







By perturbing the endpoint around the arrived at by the unspecified final position and freeing up the control from the trajectory, we see that the original problem, as solved, obtains an optimal, minimum solution. Though it is clear from the plant and cost functions that a stable, minimum cost solution lies at *x* = 0 and *u* = 0, this point can not be reached is this time span with an overall cost less than that of the optimal solution. By extending the time interval, we see that the system converges to this state, and that the solution obtained from the original interval is the first part of the solution obtained.

***Code for Solving, Plots and Cost Calculation***

function sol = NumSolveCode(Start,Stop,Guess)

solinit = bvpinit(linspace(Start,Stop,Stop-Start+1),Guess);

sol = bvp4c(@OdeFunc, @BCFunc, solinit);

x = linspace(Start,Stop);

CostFunc = inline('.5\*z(1,:).^2 + .5\*z(2,:).^2', 'z');

Cost = (Stop / length(x) \* CostFunc(y));

for Index = 2:length(Cost)

Cost(Index) = Cost(Index) + Cost(Index-1);

end

Cost(end) = Cost(end) + .5\*y(1,end)^2;

function dxdy = OdeFunc(~,y)

dxdy = [-y(1)^3 - y(2);...

3\*y(2)\*y(1)^2 - y(1)];

function res = BCFunc(ya, yb)

res = [ya(1) - 0.5;...

yb(1) - yb(2)];