McGill University
Department of Electrical and
Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

Navigation

ECSE 211: Design Principles and Methods

McGill University
Department of Electrical and
Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

Overview

- Introduction
- Derivation
- Example
- Derivation (cont'd)
- Block Diagram
- Summary

Introduction

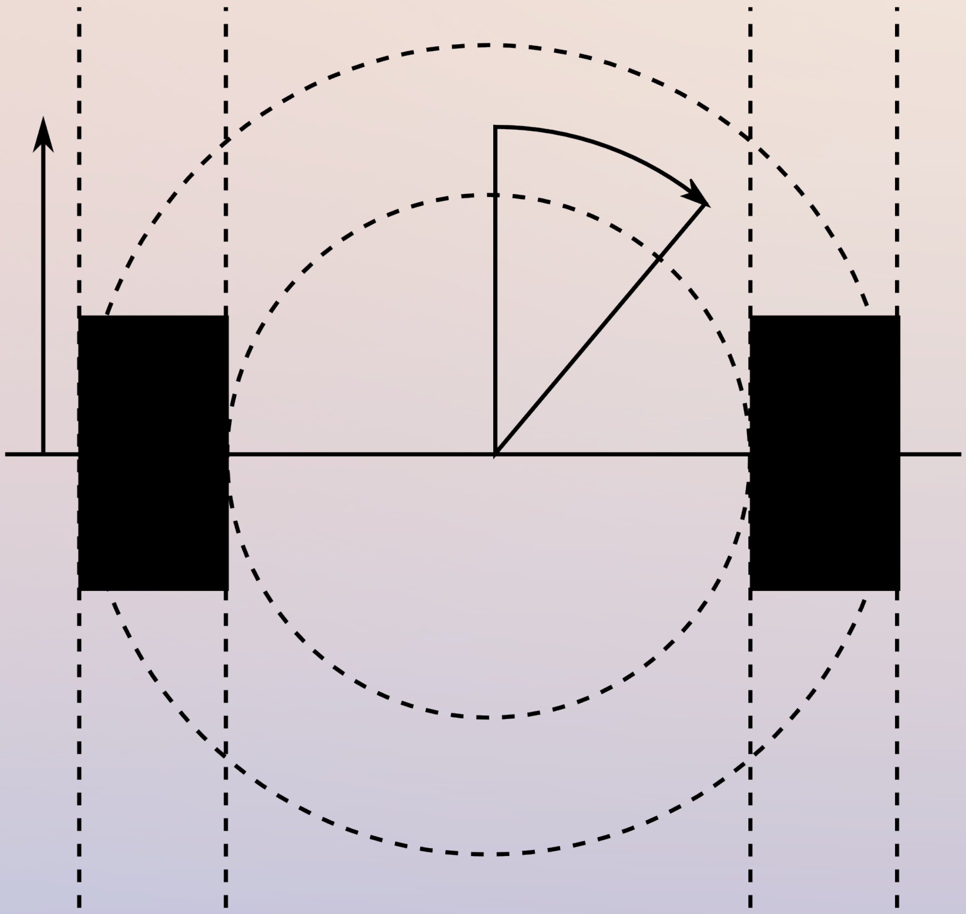
McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- With an accurate odometer, it is now possible to position the robot in the field
- However, no means has been established for moving the robot from one point to another in the field
- Using the same principles as those used to derive the algorithm for odometry, a means of driving the robot to a specific position on the field will be developed

Derivation

McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- The robot's motion is broken into two motions, forward and rotary, that are assumed to be independent



