

```
addpath("TwoLinkRobotMatlabFiles")
```

Problem 1

1.1 Design a controller such that the followign objectives are met for all initial conditions

- For all time, the parameter estimation error, $\tilde{\theta}$, and the tracking error, e , are bounded.
- The tracking error, e , enters the ball $\bar{B}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \epsilon\right)$ in finite time, where $\epsilon > 0$ is some positive constant.
- The constant ϵ can be made arbitrarily small by tuning control gains

Open loop system: $M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + F_s(\dot{q}) + F_d\dot{q} + S_q = \tau + d(t, q, \dot{q})$

where $d(t, q, \dot{q})$ is a nonvanishing disturbance bounded by \bar{d} such that $\|d(t, q, \dot{q})\| \leq \bar{d} \forall (t, q, \dot{q})$ and the given desired trajectory is $q_d(t) \in C^2(\mathfrak{R}_{\geq 0}, \mathfrak{R}^2)$.

The tracking error is defined as $e(t) = q_d(t) - \phi_q(t)$, where $\phi_q(t)$ is the q-component of the trajectory.

Define the closed-loop control system. For conciseness in the solution, system parameters will be dropped from the $M(q)$, $V_m(q, \dot{q})$, $F_s(\dot{q})$, and $d(t, q, \dot{q})$ functions.

$$M(\ddot{q}_d - \ddot{e}) + V_m(\dot{q}_d - \dot{e}) + F_s + F_d(\dot{q}_d - \dot{e}) + S(q_d - e) - \tau - d(t) = 0$$

$$M\ddot{e} = M\ddot{q}_d + V_m(\dot{q}_d - \dot{e}) + F_s + F_d(\dot{q}_d - \dot{e}) + S(q_d - e) - \tau - d(t)$$

Define $r = \dot{e} + \alpha e$ and $M\dot{r} = M\ddot{e} + M\alpha\dot{e}$

$$\text{Then } M\ddot{e} = M\dot{r} - M\alpha\dot{e} = M\ddot{q}_d + V_m(\dot{q}_d - \dot{e}) + F_s + F_d(\dot{q}_d - \dot{e}) + S(q_d - e) - \tau - d(t)$$

$$\dot{r} = M^{-1}(M\ddot{q}_d + V_m(\dot{q}_d - r + \alpha e) + F_s + F_d(\dot{q}_d - r + \alpha e) + S(q_d - e) - \tau - d(t)) + \alpha(r - \alpha e)$$

Define $z = \begin{bmatrix} e \\ r \\ \tilde{\theta} \end{bmatrix}$, where $\theta \in \mathfrak{R}^{9 \times 1}$ represents the system parameters, $\tilde{\theta} = \theta - \hat{\theta}$, and $\hat{\theta}$ represents the estimate of the system parameters

$$g(z, t) = \dot{z} = \begin{bmatrix} \dot{e} \\ \dot{r} \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} r - \alpha e \\ M^{-1}(M\ddot{q}_d + V_m(\dot{q}_d - r + \alpha e) + F_s + F_d(\dot{q}_d - r + \alpha e) + S(q_d - e) - \tau - d(t) + M\alpha(r - \alpha e)) \\ -\hat{\theta} \end{bmatrix}$$

Define $V(z, t) = \frac{1}{2}e^T e + \frac{1}{2}r^T M r + \frac{1}{2}\tilde{\theta}^T \gamma^{-1} \tilde{\theta}$

$$\dot{V}(z) = \frac{\partial V(z, t)}{\partial z} g(z, t) + \frac{\partial V(z, t)}{\partial t} = \begin{bmatrix} e^T & r^T M & \gamma^{-1} \tilde{\theta}^T \end{bmatrix} \begin{bmatrix} r - \alpha e \\ M^{-1}(M\ddot{q}_d + V_m(\dot{q}_d - r + \alpha e) + F_s + F_d(\dot{q}_d - r + \alpha e) + S(q_d - e) - \tau - d(t) + M\alpha(r - \alpha e)) \\ -\hat{\theta} \end{bmatrix}$$

