Nonlinear RISE-Based control of an n-link, rigid, revolute, serially connected robot manipulator

Arif Mohammed

Introduction

This Report focuses on the design of a Link position tracking controller design for an n-link, rigid, revolute serially connected robot manipulator using a continuous robust integral of the sign od the error control structure to compensate for the system uncertainties and sufficiently smooth bounded exogenous disturbances. A Lyapunov stability analysis is included to prove semiglobal asymptotic tracking. The proposed control scheme was implemented on a 2-link manipulator arm built by Integrated motion Inc. (IMI).

Problem Formulation:

The mathematical model for the mechanical dynamics of n-link revolute, rigid, direct drive robot is given as

$$\tau = M(q)\,\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) + F_d\dot{q} + \tau_d\,.$$

where τ_d is unmodeled exogenous disturbance. M(q) $\epsilon\,\mathbb{R}^{n\,x\,n}$, $V_m(q,\dot{q})\epsilon\mathbb{R}^{n\,x\,n}, G(q)\epsilon\,\mathbb{R}^{n}, F_d\epsilon\,\mathbb{R}^{n\,x\,n}$ accounts for the system dynamics as they describe inertia, Coriolis-centripetal;, gravity, and frictional forces respectively.

The properties and assumptions of the dynamic system are stated in Appendix 1. In Appendix – 1 the error system is developed. Followed by the development of the Lyapunov member and its stability is proven.

As the result of the stability analysis, it was found that the rise control will give a semi global asymptotic stability.

Once the error equations of e_1, e_2, r are defined the mathematical modelling of the system and controller is designed and is simulated. By trial and the Controller gains and error constants are determined and they were found to be as follows

The input torque was simulated using the equation $\tau = Y_d *\ \hat{\theta} + \mu$

The remaining dynamic parse states were simulated by dynamics as

$$\dot{e_1} = e_2 - \alpha_1 * e_1$$

$$\dot{e}_2 = r - \alpha_2 * e_2$$

$$\dot{\mu} = (K+1)r + \beta * SGN(e_2)$$

$$r = \frac{M(q)}{S - \tau + \tau_d}$$

$$S = V * \dot{q} + f_d * q + M(q) * \dot{q_d} +\ \alpha_1 * M(q) * e_2 - \alpha_1^2 * M(q) * e_2 + M(q) * \alpha_2 * e_2;$$

After trial and error of the system the best fit controller and error constants were decided to be

$$K = 50.$$

$$\alpha_1 = 2;$$

$$\alpha_2 = 2;$$

$$\beta = 3;$$

Results:

Nonlinear Rise controller is designed and tuned for a 2-link rigid, revolute, direct drive robot. The results were plotted on time scale.

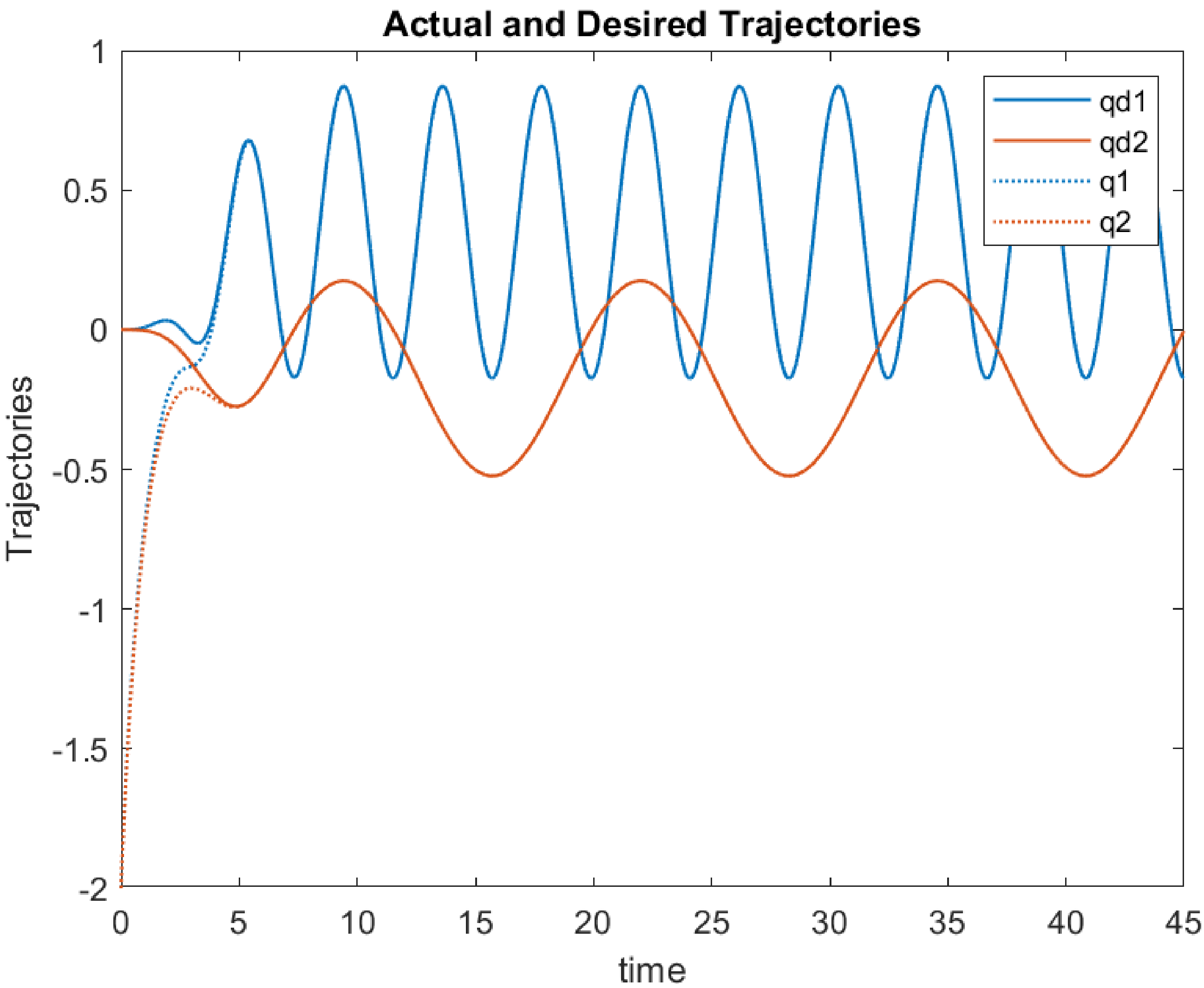

Fig1 Actual and desired trajectories of the two links.

