# A progress report on mobile robot control

Prajwal S, Paritosh Kelkar, Santhosh Shetty, Siddharth

Project Guide - Mrs. Suma K V

Dept. of Electronics and Communication,
M. S. Ramaiah Institute of Technology

We have completed phase 1 of the project, wherein we achieved go-to-goal for a single robot (the ability to make the robot navigate to any desired co-ordinate position in an uncluttered environment). The report contains details of the model used and implementation.

## 1 Theory and implementation

We used wheel encoders to compute the distance travelled and orientation of the differentially-driven unicycle model robot. (Two wheels at the rear driven by 2 controlled motors and a caster wheel in the front) A wheel encoder counts the number of revolutions of the wheels of the robot in a fixed interval of time. (In practice, the time interval is small and the fraction of one revolution is what is measured.)

The wheels of the robot are stuck with alternate white and black strips. An IR sensor is used to count the number of transitions between the white and black strips. The IR sensor consists a transmitting LED, a receiving photo-diode and an Op-amp in open-loop configuration. When the IR sensor is "looking at"(pointed at) a white strip, the Op-amp output is at positive saturation and when the IR sensor is pointed at a black strip, the Op-amp output is at negative saturation (ground level). The Op-amp output is continuously read by the microcontroller which counts the fraction of revolution completed in the fixed time interval.

We have used 24 strips of white paper on the wheel interspersed with 24 strips of the natural black of the wheel. Hence, one revolution is 48 transitions between black and white. (The greater the number of strips, the greater is the resolution of the wheel encoder. However, divergence of the IR beam is a limiting factor in deciding the spacing between 2 white strips)

The number of revolutions made by the wheel in the fixed time interval is given by

$$N = \frac{Transitions}{48} \tag{1.1}$$

If R is the radius of the wheel, the linear distance D traveled by the wheel is

$$D = 2\pi RN \tag{1.2}$$

The distance traveled by the robot (not to be confused with the distance traveled by each wheel) in

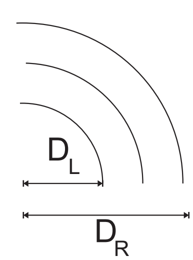


Figure 1: Difference in distance travelled by left wheel and right wheel when the robot turns

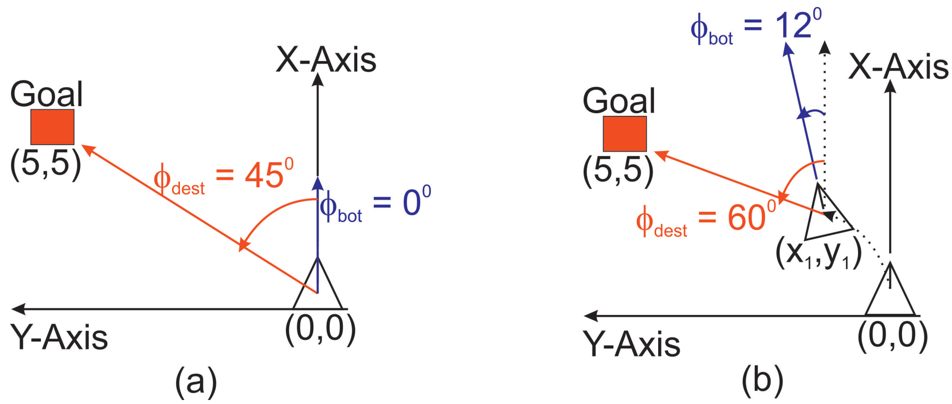


Figure 2: Movement of robot toward a set Destination

the fixed time is found by calculating the mean of distance traveled by the left and right wheels (Fig. 1). If $D_L$ is the distance traveled by the left wheel and $D_R$ is the distance traveled by the right wheel, the distance traveled by the robot is

$$D_{bot} = \frac{D_L + D_R}{2} \tag{1.3}$$

The change in orientation (angular displacement) of the robot is proportional to the difference in speeds of the left and right wheels (Fig. 1). Knowing wheel radius R and the separation between the wheels L the angular displacement ($d\phi_{bot}$) can be found as -

