

---

# Quintic polynomial trajectory

```
syms a0 a1 a2 a3
syms b0 b1 b2 b3

% Initial and Final velocity
v0=5;
vt=0;

%Initial State vector values
x0=10;
y0=20;
th0=0.5;

%Final State vector values
xt=0;
yt=0;
tht=0;

%Initial and Final Values
t0=0;
tf=10;

% Trajectory Initial
xd_0=a1+2*a2*t0+3*a3*t0^2;
yd_0=b1+2*b2*t0+3*b3*t0^2;
%thd_0=w

xd_t=a1+2*a2*tf+3*a3*tf^2;
yd_t=b1+2*b2*tf+3*b3*tf^2;
%thd_t=w

% Solving the equations
eqns=[x0-a0,xt-a0-a1*tf-a2*tf^2-a3*tf^3,y0-b0,yt-b0-b1*tf-b2*tf^2-
b3*tf^3,xd_0*cos(th0)+yd_0*sin(th0)-v0,xd_t*cos(tht)+yd_t*sin(tht)-
vt,xd_0*sin(th0)-yd_0*cos(th0),xd_t*sin(tht)-yd_t*cos(tht)];
vars = [a0 a1 a2 a3 b0 b1 b2 b3];
sol = solve(eqns, vars);

% Extracting the coefficients
sola=double([sol.a0 sol.a1 sol.a2 sol.a3]);
solb=double([sol.b0 sol.b1 sol.b2 sol.b3]);

% Defining Time interval
t = linspace(t0,tf,100);

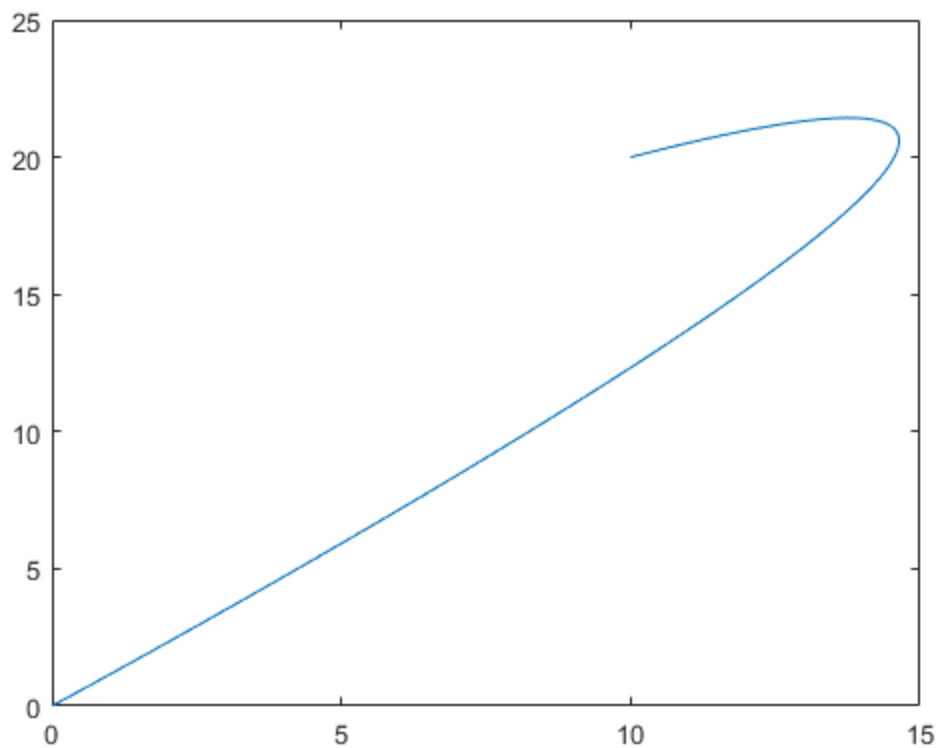
% Calculating the x and y trajectory
x=sola(1)+sola(2)*t+sola(3)*t.^2+sola(4)*t.^3;
y=solb(1)+solb(2)*t+solb(3)*t.^2+solb(4)*t.^3;
```

---

```
figure
plot(x,y);
hold on;

% Starting with initial error
z1=x0+2;
z2=y0+2;
z3=10;
z4=10;

% Store it in a single vector
z0=[z1;z2;z3;z4];
```



## Implement the trajectory tracking with ODE function

```
x0=[z1;z2;z3;z4];
options = odeset('RelTol',1e-4,'AbsTol',[1e-4, 1e-4, 1e-4, 1e-4]);

% Trajectory tracking using linear system
[T1,X_L] = ode45(@(t,x) DublinTrajectoryTracking(t,x,[sola;solb]),[t0
tf],z0, options);

% Trajectory tracking using augmented state vector
```