By skew symmetry

$$\dot{V}(z) = e^T(r - \alpha e) + r^T(M\ddot{q}_d + V_m(\dot{q}_d + \alpha e) + F_s + F_d(\dot{q}_d - r + \alpha e) + S(q_d - e) - \tau - d(t) + M\alpha(r - \alpha e)) - \gamma^{-1} \tilde{\theta}^T \hat{\theta}$$

Separation of parameters. Define $Y(e, r)\theta = M\ddot{q}_d + V_m(\dot{q}_d + \alpha e) + F_s + F_d(\dot{q}_d - r + \alpha e) + S(q_d - e) + M\alpha(r - \alpha e)$

$$\dot{V}(z) = e^T(r - \alpha e) + r^T(Y(e, r)\theta - \tau - d(t)) - \gamma^{-1} \tilde{\theta}^T \hat{\theta}$$

$$\dot{V}(z) = e^T r - e^T \alpha e + r^T(Y(e, r)\theta - \tau - d(t)) - \gamma^{-1} \tilde{\theta}^T \hat{\theta}$$

Now we must design τ and $\hat{\theta}$ to stabilize the closed loop system

Let $\tau = Y\hat{\theta} + e + kr$ and $\hat{\theta} = \gamma(Y^T r - \sigma\hat{\theta})$

$$\dot{V}(z) = -e^T \alpha e - r^T kr + \tilde{\theta}^T \sigma \hat{\theta} - r^T d(t)$$

$$\dot{V}(z) = -e^T \alpha e - r^T kr - \tilde{\theta}^T \sigma \tilde{\theta} + \tilde{\theta}^T \sigma \theta - r^T d(t)$$

We know that

$$\frac{1}{2} \min(1, \gamma^{-1}) \|z\|^2 \leq V(z) < \frac{1}{2} \max(1, \gamma^{-1}) \|z\|^2$$

We also know that

$$\dot{V}(z) \leq -\alpha||e||^2 - k||r||^2 - \sigma||\tilde{\theta}||^2 + \sigma||\tilde{\theta}|||\theta| - r||\bar{d}(t)||$$

$$\dot{V}(z) \leq -\min(\alpha, k, \sigma)||z||^2 + \max(0, -||\bar{d}(t)||, \sigma||\theta|)|z|$$

$$\dot{V}(z) \leq -\min(\alpha, k, \sigma)||z||^2 + \sigma||\theta|||z|$$

$$\dot{V}(z) \leq -\frac{1}{2}\min(\alpha, k, \sigma)||z||^2 - \left(\frac{1}{2}\min(\alpha, k, \sigma)||z|| - \sigma||\theta||\right)||z||$$

$$\dot{V}(z) \leq -\frac{1}{2}\min(\alpha, k, \sigma)||z||^2 \text{ iff } \epsilon \geq \frac{2\sigma||\theta||}{\min(\alpha, k, \sigma)}$$

$$\text{Let } \alpha_1 = \frac{1}{2}\min(1, \gamma^{-1})||z||^2, \alpha_2 = \frac{1}{2}\max(1, \gamma^{-1})||z||^2, \text{ and } \alpha_3 = \frac{1}{2}\min(\alpha, k, \sigma)||z||^2$$

1.1.1 Prove that e and $\tilde{\theta}$ are bounded for all time

Let us apply Uniform Ultimate Boundedness

$$\text{H1: } D = \Re^{13}$$

H2: $g(z)$ is Lipschitz continuous and $g(0) = 0$. It satisfies the basic conditions

H3: $V(z)$ is continously differentiable

$$\text{H4: } \alpha_{1,2,3} \text{ are positive definite and continuous iff } \epsilon \geq \frac{2\sigma||\theta||}{\min(\alpha, k, \sigma)}$$

$$\text{H5: } \alpha_{1,2,3} \text{ are K class functions on } \bar{B}(0, r)$$

Thus,

$$||\phi(t, t_0, z^0)|| \leq \beta(z^0, t - t_0) \quad \forall t_0 \leq t \leq t_0 + T$$

$$||\phi(t, t_0, z^0)|| \leq \alpha_1^{-1}(\alpha_2(\epsilon)) \quad \forall t \geq t_0 + T$$

where $\alpha_1^{-1}(\alpha_2(\epsilon)) = \sqrt{\frac{\max(1, \gamma^{-1}) \left(\frac{2\sigma \|\theta\|}{\max(\alpha, k, \sigma)} \right)^2}{\min(1, \gamma^{-1})}}$

