Control & Computing Laboratory (EE615) Report

Submitted by

R.V.Satwik (193079009)
Group number: 5

Department of Electrical Engineering,
Indian Institute of Technology, Bombay

Experiment No: 2

PID Control of Line Following Robot

AIM

To implement PID controller for position control in a line follower robot.

OBJECTIVE

Design and implement a line following algorithm in the Spark V robot. Additional condition imposed on the system is that the time of traversal of the track must be under 30 seconds.

CHALLENGES FACED

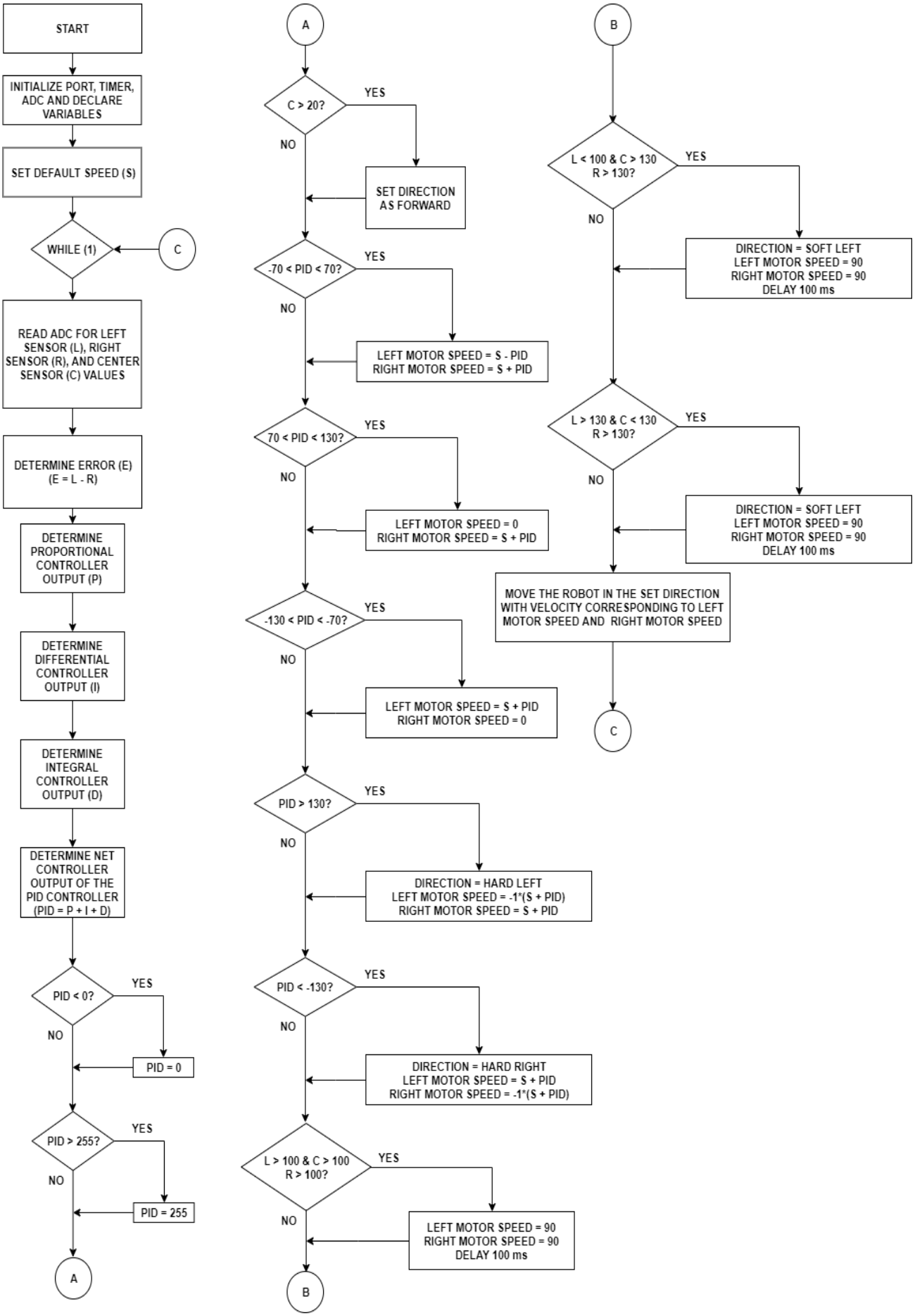
1. Tuning of the IR sensors: Three IR sensors are used to track a line in the Spark V robot. The three IR sensors have different dead zones and sensitivity to light. This presents a challenge to obtain values from the three sensors that are close to each other. Further it is important to make sure that the values do not drift apart too much for same input during the implementation of the algorithm. The three sensors are placed on a white strip so as to make sure that all the three sensors receive same input. The least value received, among the three sensors is noted. The remaining two sensors of the three are matched to this minimum value. If the matching cannot be done to the least value, the next highest value received, among the three sensors is noted. The remaining two sensors of the three are matched to this value. Keeping the values minimum ensures that the values do not drift apart too much for same input during the execution of the algorithm.