---

```
[T,X_NL] = ode45(@(t,x) DublinTrajectoryTracking_nonlinear(t,x,
[sola;solb]),[t0 tf],z0, options);

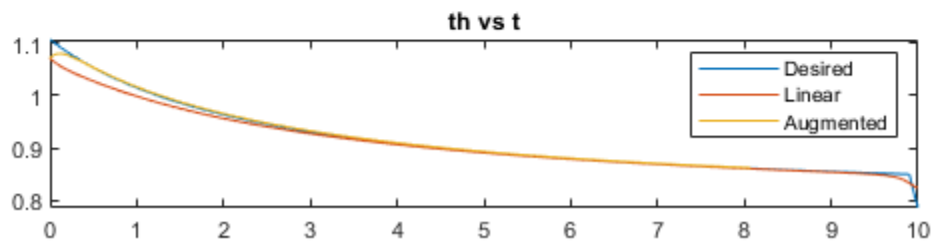
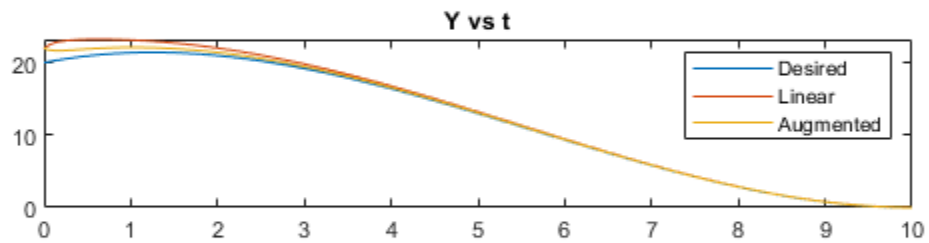
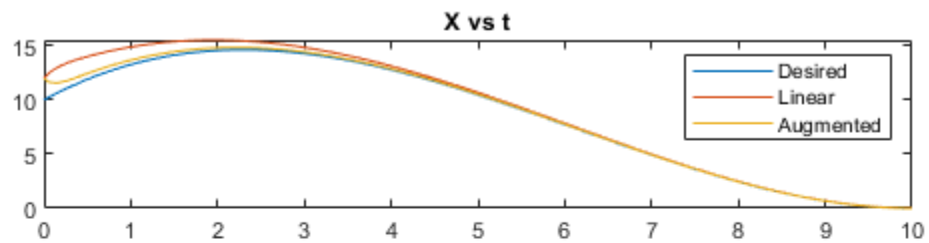
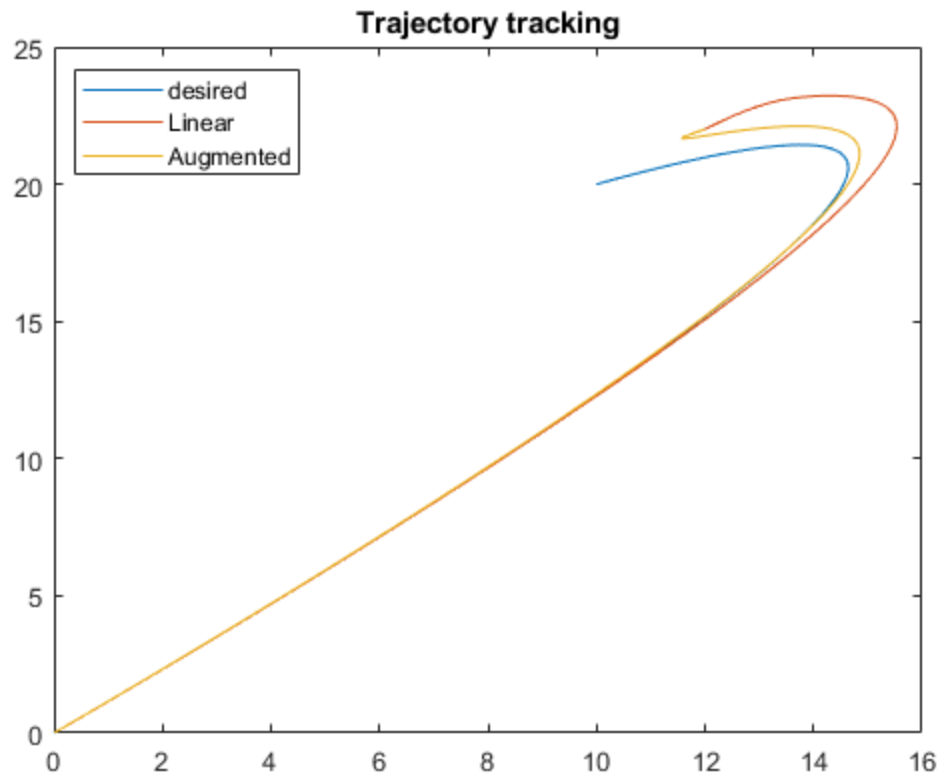
%Plotting the results of linear trajectory tracing controller
plot(X_L(:,1),X_L(:,2));
hold on

%Plotting the result of augmented state vector trajectory tracking
plot(X_NL(:,1),X_NL(:,2));
legend({'desired', 'Linear', 'Augmented'}, 'Location', 'northwest');
title('Trajectory tracking')

%Plotting the x,y,z vs t
figure
subplot(3,1,1)
plot(t,x)
hold on
plot(T1,X_L(:,1))
hold on
plot(T,X_NL(:,1))
title('X vs t')
legend('Desired', 'Linear', 'Augmented');

subplot(3,1,2)
plot(t,y)
hold on
plot(T1,X_L(:,2))
hold on
plot(T,X_NL(:,2))
title('Y vs t')
legend('Desired', 'Linear', 'Augmented');

subplot(3,1,3)
plot(t,atan2(y,x))
hold on
plot(T1,atan2(X_L(:,2),X_L(:,1)))
hold on
plot(T(1:250,1),atan2(X_NL(1:250,2),X_NL(1:250,1)))
title('th vs t')
legend('Desired', 'Linear', 'Augmented');
```



---

*Published with MATLAB® R2018b*