1.1.2 Prove that e enters $\bar{B}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \epsilon\right)$ in finite time (ie, prove UES)

H1: $D = \Re^{13}$

H2: $g(z)$ satisfies the basic conditions

H3: $g(0) = 0$

H4: $V(z)$ is continuously differentiable

H5: Let $\underline{v}\|z\|^2 = \alpha_1(z)$, $\bar{v}\|z\|^2 = \alpha_2(z)$, and $v\|z\|^2 = \alpha_3(z)$

H6: Satisfied by H1

Thus, the origin is UGES.

1.1.3 Prove that ϵ can be made arbitrarily small by tuning control gains

By UUB $\|\phi(t, t_0, z^0)\| \leq \sqrt{\frac{\max(1, \gamma^{-1}) \left(\frac{2\sigma \|\theta\|}{\max(\alpha, k, \sigma)} \right)^2}{\min(1, \gamma^{-1})}} \quad \forall t \geq t_0 + T$. By UGES, $T < \infty$.

Thus, $\|\phi(t, t_0, z^0)\| \rightarrow 0$ as a, k become much greater than σ

1.2 MATLAB experiment to track the desired trajectory

1.2.1 Implement your controller in MATLAB using the robo dynamics file available in the Exam 3 folder on Canvas.

Compute $Y(e, r)$ and save to a file

```
clearvars;
syms theta [9 1]
syms x [4 1]
syms tau [2 1]
syms alpha
syms qd dqd ddqd [2 1]
```

```

e = x(1:2);
r = x(3:4) + alpha*e;

q = qd-e;

M = [theta1+2*theta3*cos(q(2)), theta2+theta3*cos(q(2));
      theta2+theta3*cos(q(2)), theta2];
Vm = [theta3*sin(q(2))*-x4, -theta3*sin(q(2))*(-x3-x4);
      theta3*sin(q(2))*-x3, 0];
Fd = diag(theta(4:5));
Fs = theta(6:7).*tanh(-x(3:4));
S = diag(theta(8:9));

f = M*ddqd + Vm*(dq+d+alpha*e)+Fs+Fd*(dq-d+r+alpha*e)+S*(qd-e)+M*(alpha*(r-alpha*e));

Y = sym(zeros(numel(f),numel(theta)));

for i=1:numel(f)
    for j=1:numel(theta)
        temp = coeffs(f(i,:),theta(j),'All');
        if numel(temp) == 2
            Y(i,j) = temp(1);
        end
    end
end

matlabFunction(simplify(Y), 'File', 'Y', 'Vars', {x,alpha,[qd;dq;ddqd]});

```

Simulate the controller

```

clearvars;

p.alpha = 10;    % alpha > 0
p.k      = 100;
p.sigma = 10;
p.gamma = 1;

p.Y = zeros(2,9);
p.e = zeros(2,1);
p.r = zeros(2,1);

% x0 = [q; dq]
x0 = pi*(rand(4,1)-0.5);    % Pick a random starting positions and velocities

% Random initial guess for theta_hat
theta_hat_dot0 = 5*rand(9,1);

y0 = [x0; theta_hat_dot0];

tspan = [0 10];

%% Animation parameters