$$d\phi_{bot} = \frac{D_R - D_L}{L} \tag{1.4}$$

Let the floor be the co-ordinate plane and the desired position for the robot be (h,k) and it's current position be (x,y) and it's orientation be $\phi_{bot}$ with respect to the positive X-axis. The initial position of the robot is considered to be the origin and the direction it faces considered to be along the X-axis. Let the the line joining the robot and destination make an angle $\phi_{dest}$ with respect to the positive X-axis. $\phi_{error}$ is the angle between the current orientation of the robot and the desired orientation towards the destination, (i.e error in orientation),

$$\phi_{error} = \phi_{dest} - \phi_{bot} \tag{1.5}$$

The control action taken on the robot by acting on the motors is governed by the following two equations

$$\mathrm{V}_R = \frac{2 \times \mathrm{V} + (\phi_{error} \ \times \mathrm{L})}{2R} \tag{1.6}$$

$$\mathrm{V}_L = \frac{2 \times \mathrm{V} \text{ - } (\phi_{error} \times \mathrm{L})}{2R} \tag{1.7}$$

Where V is the average velocity of the two wheels (velocity of the robot). V is kept constant so that there are only 2 unknowns ($V_R$ and $V_L$) for which we can solve. These are the inputs to the right and left motors respectively.

Based on inputs to the motors, in the fixed time interval, the robot moves to a new position and changes its orientation as well. (Fig. 2) Using the measurements from the wheel encoders, the new position and orientation is tracked as follows

$$x = x + \mathrm{D}_{bot} \cos(\phi_{bot}) \tag{1.8}$$

$$y = y + \mathrm{D}_{bot} \sin(\phi_{bot}) \tag{1.9}$$

$$\phi_{bot} = \phi_{bot} + \frac{D_R - D_L}{L} \tag{1.10}$$

In summary, based on the error in orientation ($\phi_{error}$), the input to the motors is continuously updated till the robot reaches the destination. It is to be noted that $\phi_{dest}$ also continuously changes and not just $\phi_{bot}$.

## 2 Problems encountered

1. Since the transmitting LED of the IR sensor emits a divergent beam and receiver photodiode of the IR sensor has a wide aperture, there is a high possibility of two adjacent white strips being interpreted as a single white strip with the black strip between them being skipped Fig. 3. In such a scenario, there is a miscount of 2 transitions (failure to count 1 white to black and 1 black to white transition). To overcome this difficulty, careful tuning of the potentiometer which controls the Op-amp reference signal was employed. Also, by experimentation an appropriate strip width for the white paper strips was found.

2. The output of the sensor is constantly about 2V lesser than the Vcc of the sensor's op amp. Hence, when a regulated 5V Vcc was used, the output of the sensor sometimes fell below 3V (which is required by the micro-controller to recognize the signal as a HIGH). As a result, the micro-controller would sometimes misinterpret the signal as logic LOW when it was actually logic HIGH. This problem was especially significant when the battery not only powered the sensor but the motors as well. This led to us looking at the following options to overcome the specified problem:

   (a) Using the unregulated 12V directly from the Lead Acid battery and a potential divider circuit between the micro-controller and the sensor to supply the requisite 5V to the microcontroller. However, the current flowing through the IR transmitter LED and the indicator led was around 30mA which is potentially harmful to the device.

   (b) Using a separate set of 4 AA batteries (yielding 1.5V×4=6V) to power the sensors alone. Keeping in mind that more sensors would be required to accomplish future tasks such as obstacle avoidance, this option is the logical choice.

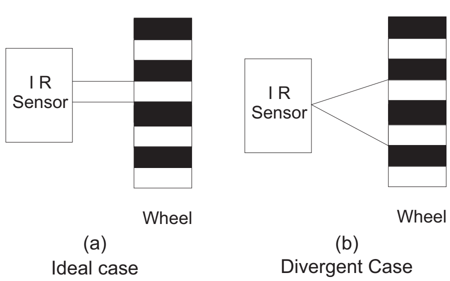


Figure 3: Two adjacent white strips interpreted as a single white strip resulting in error in the non-ideal (actual) case

## 3 Particular Logical Scenarios

This section specifically explains two interesting cases in the movement of the robot that we had not foreseen.