2. The PID algorithm fails when values are printed onto LCD: An LCD print instruction takes significant machine cycles for execution. This increases the amount of time taken for execution of a single loop of the algorithm. The values of the sensors and the error are determined only once at the beginning of the loop. This implies if the loop takes longer to execute then the sensor values and the errors are refreshed after more delay. This sometimes results in the robot completely missing the line before the next loop of the code begins execution, especially during the turns. Hence all print commands are avoided during tuning of the PID and in turn during the execution of the code.

3. Determining the default speed of the robot. The default speed of the robot is modified by the PID algorithm for differential drive. Due to the above reasons it is important that the default speed is not too high. The setting of the default speed can be done only by the trial and error method.

4. Determining the PID output ranges for soft and hard turns.

FLOW CHART


START
INITIALIZE PORT, TIMER, ADC AND DECLARE VARIABLES
SET DEFAULT SPEED (S)
WHILE (1)
C
READ ADC FOR LEFT SENSOR (L), RIGHT SENSOR (R), AND CENTER SENSOR (C) VALUES
DETERMINE ERROR (E) (E = L - R)
DETERMINE PROPORTIONAL CONTROLLER OUTPUT (P)
DETERMINE DIFFERENTIAL CONTROLLER OUTPUT (I)
DETERMINE INTEGRAL CONTROLLER OUTPUT (D)
DETERMINE NET CONTROLLER OUTPUT OF THE PID CONTROLLER (PID = P + I + D)
PID < 0?
YES
NO
PID = 0
PID > 255?
YES
NO
PID = 255
A
A
C > 20?
YES
NO
SET DIRECTION AS FORWARD
-70 < PID < 70?
YES
NO
LEFT MOTOR SPEED = S - PID
RIGHT MOTOR SPEED = S + PID
70 < PID < 130?
YES
NO
LEFT MOTOR SPEED = 0
RIGHT MOTOR SPEED = S + PID
-130 < PID < -70?
YES
NO
LEFT MOTOR SPEED = S + PID
RIGHT MOTOR SPEED = 0
PID > 130?
YES
NO
DIRECTION = HARD LEFT
LEFT MOTOR SPEED = -1*(S + PID)
RIGHT MOTOR SPEED = S + PID
PID < -130?
YES
NO
DIRECTION = HARD RIGHT
LEFT MOTOR SPEED = S + PID
RIGHT MOTOR SPEED = -1*(S + PID)
L > 100 & C > 100
R > 100?
YES
NO
LEFT MOTOR SPEED = 90
RIGHT MOTOR SPEED = 90
DELAY 100 ms
B
B
L < 100 & C > 130
R > 130?
YES
NO
DIRECTION = SOFT LEFT
LEFT MOTOR SPEED = 90
RIGHT MOTOR SPEED = 90
DELAY 100 ms
L > 130 & C < 130
R > 130?
YES
NO
DIRECTION = SOFT LEFT
LEFT MOTOR SPEED = 90
RIGHT MOTOR SPEED = 90
DELAY 100 ms
MOVE THE ROBOT IN THE SET DIRECTION WITH VELOCITY CORRESPONDING TO LEFT MOTOR SPEED AND RIGHT MOTOR SPEED
C

RESULT

$K_P = 2$

$K_I = 0$

$K_D = 4$

Time for traversal = 28 seconds

INFERENCE

The effect of proportional gain is tight adherence to the track. For very low proportional gain values the robot oscillates even when the line being tracked is straight. As the proportional gain increases, the small values of error are amplified and this results in reduced oscillations on straight track.

However large proportional gain results in oscillations at turns. This is because the proportional gain output calculated becomes very large at turns of the track. The differential action makes sure that the changes in the PID output are not too abrupt. The differential gain output is low during straight tracks. At the turnings the differential action jumps in and damps the oscillations produced due to proportional action. The integral action is not used as it easily saturates the PID output value.

Another important observation is that an LCD print instruction takes significant machine cycles for execution. This increases the amount of time taken for execution of a single loop of the algorithm. The values of the sensors and the error are determined only once at the beginning of the loop. This implies if the loop takes longer to execute then the sensor values and the errors are refreshed after more delay. This sometimes results in the robot completely missing the line before the next loop of the code begins execution, especially during the turns. Hence all print commands are avoided during tuning of the PID and in turn during the execution of the code.

CONTROL & COMPUTING LAB
RESULT SHEET

	Student Name	Roll No
1	R. V. Sahwik	193079009
2	Alen Rujis	193079008
3		

Group No -

$K_p = 2$

$K_I = 0$

$K_d = 4$

Track traced within 28 seconds.

	TA Name	Date	Signature
1	TARUN S.	2/10/2019	
2	Vatsal Kedia	2/10/2019	