```

```

L=[1;0.5]; %Link lengths
w=[0.1;0.05]; %Link widths

%%% USE THIS CHECKBOX TO ENABLE ANIMATIONS -> WILL SIGNIFICANTLY INCREASE RUNTIME
%
    enableAnimation = false;
%%%

options = odeset(...
    'OutputFcn',@(t,y,flag) nLinkAnimateOutputFcn(t,y,flag,L,w)...
);
if enableAnimation == 1
    [t,y] = ode45(@(t,y) closedLoopDynamicsDDR(t,y,p), tspan, y0, options);
else
    [t,y] = ode45(@(t,y) closedLoopDynamicsDDR(t,y,p), tspan, y0);
end

q0 = traj(t');
e = q0(1:2,:) - y(:,1:2);

figure
subplot(2,2,1)
plot(t,e, "LineWidth", 2);

xlabel("t (s)", "Interpreter", "latex")
legend("Link 1", "Link2");
title("Model based error over time")
ylabel("e(t) (rad)", "Interpreter", "latex")
grid on;
tau = [];

for i = 1:size(y,1)
    tau = [tau controlDDR(t(i), y(i,:), p)];
end

subplot(2,2,2)

plot(t,tau, "LineWidth", 2);
xlabel('$t$ (s)', 'Interpreter', 'Latex');
ylabel('$\tau(t)$ (Nm)', 'Interpreter', 'Latex');
legend('Link 1', 'Link 2');
grid on;
title("Model based torque")

```

1.2.2 Compare against a PD controller

```

%% Simulation
options = odeset('OutputFcn',@(t,y,flag) nLinkAnimateOutputFcn(t,y,flag,L,w));

if enableAnimation == 1
    [t,x]=ode45(@(t,x) robotDynamicsExam3(t,x,controlFunExam3(t,x),"Smooth"), tspan, x0, opti
    xlabel('$x$ (m)', 'Interpreter', 'Latex')

```

```

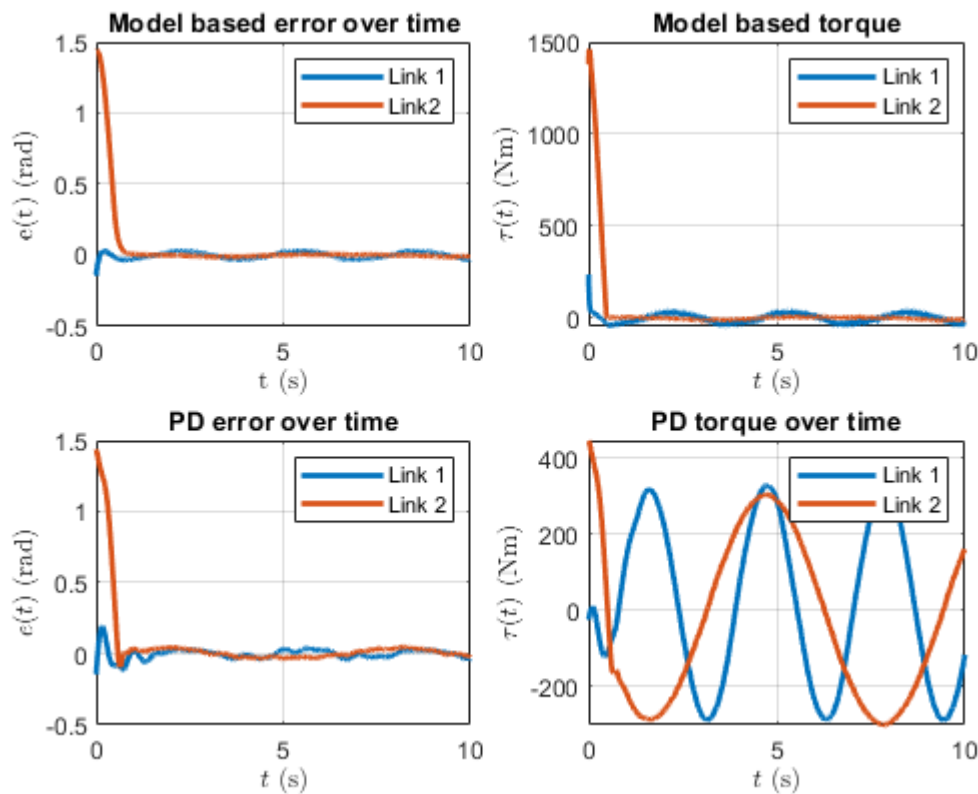
        ylabel('$y$ (m)', 'Interpreter', 'Latex')
else
    [t,x]=ode45(@(t,x) robotDynamicsExam3(t,x,controlFunExam3(t,x),"Smooth"), tspan, x0);
end

%% Plots
x=x';
q = x(1:2,:);
qd = [(1 - exp(-2 * t.')) .* cos(2 * t.')]
      (1 - exp(-2 * t.')) .* sin(t.')]
e = qd - q;

subplot(2,2,3)
plot(t,e, "LineWidth", 2)
xlabel('$t$ (s)', 'Interpreter', 'Latex')
ylabel('$e(t)$ (rad)', 'Interpreter', 'Latex')
legend('Link 1', 'Link 2');
grid on;
title("PD error over time")

subplot(2,2,4)
tau=controlFunExam3(0,x);
plot(t,tau, "LineWidth", 2);
xlabel('$t$ (s)', 'Interpreter', 'Latex');
ylabel('$\tau(t)$ (Nm)', 'Interpreter', 'Latex');
legend('Link 1', 'Link 2');
grid on;
title("PD torque over time")