In figure qd1, qd2 the thick lines are the desired trajectory of the link and the q1, q2 the dotted lines are the actual output trajectory of the links. From this figure it is evident that the link position tracking is the same as desired and the errors in tracking are led to zero. The system shows perfect tracking of the input command.

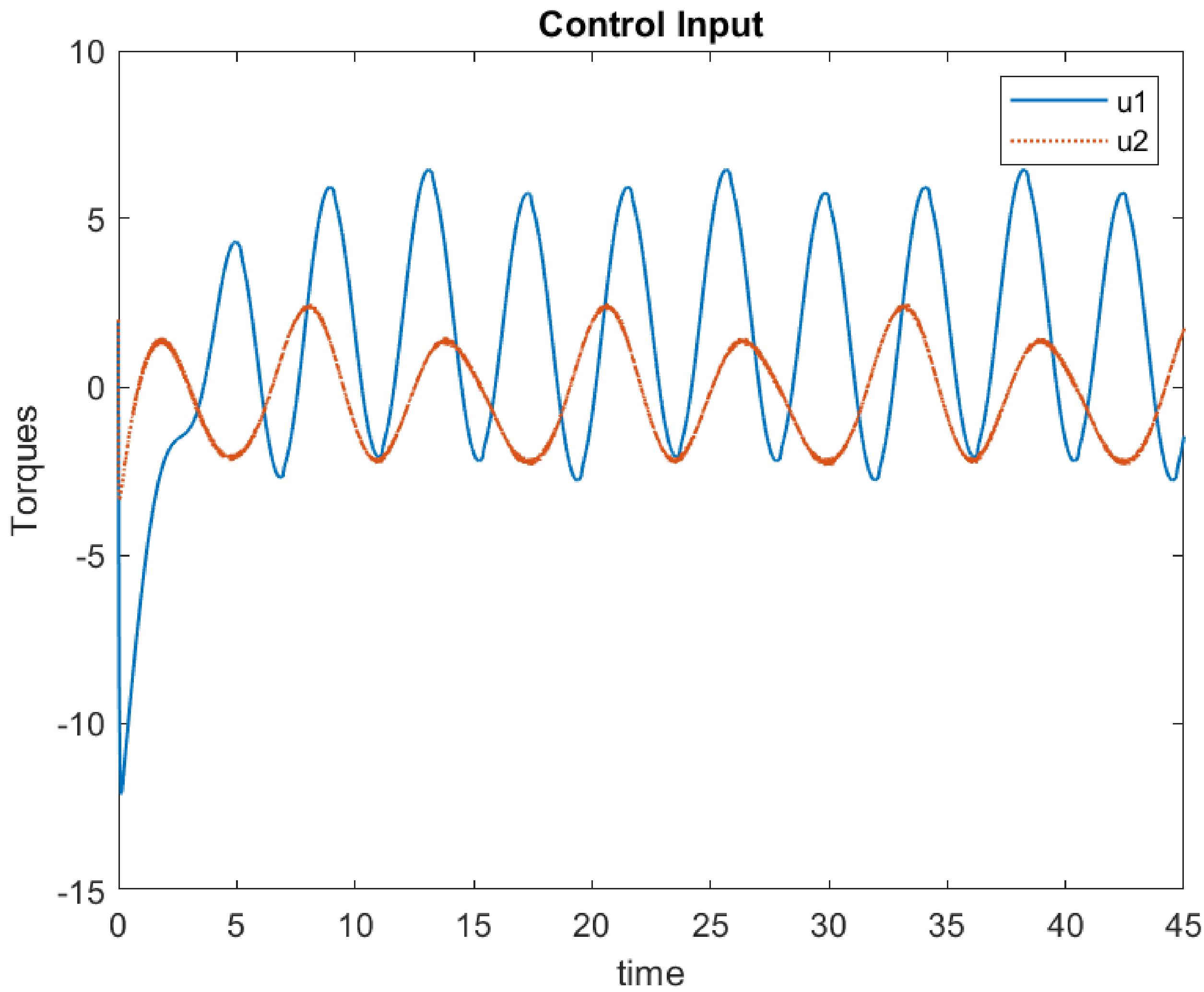

Figure 2 Input torques for the 2 links.

A desired trajectory is simulated using the desired trajectory function as mentioned in the MATLAB code in the appendix.