### 3.1 Case 1: The About Turn

**What was happening**:
The robot could not execute an about turn (180°) in order to get to the destination behind itself. It would start making the turn and would then suddenly turn the opposite direction and the process repeated. The process is illustrated in Fig. 4.

**Why it was happening:**
$\phi_{error}$ = $\phi_{dest}$ - $\phi_{bot}$ is what is fed to the robot that makes it turn in particular directions. A positive error implied it should make a counter clockwise turn. In this particular case (notice in Fig. 4 the position of the robot at different instances), the error in angle alternately changed signs and the absolute difference between consecutive errors was always very large. This caused the robot to rapidly turn in opposing directions.

**What is the solution:**
This was tackled by providing a case within the code that recognised the drastic error changes and forced the error to retain its sign thereby making the robot turn only in one direction.

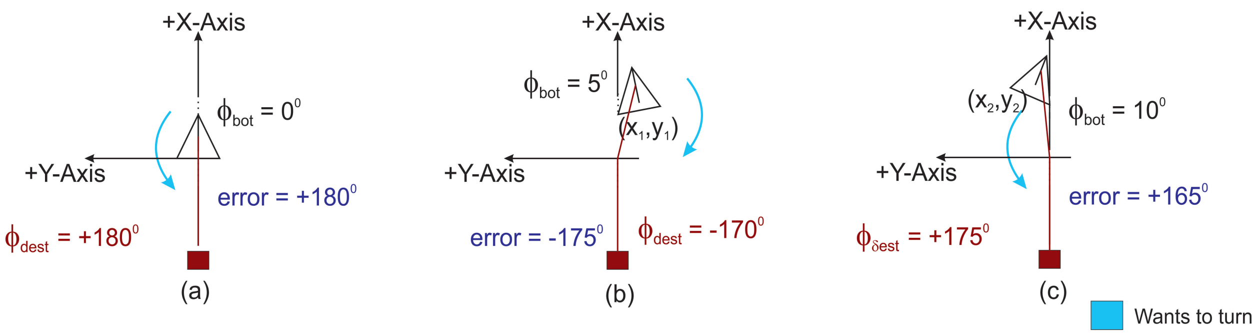


Figure 4: Illustration of the problem of the robot executing opposing turns

### 3.2 Case 2: Taking the larger turn instead of the shorter one

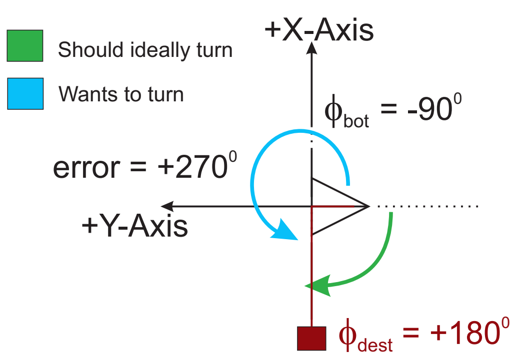


Figure 5: The robot wanting to take the longer turn instead of the shorter one

**What was happening**: Assuming the robot to be in the position shown in fig, instead of making the shorter clockwise 90° turn, it would turn a full 270° counter clockwise to orient itself towards the destination Fig. 5.

**Why it was happening:** As the figure illustrates, the error that is calculated by the robot is +270°. This positive error is making the robot turn counter clockwise. It can be discerned that, this scenario could arise for any $\phi_{error}$ greater than +180° and less than -180°

**What is the solution:** The solution here was to simply limit the error to between -180 & +180°. Hence, an error of 270° would translate into (270-360)°, i.e , -90°. The negative sign forces the robot to turn clockwise as is required.

## 4 Next phase of the project

The focus will now be to implement obstacle avoidance and integrate it with go-to-goal behaviour for navigation in a cluttered environment.

## References

[1] Magnus Egerstedt, *Control of Mobile Robots*, Georgia Institute of Technology, Coursera (video lectures).