```



Tuning the PD controller is tricky due to the nonvanishing disturbance. High gains were required to reduce the error to a magnitude comparable to that of the model based controller. The PD controller also exhibits significantly more jerk due to the disturbance. While the model based controller required a significant torque at the initial timestep, the torque decays quickly to less than 100 Nm on each joint. The PD controller continues to require greater than 250 Nm of torque in steady state. This implies that the model aware nature of our derived controller allows for more efficient and robust control.

Problem 2

Question 2 (30 points). Consider the dynamical system

$$\dot{x}_1 = ax_1 + bx_2, \quad \dot{x}_2 = u$$

where $x_1, x_2 \in \mathbb{R}$, a and b are **unknown** constants, and $b > 0$. Design a controller so that x_1 and x_2 decay to zero as time goes to infinity.

Hint: Combine adaptive control and backstepping.

Let $z_1 = x_1$ and $z_2 = x_2 - \phi_1(z_1)$

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \text{ and } f(z) = \dot{z} = \begin{bmatrix} az_1 + b(\phi_1(z_1) + z_2) \\ u - \frac{\partial \phi_1(z_1)}{\partial z_1} (az_1 + b(\phi_1(z_1) + z_2)) \end{bmatrix}$$

Now let $\theta_1 = \frac{a}{b}$ and $\theta_2 = \begin{bmatrix} a \\ b \end{bmatrix}$

$$w = \begin{bmatrix} z \\ \tilde{\theta}_1 \\ \tilde{\theta}_2 \end{bmatrix} \quad g(w) = \begin{bmatrix} \dot{z} \\ -\hat{\theta}_1 \\ -\hat{\theta}_2 \end{bmatrix}$$

$$V(w) = \frac{1}{2b} z_1^2 + \frac{1}{2} z_2^2 + \frac{1}{2} \tilde{\theta}_1^T \tilde{\theta}_1 + \frac{1}{2} \tilde{\theta}_2^T \tilde{\theta}_2$$

$$\frac{\partial V(w)}{\partial w} = \begin{bmatrix} \frac{z_1}{b} & z_2 & \tilde{\theta}_1^T & \tilde{\theta}_2^T \end{bmatrix}$$

$$\dot{V}(w) = \frac{z_1}{b} (az_1 + b(\phi_1 + z_2)) + z_2 \left(u - \frac{\partial \phi_1(z_1)}{\partial z_1} (az_1 + b(\phi_1(z_1) + z_2)) \right) - \tilde{\theta}_1^T \hat{\theta}_1 - \tilde{\theta}_2^T \hat{\theta}_2$$

$$\dot{V}(w) = z_1(\theta_1 z_1 + \phi_1) + z_2 \left(z_1 + u - \frac{\partial \phi_1(z_1)}{\partial z_1} (az_1 + b(\phi_1(z_1) + z_2)) \right) - \tilde{\theta}_1^T \hat{\theta}_1 - \tilde{\theta}_2^T \hat{\theta}_2$$