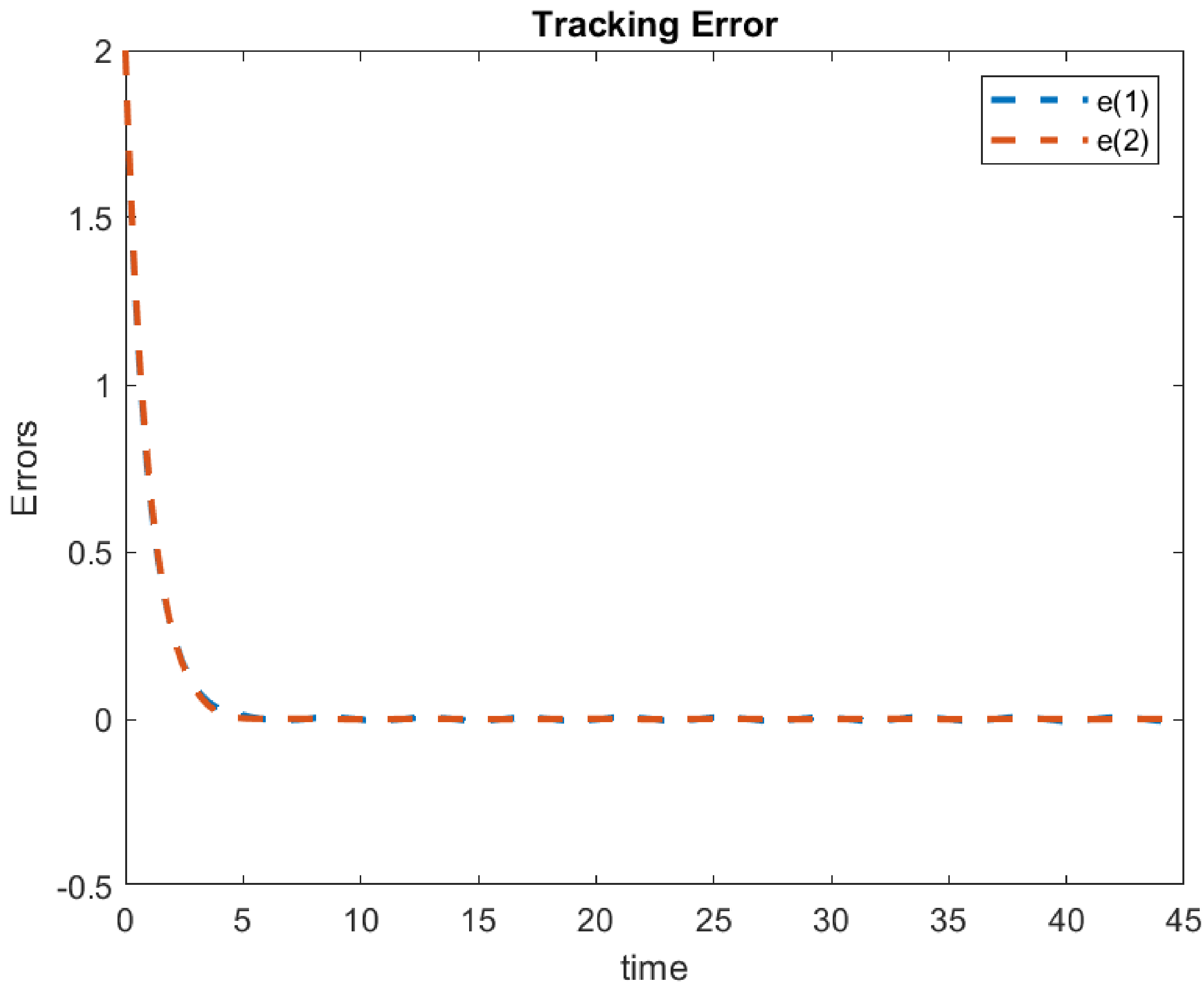

Figure 3 tracking errors $e_1(t), e_2(t)$

It is evident from the figure 3 that the tracking error is reaching to zero as the system reaches infinity

$$\lim_{t \to \infty} e_1(t) = 0;$$

It is also seen that the filtered error 1 is wobbling around the stability point. As this is claimed to asymptotic stability and the error is within a upper and lower bound which shows it doesn't go to instability. As time passes the system gain perfect stability.

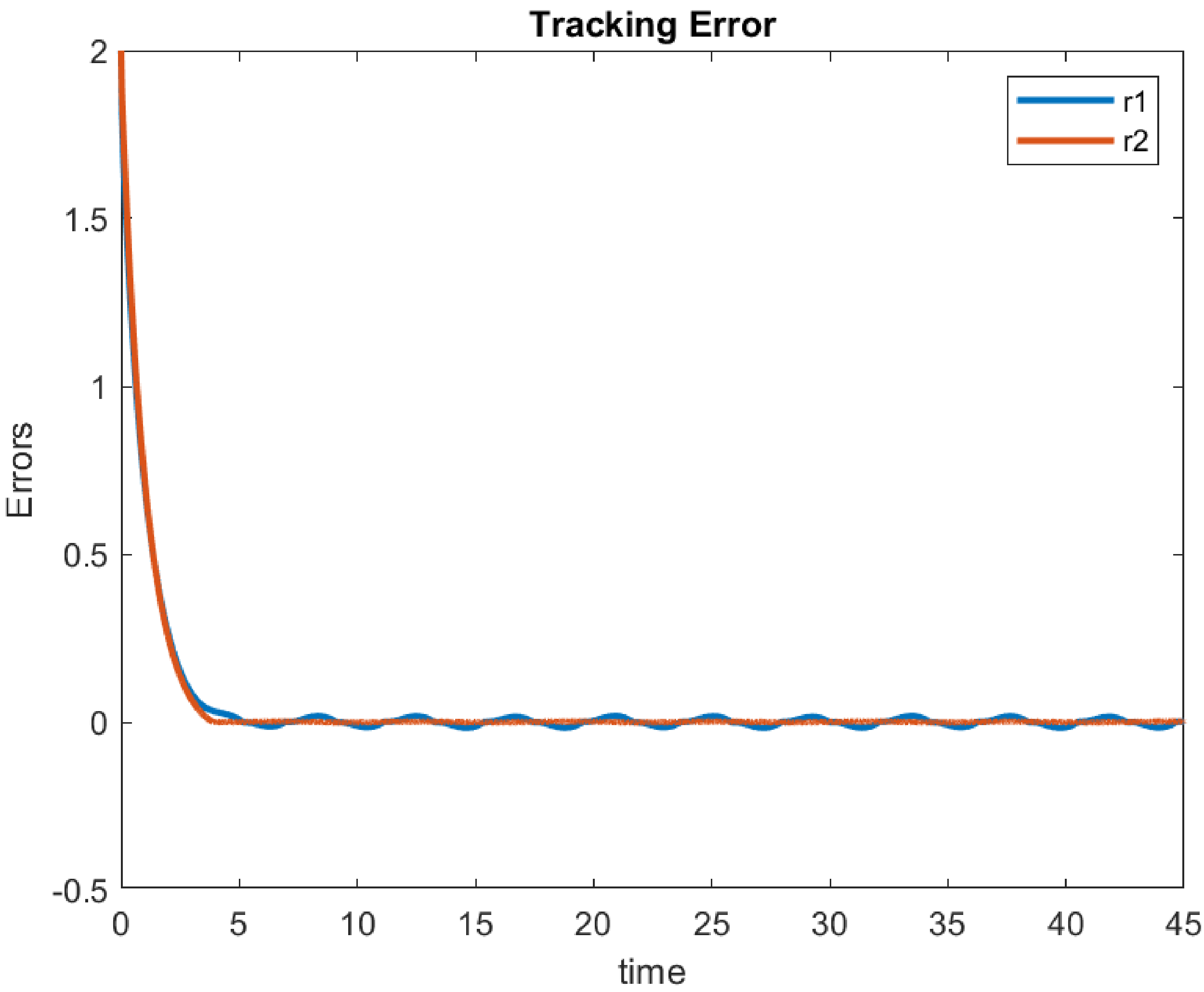

Figure 4 filtered tracking error r(t)

It is evident from the figure 3 that the tracking error is reaching to zero as the system reaches infinity

$$\lim_{t \to \infty} r(t) = 0;$$

It is also seen that the filtered error r is wobbling around the stability point. As this is claimed to asymptotic stability and the error is within an upper and lower bound which shows it doesn't go to instability. As time passes the system gain perfect stability.

