case 5:

int kP, kI, kD;

int prevError;

int Ictrl, Dctrl;

long int Pctrl;

long int errorSum = 0, maxErrorSum;

double dt = 0.001;

i = 0;

j = 0;

// message format: "case , target , kP , kI , kD, interference "

//control rate 1000Hz = 1ms = dt

//sample rate 100Hz = 10ms

sscanf(message + 2, "%d,%d,%d,%d, %d", &target, &kP, &kI, &kD, &interf);

maxErrorSum = 255 / (double)(kI \* dt); //calculate top boundary for error sum

start\_time = millis();

**Serial**.println(">Case 5 is running");

while ((millis() - start\_time) <= 2005)

{

if((millis() - start\_time) >= 1000) //light interference LED2 after 1 second

analogWrite(L2, interf);

if ((millis() - start\_time) >= i) //condition for control rate

{

error = target - analogRead(LDR); //max error is: target, min error is: 0

/\* P \*/

Pctrl = (long int)kP \* error;

/\* I \*/

errorSum += error;

if (errorSum >= maxErrorSum)

errorSum = maxErrorSum;

Ictrl = kI \* (dt \* errorSum);

/\* D \*/

Dctrl = -1\* kD \* (prevError - error);

prevError=error;

/\* PID duty \*/

duty = Pctrl + Ictrl + Dctrl;

if (duty > 255)

duty = 255; //full gas

if (duty < 0)

duty = 0; //OFF

analogWrite(L1, duty);

i++;

}

//Take sample from LDR

if ((millis() - start\_time) >= j \* 10) //sample to graph 100Hz

{

xArr[j] = millis() - start\_time;

yArr[j] = analogRead(LDR);

j++;

}

}

//Send data to CVI

for (j = 1; j < 201; j++)

{

sprintf(message, "%d,%d", xArr[j], yArr[j]);

**Serial**.println(message);

}

**Serial**.println("\*");

//shut down lights

digitalWrite(L1, LOW);

digitalWrite(L2, LOW);

state = 0;

**Serial**.println(">Returning to State 0");

break;

}

}

}

}