$$\phi_1 = -\hat{\theta}_1 z_1 - \alpha z_1 = -(\hat{\theta}_1 + \alpha) z_1$$

$$\dot{V}(w) = -\alpha z_1^2 + z_1^2 \tilde{\theta}_1 + z_2 (z_1 + u + (\hat{\theta}_1 + \alpha) (az_1 + b(-(\hat{\theta}_1 + \alpha) z_1 + z_2))) - \tilde{\theta}_1^T \hat{\theta}_1 - \tilde{\theta}_2^T \hat{\theta}_2$$

$$Y = \begin{bmatrix} z_1 & -(\hat{\theta}_1 + \alpha) z_1 + z_2 \end{bmatrix}$$

$$\dot{V}(w) = -\alpha z_1^2 + z_2 (z_1 + u + (\hat{\theta}_1 + \alpha) Y \theta_2) - \tilde{\theta}_1^T \hat{\theta}_1 - \tilde{\theta}_2^T \hat{\theta}_2$$

$$u = -(\hat{\theta}_1 + \alpha) Y \hat{\theta}_2 - z_1 - \beta z_2$$

$$\dot{V}(w) = -\alpha z_1^2 - \beta z_2^2 + z_1^2 \tilde{\theta}_1 + z_2 (\hat{\theta}_1 + \alpha) Y \tilde{\theta}_2 - \tilde{\theta}_1^T \hat{\theta}_1 - \tilde{\theta}_2^T \hat{\theta}_2$$

$$\hat{\theta}_1 = z_1^2$$

$$\hat{\theta}_2 = Y^T z_2 (\hat{\theta}_1 + \alpha)$$

$$\dot{V}(w) = -\alpha z_1^2 - \beta z_2^2$$

$\dot{V}(w)$ is negative semidefinite along w

Apply the invariance principle:

$$1.) D = \mathfrak{R}^5$$

2.) $g(w)$ satisfies the basic conditions

3.) $V(w)$ is continuously differentiable

4.) $V(w)$ is radially unbounded. Given any $\beta > 0$, Ω'_β is a filled ellipse, compact, connected, and in $D = \mathfrak{R}^5$ and is positively invariant. Let $\Omega = \Omega'_\beta$ for any $\beta > 0$.

5.) $\dot{V}(w)$ is negative semidefinite

$$6.) E = \{w \in \Omega \mid \dot{V}(w) = 0\} = \Omega \cap \left\{ \begin{bmatrix} z \\ \tilde{\theta}_1 \\ \tilde{\theta}_2 \end{bmatrix} \in \mathfrak{R}^5 \mid z = 0 \right\}$$

7.) Find $M \subset E$, where M is the largest invariant set in E .

Need M such that

$$\dot{z} = \begin{bmatrix} az_1 + b(-(\hat{\theta}_1 + \alpha)z_1 + z_2) \\ -(\hat{\theta}_1 + \alpha)Y\hat{\theta}_2 - z_1 - \beta z_2 + (\hat{\theta}_1 + \alpha)Y\theta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$-\hat{\theta}_1 = z_1^2 = 0$$

$$-\hat{\theta}_2 = Y^T z_2 (\hat{\theta}_1 + \alpha) = 0$$

Acknowledging $M \in E$

$$\dot{z} = \begin{bmatrix} 0 \\ (\hat{\theta}_1 + \alpha)Y\tilde{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$-\hat{\theta}_1 = 0 = 0$$

$$-\hat{\theta}_2 = 0 = 0$$

Thus, the largest invariant set is $M = \left\{ \begin{bmatrix} z \\ \tilde{\theta}_1 \\ \tilde{\theta}_2 \end{bmatrix} \in \mathbb{R}^3 \mid z, \tilde{\theta}_2 = \vec{0} \right\}$. Given any $\zeta^0 \in \Omega$, $\lim_{t \rightarrow \infty} \text{dist}(\phi(t, \zeta^0), M) = 0$ and

$\lim_{t \rightarrow \infty} \phi_z(t, \zeta^0) = 0$. The system is LAS.

In other words, z_1, z_2 decay to zero as time goes to infinity. Backstepping introduces an affine transformation from $x \rightarrow z$, implying that x_1, x_2 also decay to zero.