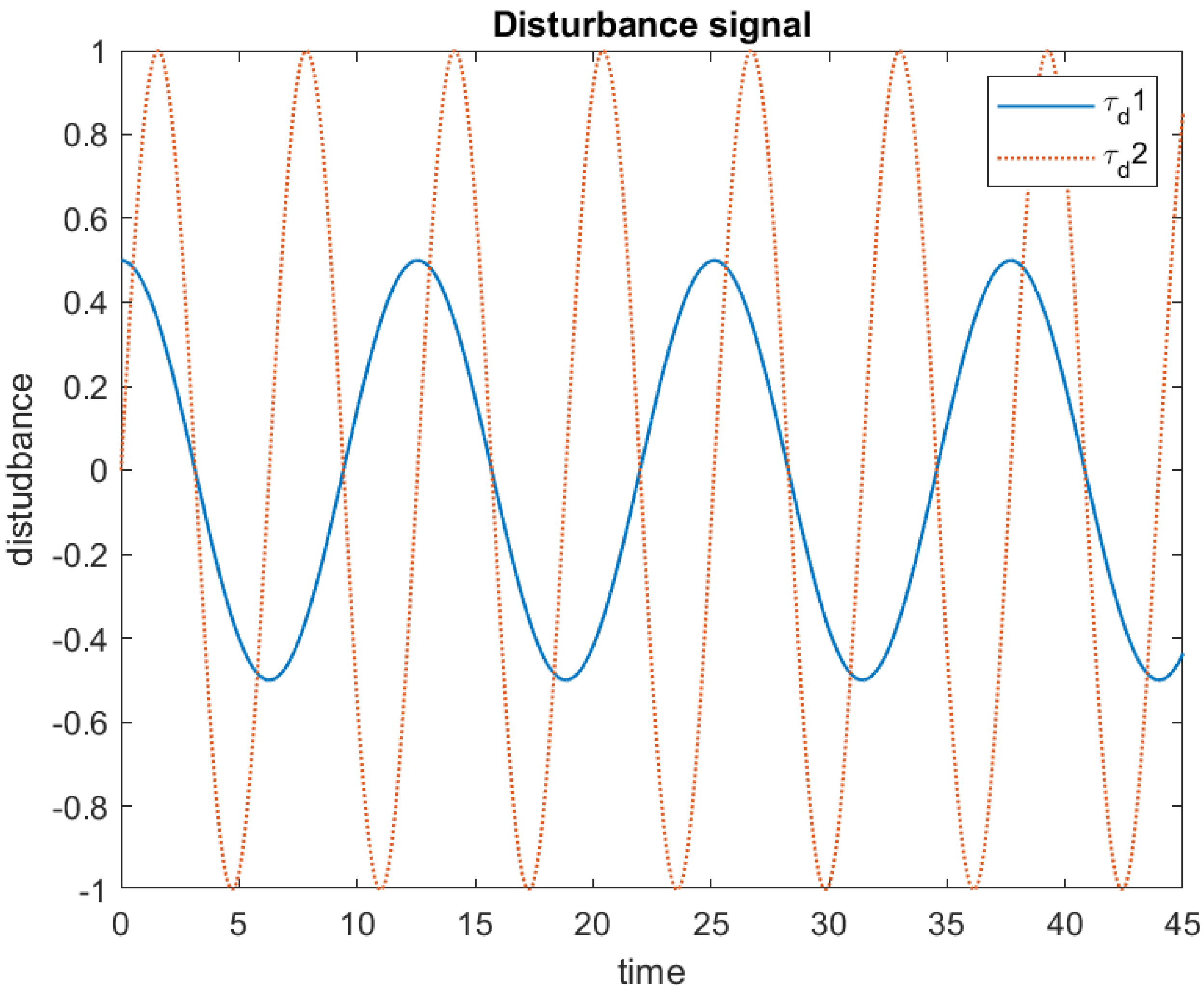

Figure 5 shows the disturbances introduced into the system τ_d

These are sinusoidal signals of different wavelength and phase. This is included to check the robustness of the system due to unmodeled disturbances caused by external disturbances.

Conclusion:

A nonlinear controller is designed for Link position tracking controller design of an n-link, rigid, revolute serially connected robot manipulator using a continuous robust integral of the sign of the error control structure to compensate for the system uncertainties and sufficiently smooth bounded exogenous disturbances. A Lyapunov stability analysis is included to prove semiglobal asymptotic tracking. The proposed control scheme was implemented on a 2-link manipulator arm built by Integrated motion Inc. (IMI). The results are plotted with the gains mentioned.

References:

1. A redesigned DCAL controller without velocity measurements: theory and demonstration* by T. Burg, D. Dawson, and P. Vedagarbha

2. F. Lewis, C. Abdallah and D. Dawson, Control of Robot Manipulators (New York Macmillan Publishing Co., 1993).

3. W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach, Birkhauser Boston, 2003, ISBN: 0- 8176-4265-X.

4. A. Behal, W. E. Dixon, B. Xian, and D. M. Dawson, Lyapunov-Based Control of Robotic Systems, Taylor and Francis, 2009, ISBN: 0849370256.

5. EML 6350 – Nonlinear control lecture 26 by Dr. Warren Dixon (Robust integral of the sign of the error Controller design and Lyapunov Analysis).

