In this laboratory, your group will develop four subroutines in the C language. The file names and contents are:

ctrx1.h : x axis position control using proportional (P) control
ctrx2.h : x axis position control using controller of your own choice
ctry1.h : y axis position control using proportional (P) control
ctry2.h : y axis position control using controller of your own choice

The subroutine declarations are:

```c
int = AxisCtrX1(double Xr, double X, double Vx);
int = AxisCtrX2(double Xr, double X, double Vx);
int = AxisCtrY1(double Yr, double Y, double Vy);
int = AxisCtrY2(double Yr, double Y, double Vy);
```

The closed-loop system time constant should be at least 20% less than the open-loop time constant and the steady-state error should be within 20%. A skeleton of a file is given below.

Inputs: reference axis position (Xr and Yr; double precision; meter), measured axis position (X and Y; double precision; meter), and axis velocity (Vx and Vy; double precision; meter/second).

Note the axis velocity does not have to be used. Axis positions are absolute positions and are referenced to the machine tool position when the program is started. Output: digital value to output to D/A converter (integer) that is in the range -128 to 127. The controllers must include saturation and quantization effects (i.e., the output must be within the specified range and the output must be an integer). Data that will be saved: digital output, reference axis position (meter), measured axis position (meter), and axis velocity (meter/second).

Use the Robotool axis models developed in Assignment 1 to design the controllers. Turn in the four subroutines (make sure they are well documented), the detailed controller designs, the expected performances via simulation, and the actual performances. Discuss your results.
/ctrx1.h
int = AxisCtrX1(double Xr, double X, double Vx);

int = AxisCtrX1(double Xr, double X, double Vx)
{
int a;
double b;
static double c;
return a;
}

A double slash (//) will comment the line. Declaring a variable as static will allow it to remain intact when the subroutine ends while a non-static variable will be lost.

The x axis and y axis proportional controllers are designed such that the steady-state axial errors for a commanded velocity of 15 mm/s is 1.0 mm. From equation (3.14) in the Servomechanism Control Handout, the required proportional gains for the x and y axis controllers, respectively, are

\[
K_{p_x} = \frac{V_x}{K_{x e_{ss}}} = \frac{0.015}{0.0002543} \frac{m}{(m/s) \ digital \ number} \times \frac{0.001m}{0.001} = 58985 \ \frac{digital \ number}{m}
\]

(1)

\[
K_{p_y} = \frac{V_y}{K_{y e_{ss}}} = \frac{0.015}{0.0002284} \frac{m}{(m/s) \ digital \ number} \times \frac{0.001m}{0.001} = 65674 \ \frac{digital \ number}{m}
\]

(2)

The simulations are given below followed by the experiments.
Simulink Block Diagram for Proportional Controllers

X Axis Simulations

Y Axis Simulations
Y Axis Simulations
M File for Simulations

% lab2file.m

% This file simulates the Robotool x and y axis controller for
% linear contours for Laboratory #2 for ME584 Winter 2000 semester.

% Last Update: March 25, 2000 by Robert G. Landere

vh = 0.015; % high reference velocity (m/s)
v1 = 0.005; % low reference velocity (m/s)

vx = 0.015; % x axis reference velocity (m/s) 
vy = 0.015; % y axis reference velocity (m/s)

nex = 1e-5; % x axis encoder resolution (m)
ney = 1e-5; % y axis encoder resolution (m)

essx = 1e-3; % x axis desired steady state error (m)
issy = 1e-3; % y axis desired steady state error (m)

gx = 2.543e-4; % x axis DC gain ((m/s)/digital number)
tx = 0.04; % x axis time constant (s)
gy = 2.284e-4; % y axis DC gain ((m/s)/digital number)
ty = 0.045; % y axis time constant (s)

Kpx = vx/(gx*essx) % x axis proportional gain (digital number/m)
Kpy = vy/(gy*issy) % y axis proportional gain (digital number/m)

Xmax = 127; % x axis upper saturation limit (digital number)
Xmin = -128; % x axis lower saturation limit (digital number)
Ymax = 127; % y axis upper saturation limit (digital number)
Ymin = -128; % y axis lower saturation limit (digital number)

t1 = 1; t2 = 2; t3 = 3; t4 = 4; % time for each motion (s)

T = 0.01; % sample period (s)
Tf = t4; % final time (s)

for i=1:Tf/T+1, time(i,1)=(i-1)*T; end;
sim('lab2',Tf)
X Axis Experiments

Y Axis Experiments
**X Axis Controller Subroutine for Experiments**

// ctrx1.h


// This function is a proportional x-axis position controller
// for laboratory #2 ME584 Winter 2000 semester.

int AxisCtrX1(double Xr,double X,double V);

// Xr : reference position (m)
// X  : measured position (m)
// V  : measured axis velocity (m/s)

int AxisCtrX1(double Xr,double X,double V)
{
  double Ex; // current position error (m)
  double U; // current control value (digital number)

  static double Umax = 127;
  static double Umin = -128;

  static double Kp = 58.985; // proportional gain (digital number/mm)

  Ex = 1000.0*(Xr-X); // current position error (mm)

  // calculate control command
  U = Kp*Ex;

  // check and enforce saturation values
  if (U > Umax) { U = Umax; }
  if (U < Umin) { U = Umin; }

  // return control value
  return (int) (U);
}
Y Axis Controller Subroutine for Experiments

// ctry1.h

// This function is a proportional y axis position controller
// for laboratory #2 ME584 Winter 2000 semester.

int AxisCtrY1(double Xr,double X,double V);

// Xr : reference position (m)
// X  : measured position (m)
// V  : measured axis velocity (m/s)

int AxisCtrY1(double Xr,double X,double V)
{
    double Ex; // current position error (m)
    double U; // current control value (digital number)

    static double Umax = 127;
    static double Umin = -128;

    static double Kp = 100.0; // proportional gain (digital number/mm)

    Ex = 1000.0*(Xr-X); // current position error (mm)

    // calculate control command
    U = Kp*Ex;

    // check and enforce saturation values
    if (U > Umax) { U = Umax; }
    if (U < Umin) { U = Umin; }

    // return control value
    return (int) (U);
}