta_g \eta_m k_g k_t m_p - 4.\, J_p \eta_g \eta_m k_g k_t\right)}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)} \\ \dfrac{2.\, L_p L_r \eta_g \eta_m k_g k_t m_p}{r_m\left(4.\, J_p L_r^2 m_p + J_r L_p^2 m_p + 4.\, J_p J_r\right)} \end{bmatrix}$$

(2.1.2)

$$B\_data := \begin{bmatrix} 0 \\ 0 \\ 32.97622171 \\ 25.56289378 \end{bmatrix} \quad \textbf{(2.1.2)}$$

```
> C_output := Matrix([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]);
```

$$C\_output := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(2.1.3)}$$

```
> state := var;
```

$$state := [\theta(t), \alpha(t), thetadot(t), alphadot(t)] \quad \textbf{(2.1.4)}$$

```
> Eigenvalues(A_data);
```

$$\begin{bmatrix} 0.\,\mathrm{I} \\ -19.1026211185592 + 0.\,\mathrm{I} \\ 6.53211884198625 + 0.\,\mathrm{I} \\ -4.97284767342700 + 0.\,\mathrm{I} \end{bmatrix} \quad \textbf{(2.1.5)}$$

Converting to Matlab

```
> Matlab(A, resultname="A");
A = [0 0 1 0; 0 0 0 1; 0 0.1e1 / (0.4e1 * J__p * L__r ^ 2 *
m__p + J__r * L__p ^ 2 * m__p + 0.4e1 * J__p * J__r) * L__p
^ 2 * L__r * g * m__p ^ 2 -0.1e1 / r__m / (0.4e1 * J__p *
L__r ^ 2 * m__p + J__r * L__p ^ 2 * m__p + 0.4e1 * J__p *
J__r) * (L__p ^ 2 * eta__g * eta__m * k__g ^ 2 * k__m * k__t
* m__p + 0.4e1 * J__p * eta__g * eta__m * k__g ^ 2 * k__m *
k__t + B__r * L__p ^ 2 * m__p * r__m + 0.4e1 * B__r * J__p *
r__m) -0.2e1 / (0.4e1 * J__p * L__r ^ 2 * m__p + J__r * L__p
^ 2 * m__p + 0.4e1 * J__p * J__r) * B__p * L__p * L__r *
m__p; 0 -0.2e1 * (-0.1e1 * L__p * L__r ^ 2 * g * m__p ^ 2 *
r__m - 0.1e1 * J__r * L__p * g * m__p * r__m) / r__m /
(0.4e1 * J__p * L__r ^ 2 * m__p + J__r * L__p ^ 2 * m__p +
0.4e1 * J__p * J__r) -0.2e1 * (L__p * L__r * eta__g * eta__m
* k__g ^ 2 * k__m * k__t * m__p + B__r * L__p * L__r * m__p
* r__m) / r__m / (0.4e1 * J__p * L__r ^ 2 * m__p + J__r *
L__p ^ 2 * m__p + 0.4e1 * J__p * J__r) -0.2e1 * (0.2e1 *
B__p * L__r ^ 2 * m__p * r__m + 0.2e1 * B__p * J__r * r__m)
/ r__m / (0.4e1 * J__p * L__r ^ 2 * m__p + J__r * L__p ^ 2 *
m__p + 0.4e1 * J__p * J__r);];
> Matlab(B, resultname="B");
B = [0; 0; -0.1e1 / r__m / (0.4e1 * J__p * L__r ^ 2 * m__p +
J__r * L__p ^ 2 * m__p + 0.4e1 * J__p * J__r) * (-0.1e1 *
L__p ^ 2 * eta__g * eta__m * k__g * k__t * m__p - 0.4e1 *
J__p * eta__g * eta__m * k__g * k__t); 0.2e1 * L__p * L__r *
eta__g * eta__m * k__g * k__t * m__p / r__m / (0.4e1 * J__p
```

```
* L__r ^ 2 * m__p + J__r * L__p ^ 2 * m__p + 0.4e1 * J__p *
J__r);];
> Matlab(C_output, resultname="C");
C = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1;];
```