Appendix

```
close all; clear all; clc;
% RISE Based Non-Linear controller for 2-Link rigid, revolute, direct drive
% robot Arm.
% 2 - link IMI arm dynamics
p1      = 3.473;   % kg.m^2
p2      = 0.196;   % kg.m^2
p3      = 0.242;   % kg.m^2
f1      = 5.3;     % Nm.sec
f2      = 1.1;     % Nm.sec
% Constant system parameters
theta    = [p1;p2;p3;f1;f2];
% Simulation time
tf      = 45;
% initial coditions vector X0 = [e0;r0;thetahat0];
X0      = 2*[ones(6,1)];
% ode options
options  = odeset('RelTol',1e-3,'AbsTol',1e-3);

% Integration of Dynamic system model
[t,X]    = ode45(@(t,X) twoLinkdynamics(t,X,theta),[0 tf],X0,options);
% initializing the input vector
u        =   zeros(2,length(t));
for i     =   1:length(t)
   u(:,i)  =   getcontrol(t(i),X(i,:),theta);
end
% Desired trajectory
for i   =      1:length(t)
   t1 = t(i);
   qd(:,i)   = [(30*(pi/180)*sin(1.5*t1)+20*(pi/180))*(1-exp(-0.01*t1^3)); -
(20*(pi/180)*sin(0.5*t1)+10*(pi/180))*(1-exp(-0.01*t1^3))];
   Td(:,i)   = [0.5*cos(0.5*t1);sin(t1)];

end
% Computing the error
e          =      X(:,1:2)';
r          =      X(:,3:4)';
Usign      =      X(:,5:6)';
% Computing the actual link parameter(Trajectory) q
q          =      qd-e;

% Trajectory Plots ( Actual, Desired Trajector V/s Time)
figure(1)
plot(t,qd,'-','LineWidth',1)
hold on
ax = gca;
```

```matlab
ax.ColorOrderIndex = 1;
plot(t,q,':','LineWidth',1)
title('Actual and Desired Trajectories')
xlabel('time') % x-axis label
ylabel('Trajectories') % y-axis label
legend('qd1','qd2','q1','q2');
hold off

% Tracking error plot
figure(2)
plot(t,e,'--','LineWidth',2)
title('Tracking Error')
xlabel('time') % x-axis label
ylabel('Errors') % y-axis label
legend('e(1)','e(2)')

% Control Input Plot
figure(3)
plot(t,u(1,:),'-',t,u(2,:),':','LineWidth',1)
title('Control Input')
xlabel('time') % x-axis label
ylabel('Torques') % y-axis label
legend('u1','u2');

%Disturbance Vector plot
figure(4)
plot(t,Td(1,:),'-',t,Td(2,:),':','LineWidth',1)
title('Disturbance signal')
xlabel('time') % x-axis label
ylabel('distudbance') % y-axis label
legend('\tau_d1','\tau_d2')

% Filtered Tracking error plot
figure(5)
plot(t,r,'-','LineWidth',2)
title('Tracking Error')
xlabel('time') % x-axis label
ylabel('Errors') % y-axis label
legend('r1','r2')

% Filtered Tracking error plot
figure(6)
plot(t,Usign,'-','LineWidth',2)
title('Parse state Usign')
xlabel('time') % x-axis label
ylabel('Errors') % y-axis label
```

```
legend('Usign1','Usign2')

function [XDot] = twoLinkdynamics(t,X,theta)
global e2i
% Parse parameter vector
p1 = theta(1);
p2 = theta(2);
p3 = theta(3);
f1 = theta(4);
f2 = theta(5);

% Controller Gains tuning
K       = 50;
a1      = 2;
a2      = 2;
B       = 3;

% Desired trajectory and needed derivatives
[qd,qdDot,qdDotDot] = des_trajectory(t);
%  Parse State Computing
e       = [X(1);X(2)];
e2      = [X(3);X(4)];
Usign   = [X(5);X(6)];

% Compute current q and qDot for convenience
q       = qd-e;
qDot    = -e2 + a1*e + qdDot;

% Compute cos(x2) and sin(x2) for convenience
c2      = cos(q(2));
s2      = sin(q(2));

% Compute current matrices for the dynamics
M       = [p1 + 2*p3*c2 p2 + p3*c2;p2 + p3*c2 p2];
V       = [-p3*s2*qDot(2) -p3*s2*(qDot(1) + qDot(2));p3*s2*qDot(1) 0];
fd      = [f1 0;0 f2];
Td      = [0.5*cos(0.5*t);2*sin(t)];

S = V*qDot+fd*q+M*qdDot+a1*M*e2-a1^2*M*e2+M*a2*e2;

% Design controller
if (t==0)
   e2i = e2;
end
u       =(K+1)*(e2-e2i)+Usign;
r       = M\(S-u+Td);
```

```
%udot=(K+1)*r+B*sign(e2);
% Compute current closed-loop dynamics
eDot      = e2 - a1*e;
e2Dot     = r - a2*e2;
UsignDot   = (K+1)*r+B*sign(e2);

% Stacked dynamics vector (XDot is the same size and "form" as X)
XDot      = [eDot;e2Dot;UsignDot];
end
% Desired Trajectory function
function [qd,qd_dot,qd_dot_dot] = des_trajectory(t)

qd = [(30*(pi/180)*sin(1.5*t)+20*(pi/180))*(1-exp(-0.01*t^3)); -
(20*(pi/180)*sin(0.5*t)+10*(pi/180))*(1-exp(-0.01*t^3))];
qd_dot = [    (3*t^2*((6283*sin((3*t)/2))/12000 +
6283/18000))/(100*exp(t^3/100)) - (6283*cos((3*t)/2)*(1/exp(t^3/100) -
1))/8000;
      (6283*cos(t/2)*(1/exp(t^3/100) - 1))/36000 -
(3*t^2*((6283*sin(t/2))/18000 + 6283/36000))/(100*exp(t^3/100)) ];
qd_dot_dot=[(18849*t^2*cos((3*t)/2))/(400000*exp(t^3/100)) +
(3*t*(6283/18000+(6283*sin((3*t)/2))/12000))/(50*exp(t^3/100))-
(9*t^4*(6283/18000+(6283*sin((3*t)/2))/12000))/(10000*exp(t^3/100))+(18
849*(-1+exp(-t^3/100))*sin((3*t)/2))/16000;
   (-6283*t^2*cos(t/2))/(600000*exp(t^3/100))-
(3*t*(6283/36000+(6283*sin(t/2))/18000))/(50*exp(t^3/100))+(9*t^4*(6283/
36000+(6283*sin(t/2))/18000))/(10000*exp(t^3/100))-(6283*(-1+exp(-
t^3/100))*sin(t/2))/72000];
end

function [u] = getcontrol(t,X,theta)
global e2i
% Parse parameter vector
p1 = theta(1);
p2 = theta(2);
p3 = theta(3);
f1 = theta(4);
f2 = theta(5);

% Select gains for controller
K      = 50;
a1      = 2;
a2      = 2;
% Desired trajectory and needed derivatives
[qd,qdDot,qdDotDot] = des_trajectory(t);
% Current error [e;r;thetahat])
e         = [X(1);X(2)];
```

```
e2          = [X(3);X(4)];
Usign       = [X(5);X(6)];
% Compute current x and xDot for convenience
q          = qd-e;
qDot       = -e2 + a1*e + qdDot;

% Compute cos(q2) and sin(q2) for convenience
c2          = cos(q(2));
s2          = sin(q(2));

% Compute current matrices for the dynamics
M           = [p1 + 2*p3*c2 p2 + p3*c2;p2 + p3*c2 p2];
V          = [-p3*s2*qDot(2) -p3*s2*(qDot(1) + qDot(2));p3*s2*qDot(1) 0];
fd          = [f1 0;0 f2];
Td          = [0.5*cos(0.5*t);sin(t)];

% S = V*qDot+fd*q+M*qdDot+a1*M*e2-a1^2*M*e2+M*a2*e2;
if (t==0)
   e2i = e2;
end
u       = (K+1)*(e2-e2i)+Usign;
end
```