Problem 3

Question 3 (20 points). Consider a dynamical system $\dot{x} = f(t, x)$ where $f : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ satisfies the basic conditions. Furthermore, we know that there exists a function $V \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R})$, a constant $a > 0$ and functions $\alpha_1 \in \mathcal{K}([0, a], \mathbb{R})$, $\alpha_2 \in \mathcal{K}([0, a], \mathbb{R})$, and $\alpha_3 \in \mathcal{K}([0, a], \mathbb{R})$ such that for all $(t, x) \in \mathbb{R}_{\geq t_0} \times B(0, a)$, we have

$$\alpha_1(\|x\|) \leq V(t, x) \leq \alpha_2(\|x\|) \quad (1)$$

$$\frac{\partial V}{\partial x}(t, x) f(t, x) + \frac{\partial V}{\partial t}(t, x) \leq -\alpha_3(\|x\|) \quad (2)$$

3.1 Show that the origin is UAS

H1: Will come back to this...

H2: $f(t, x)$ satisfies the basic conditions, ie: is locally Lipschitz and piecewise continuous in t

H3: $f(t, x)$ satisfies the basic conditions

H4: V is stated to be continuously differentiable

H5: $\alpha_1, \alpha_2, \alpha_3$ are K class functions, thus are positive definite and are continuous

H6: Satisfied by (1) iff $\|x\| < a$

H7: Satisfied by (2) iff $\|x\| < a$

H6 & H7 imply that $D = B(0, a)$ (H1)

All hypothesis are satisfied and the origin is UAS.

3.2 Estimate the region of attraction of the origin in terms of the functions α_i and the constant a

The region of attraction consists of all x such that α_i are defined. Thus, it may be defined as $B(0, a)$.

This is evident as the origin is UAS when $\|x\| < a$.

```
function [dy] = closedLoopDynamicsDDR(t, x, p)
    dx = robotDynamicsExam3(t, x, controlDDR(t, x, p), "Smooth");
    theta_hat_dot = updateLawDDR(t, x, p);
    dy = [dx; theta_hat_dot];
end

function [tau] = controlDDR(t, x, p)
    q0 = traj(t);

    p.e = q0(1:2) - x(1:2);
    de = q0(3:4) - x(3:4);
    p.r = de + p.alpha*p.e;

    p.Y = Y(x,p.alpha,q0);

    %tau = max(min(p.Y*x(5:13) + p.e + p.k*p.r, 60), -60);
    tau = p.Y*x(5:13) + p.e + p.k*p.r;
end

function [ tau ] = controlFunExam3( t,x )
    %Controller Design
    kp=[300 0;0 300];
    kd = [10 0; 0 10];

    qd = [(1-exp(-2*t)) * cos(2*t);
          (1-exp(-2*t)) * sin(t)];
    dqd = [2.*sin(2.*t).*(exp(-2.*t) - 1) + 2.*cos(2.*t).*exp(-2.*t);
           2.*exp(-2.*t).*sin(t) - cos(t).*(exp(-2.*t) - 1)];

    e=qd-[x(1,:);x(2,:)];
    de = dqd - [x(1,:);x(2,:)];

    tau = kp*e + kd*de;
end

function theta_hat_dot = updateLawDDR(~, x, p)
    theta_hat_dot = p.gamma*(p.Y'*p.r - p.sigma*x(5:13));
end
```

```

function [q0] = traj(t)
    qd = [(1-exp(-2.*t)) .* cos(2.*t);
          (1-exp(-2.*t)) .* sin(t)];

    dqd = [2.*sin(2.*t).*(exp(-2.*t) - 1) + 2.*cos(2.*t).*exp(-2.*t);
           2.*exp(-2.*t).*sin(t) - cos(t).*(exp(-2.*t) - 1)];

    ddqd = [-4.*cos(2.*t) - 8.*sin(2.*t).*exp(-2.*t);
            4.*exp(-2.*t).*cos(t) - sin(t) - 3.*exp(-2.*t).*sin(t)];

    q0 = [qd;dqd;ddqd];
end

```