Appendix 2

Dynamics of n-dof robot manipulator and Lyapunov based stability analysis of the RISE (Robust Integral of the sign of the error) Controller.

Robust Integral of the Sign of the Error

Dynamics of n-link, rigid, revolute, direct drive robot manipulator.

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + f(\dot{q}) + G(q) + \tau_d \triangleq \tau.$$

where $M(q) \in \mathbb{R}^{n\times n} \rightarrow$ Inertia matrix

$V_M(q,\dot{q}) \in \mathbb{R}^{n\times n} \longrightarrow$ Coriolis – Centripetal matrix

$G(q) \in \mathbb{R}^{n} \longrightarrow$ Gravity

$F(\dot{q}) \in \mathbb{R}^{n\times n} \longrightarrow$ frictional forces.

$\tau_d \longrightarrow$ unmodeled exogenous disturbance. bounded by

$\|\tau_d\|, \|\dot{\tau}_d\|, \|\tau_d\| \leq C$

Assumptions and properties of the dynamic model:

① $x^T\left(\frac{1}{2}\dot{M}(q) - V_m(q,\dot{q})\right)x = 0$ (Skew Symmetric property)

② $M(q) > 0$

③ $\underline{M}(q) \leq M(q) \leq \overline{M}(q)$

④ $\underline{V_m}(q,\dot{q}) \leq V_m(q,\dot{q}) \leq \overline{V_m}(q,\dot{q})$

⑤ $\underline{f}(\dot{q}) \leq f(\dot{q}) \leq \bar{f}(\dot{q})$

Defining error System

$$e_1(t) \triangleq q_d - q$$

$$e_2(t) \triangleq \dot{e}_1 + \alpha_1 e_1$$

$$r(t) \triangleq \dot{e}_2 + \alpha_2 e_2$$

→ Dynamic modification

$$M(q)r = M(q)(\dot{e}_2 + \alpha_2 e_2) = m(\ddot{e}_1 + \alpha\dot{e}_1) + \alpha_2 m e_2 = m(\ddot{q}_d - \ddot{q} + \alpha\dot{e}_1) + \alpha_2 m e_2$$

$$= M(q)\ddot{q}_d + V_M(q,\dot{q})\dot{q} + f(\dot{q}) + G(q) + \tau_d - \tau + M(q)(\alpha_1\dot{e}_1 + \alpha_2 e_2)$$

let $y_d\theta$ be the model based feedforward term.

Add & Subtract $y_d\theta$ on RHS

$$y_d\theta = M(q_d)\ddot{q}_d + V_m(q_d, \dot{q}_d)\dot{q}_d + f(\dot{q}_d) + G(q_d)$$

$$y(q,\dot{q},\ddot{q}_d)\theta \triangleq M(q)\ddot{q}_d + V_m(q,\dot{q})\dot{q} + f(\dot{q}) + G(q)$$

$$\Rightarrow M(q)r = M(q)\ddot{q}_d + V_m(q,\dot{q})\dot{q} + f(\dot{q}) + G(q) + \tau_d - \tau + M(q)(\alpha_1\dot{e}_1 + \alpha_2 e_2) + y_d\theta - y_d\theta$$

$$= \underbrace{y\theta - y_d\theta + M(q)(\alpha_1\dot{e}_1 + \alpha_2 e_2)}_{S_1} + y_d\theta + \tau_d - \tau$$

$$\boxed{M(q)r = S_1 + y_d\theta + \tau_d - \tau.} \quad \text{— (1)}$$

let $z(t)$ be Composite errors

$$z(t) = \begin{bmatrix} e_1^T & e_2^T & r^T \end{bmatrix}^T$$

Using the mean value theorem $\|S\| \leq \xi_1 \|z\|$

Design Control input as

$$\tau = y_d \cdot \hat{\theta} + \mu_1$$

By Control upadation laws

$$\dot{\hat{\theta}} = \Gamma \dot{y}_d r + \Gamma \dot{y}_{df}^T \varepsilon(t)$$

$$\cdot\ \dot{\mu}_1(t) \triangleq (k_1+1)r(t) + \beta\, \mathrm{sgn}(e_2)$$

taking time derivative of (1) & the closed loop error System

$$\dot{M}(q)r + M(q)\dot{r} = \dot{y}_d\tilde{\theta} - y_d\dot{\hat{\theta}} + \dot{S}_1 + \dot{\tau}_d - \dot{\mu}_1$$

$$M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r + \dot{y}_d\tilde{\theta} - y_d(\Gamma\dot{y}_d^T \cdot r + \Gamma\dot{y}_{df}^T\varepsilon) - \frac{1}{2}\dot{M}(q)r + \dot{S}_1 + \dot{\tau}_d - \dot{\mu}_1 + e_2 r$$

$$= -\frac{1}{2}\dot{M}(q)r + \dot{y}_d\tilde{\theta} - y_d\Gamma\dot{y}_{df}^T z - e_2 - (k_1+1)r - \beta\,\mathrm{sgn}(e_2) - \frac{1}{2}\dot{M}(q)r$$
$$- y_d\Gamma\dot{y}_d^T r + \dot{S}_1 + e_2 + \dot{\tau}_d$$

let

$$\tilde{N}_1(t) \triangleq -\frac{1}{2}\dot{M}(q)r + \dot{S}_1 + e_2 - y_d\cdot\Gamma\dot{y}_d^T r$$

$$\dot{\tau}_d \triangleq N_{1B}$$

$\because$ all the terms in $\|\tilde{N}_1\|$ are known or measured

$$\|\tilde{N}_1\| \leq \rho_1(\|z\|)(\|z\|)$$

$$\|N_{1B}\| \leq \xi_2 \quad \|\dot{N}_{1B}\| \leq \xi_3$$

Prediction error $\varepsilon(t) \triangleq \tau_f - \hat{\tau}_f$

where $\tau_f \triangleq f^*(y\theta + \tau_d)$

$= f^*(y_d \cdot \theta + y\theta - y_d\theta) + \tau_d$

let $y\theta = x_1$

$y_d\theta = x_{d1}$

$f^* x_1 = \dot{f} \times m\dot{q} + f(0) m\dot{q} - fm(q(0))\dot{q}(0) + f * g$

$= x_{1f}(q, \dot{q}) + \omega(t)$

$f * x_{d1} = x_{df}(q_d, \dot{q}_d) + \omega_d$

where $\omega_d \triangleq -f\, m(q_d(0))\dot{q}_d(0)$

$\tau_f = y_{df}\theta + x_1 f - x_1 df + \omega - \omega_d + \tau_d f$

design $\hat{\tau}_f \triangleq y_d f(q_d, \dot{q}_d)\hat{\theta} + \mu_2(t)$

$\varepsilon = y_{df}\tilde{\theta} + x_1 f - x_1 df + \omega - \omega_d + \tau_{df} - \mu_2$

$\frac{d_\varepsilon}{dt} \Downarrow$

$\dot{\varepsilon} = \dot{y}_{df}\tilde{\theta} - y_{df}\Gamma \dot{y}_{df}^T\varepsilon + \tilde{N}_2 + N_{2B} - \dot{\mu}_2$

where

$\dot{\mu}_2 \triangleq k_2\varepsilon + \beta_2 \operatorname{sgn}(\varepsilon)$

$N_{2B} \triangleq \dot{\omega} - \dot{\omega}_d + \dot{\tau}_{df}$

$\tilde{N}_2 \triangleq \dot{x}_1 f - \dot{x}_{1d} - y_d f\, \Gamma \dot{y}_d^T r$

The terms $\|N_{2B}\| \le \xi_4$

$\|\tilde{N}_2\| \le \rho_2(\|z\|)(\|z\|)$

Lyapunov's Analysis:

$V(M(q), e_1, e_2, r, \varepsilon, P_1, P_2, \tilde{\theta}) \triangleq$

$$\frac{1}{2} r^T M(q) r + \frac{1}{2} e_1^T e_1 + \frac{1}{2} e_2^T e_2 + \frac{1}{2}\varepsilon^T \cdot \varepsilon + P_1 + P_2 + \frac{1}{2}\tilde{\theta}^T \Gamma^{-1}\tilde{\theta}$$

where $\dot{P}_1 \triangleq -L_1$

$\dot{P}_2 \triangleq -L_2$

design $L_1 = r^T \left(N_{1B} - \beta_1 \operatorname{sgn}(e_2) \right)$

$L_2 = \varepsilon^T \left(N_{2B} - \beta_2 \operatorname{sgn}(e_2) \right)$

$\dot{P}_1 = -r^T \left(N_{1B} - \beta_1 \operatorname{sgn}(e_2) \right)$

$\dot{P}_2 = -\varepsilon^T \left(N_{2B} - \beta_2 \operatorname{sgn}(e_2) \right)$

To prove lyapunov fn to be positive both P_1 & P_2 should be positive

$$\dot{P} = -L \Rightarrow L = r^T \left(N_{1B} - \beta \operatorname{sgn}(e_2) \right)$$

$$P = \beta \|e_2(0)\| - e_2(0)\, N_{1B} - \int_0^t L(\sigma)\, d\sigma \quad \text{[Integration chain rule].}$$

In order to show P is positive, the conditions should be satisfied.

① $\int_0^t L(\sigma)\, d\sigma \le \beta \|e_2(0)\| - e_2^T(0)\, N_{1B}(0)$

② $\beta \|e_2(0)\| - e_2^T(0)\, N_{1B}(0) > 0$

Condition - ①

$$\int_0^t L(\sigma)\, d\sigma = \int_0^t r^T \left(N_{1B} - \beta \operatorname{sgn}(e_2) \right) \Rightarrow \text{Sub } r = \dot{e}_2^T + \alpha_2 e_2^T$$

$$= e_2^T N_{1B} - e_2^T(0)\, N_{1B}(0) - \beta \|e_2(t)\| + \beta \|e_2(0)\| + \int_0^t \alpha_2 e_2^T \left(N_{1B} - \frac{1}{\alpha_2} \dot{N}_{1B} - \beta \operatorname{sgn}(e_2) \right)$$

$$\le \|e_2(t)\| \bar{c}_1 - \beta \|e_2(t)\| + \beta \|e_2(0)\| - e_2^T(0)\, N_{1B}(0) + \int_0^t \alpha_2 \|e_2\| \left(\bar{c}_1 + \frac{1}{\alpha_2} \bar{c}_2 - \beta \right)$$

assume

$$\beta > \bar{c}_1 + \frac{1}{\alpha_2} \bar{c}_2$$

$$\int_0^t L(\sigma)\, d\sigma \le \beta \|e_2(0)\| - e_2^T(0)\, N_{1B}(0)$$

To prove condition - 2

$$e_2^T(0)\, N_{1B}(0) \le \|e_2(0)\| \bar{c}_1 \Rightarrow \beta \|e_2(0)\| - e_2^T(0)\, N_{1B}(0) \le \|e_2(0)\| (\beta - \bar{c}_1)$$

$$\le 0$$

$$\dot V = \frac{1}{2} r^T \dot M r + r^T \left(-\frac{1}{2} \dot M r + \dot y_d \tilde\theta - y_d \Gamma \dot y_{df}^T \varepsilon + \tilde N_1 + N_{1B} - (k_1+1) r - \beta_1 \, sgn(e_2) - e_2 \right)$$
$$+ e_1^T (e_2 - \alpha_1 e_1) + e_2^T (r - \alpha_2 e_2) + \varepsilon^T \left[\dot y_{df} \tilde\theta - y_{df} \Gamma \dot y_{df} \varepsilon + \tilde N_2 + N_{2B} - k_2 \varepsilon - \beta_2 \, sgn(\varepsilon) \right.$$
$$- r^T \left[N_{1B} - \beta_1 \, sgn(e_2) - \varepsilon^T (N_{2B} - \beta_2 \, sgn(\varepsilon)) \right] - \tilde\theta^T \Gamma^{-1} \left(\Gamma \dot y_d^T r + \Gamma \dot y_{df}^T \varepsilon \right)$$
$$\dot V = e_1^T e_2 - \alpha_1 e_1^T e_1 - \alpha_2 e_2^T e_2 - r^T y_d \Gamma \dot y_{df}^T \varepsilon + r^T \tilde N_1 - (k_1+1) r^T r - \varepsilon^T y_{df} \Gamma \dot y_{df} \varepsilon$$
$$+ \varepsilon^T \tilde N_2 - k_2 \varepsilon^T \varepsilon$$
where $-r^T y_d \Gamma \dot y_{df}^T \varepsilon \le c_1 \|r\| \|\varepsilon\|$

$$-\varepsilon^T y_{df} \Gamma \dot y_{df}^T \varepsilon \le c_2 \|\varepsilon\|^2$$
$$\dot V \le -\left(\alpha_1 - \frac{1}{2}\right) \|e_1\|^2 - \left(\alpha_2 - \frac{1}{2}\right) \|e_2\|^2 - \|r\|^2 + c_1 \|r\| \|\varepsilon\| + \|r\| \rho_1 \|z\| - k_1 \|r\|^2$$
$$+ c_2 \|\varepsilon\|^2 + \|\varepsilon\| \rho_2 \|z\| - k_{2a} \|\varepsilon\|^2 - k_{2b} \|\varepsilon\|^2$$
where $k_2 = k_{2a} + k_{2b}$

$$\|\tilde N_1\| \le \rho_1(\|z\|) \|z\|$$
$$\|\tilde N_2\| \le \rho_2(\|z\|) \|z\|$$
$$\dot V \le -\lambda_3 \|z\|^2 - k_{2b} \|\varepsilon\|^2 - \underbrace{\left[k_1 \|r\|^2 - \rho_1 \|r\| \|z\| \right]}_{\le \frac{\rho_1^2(\|z\|)(\|z\|)^2}{4k_1}} - \underbrace{\left[(k_{2a} - c_2) \|\varepsilon\|^2 - (\rho_2 + c_1) \|\varepsilon\| \|z\| \right]}_{\le \frac{(\rho_2(\|z\|) + c_1)^2 \|z\|^2}{4(k_{2a} - c_2)}}$$
$$\dot V \le -\lambda_3 \|z\|^2 - k_{2b} \|\varepsilon\|^2 + \frac{\max(k_{2a} - c_2, k_1)}{k_1 (k_{2a} - c_2)} \left[\frac{\rho_1^2 + (\rho_2 + c_2)^2}{4} \right] \|z\|^2$$
$$\le -\left[4 \lambda_3 k - \rho^2(\|z\|) \right] \frac{\|z\|^2}{4k} - k_{2b} \|\varepsilon\|^2$$
$$\lambda_3 = \min\left(\alpha_1 - \frac{1}{2}, \alpha_2 - \frac{1}{2}, 1 \right)$$
$$\rho^2(\|z\|) = \rho_1^2(\|z\|) + \left(\rho_2(\|z\|) + c_2 \right)^2$$
$$k = \min(k_1, k_{2a} - c_2)$$

Choose $4\lambda_3 k \geq \rho^2(\|z\|)$

$\Rightarrow \dot{V} \leq 0$, Lyapunov function is positive definite, radially unbounded,

$\dot{V}$ = Negative Semidefinite.

$\Rightarrow q_d, \dot{q}_d, \ddot{q}_d \in L_\infty$

$\Rightarrow V(r, e_1, e_2, \varepsilon, P_1, P_2, \tilde{\theta}) \in L_\infty \Rightarrow r, e_1, e_2, \varepsilon, P_1, P_2, \tilde{\theta} \in L_\infty$

$\Rightarrow \dot{e}_1, \dot{e}_2, q, \theta, \dot{\tilde{\theta}}, q_f, \hat{q}_f \in L_\infty \Rightarrow \dot{q}, \ddot{q}, u_1, u_2, \dot{u}_1, \dot{u}_2 \in L_\infty$

$\Rightarrow \dot{r} \in L_\infty \Rightarrow M(q), \dot{S} \in L_\infty, \Rightarrow M(q)\dot{r} \in L_\infty$

$\Rightarrow \dot{r} \in L_\infty$

Since $\dot{e}_1, \dot{e}_2, \dot{r}$ are uniformly continuous and $e_1, e_2, r \in L_2$,

$$\lim_{t\to\infty} r(t) = 0$$

$$\lim_{t\to\infty} e_1(t) = 0$$

$$\lim_{t\to\infty} e_2(t) = 0$$

By Barbalat's lemma, the result is semi-global asymptotic tracking.