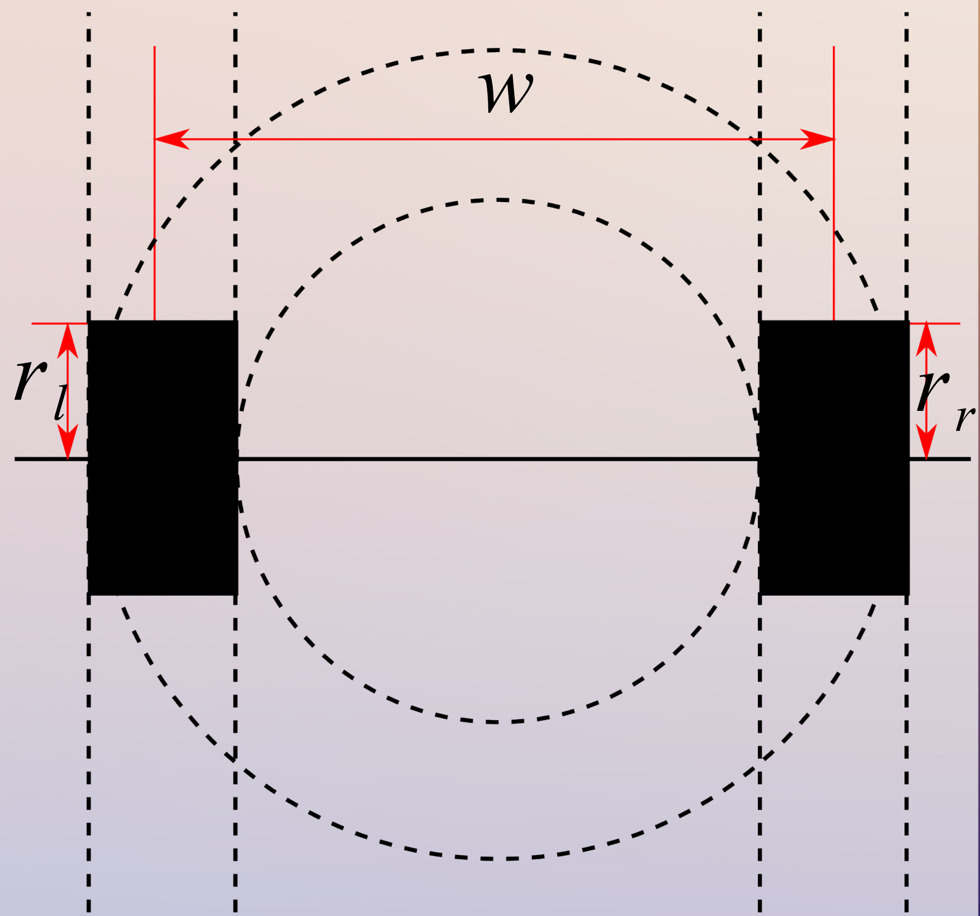
McGill University
Department of Electrical and
Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

Derivation

- As with odometry, the robot's wheel radii and its width are defined
- If the robot is moving forward at a velocity v, then it holds that:

$$\omega_l = \frac{v}{r_l}, \quad \omega_r = \frac{v}{r_r}$$

where ω_l and ω_r are the angular velocities of the left and right wheels, respectively.



Derivation

McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- Similarly, if the robot is only rotating clockwise on point at a speed ω, then it holds that:

$$\omega_l = \frac{w}{2r_l}\omega, \quad \omega_r = \frac{-w}{2r_r}\omega$$

- Since the value $r_l\omega_l + r_r\omega_r$ is 0 when the robot is rotating, and $r_l\omega_l - r_r\omega_r$ is 0 when the robot is moving forward, we solve for v and ω in terms of $r_l\omega_l + r_r\omega_r$ and $r_l\omega_l - r_r\omega_r$:

$$v = \frac{r_l\omega_l + r_r\omega_r}{2}, \quad \omega = \frac{r_l\omega_l - r_r\omega_r}{w}$$

Example

McGill University
Department of Electrical and
Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- The robot's right wheel is travelling in an arc of radius 1 m. If the robot is 15 cm wide (wheel center to wheel center), and its forward velocity is 10 cm/s, determine its angular velocity, assuming the left wheel is rotating faster than the right wheel.

  Solution: The robot is travelling in a circle of radius 1.075 m. Since it is travelling at a forward velocity of 0.1 m / s, it will complete 1 rad of this circle in 10.75 s. Its angular velocity is thus:

$$\omega = \frac{1}{10.75} \approx 0.093\ \mathrm{rad/s}$$

McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

Example

- If we assume wheel radii of 2.8 cm, we can then compute the speed of the robot's wheels:

$$\omega_l = \frac{v}{r_l} + w\frac{\omega}{2r_l} \approx 3.82\,\mathrm{rad/s}$$

$$\omega_r = \frac{v}{r_r} - w\frac{\omega}{2r_r} \approx 3.32\,\mathrm{rad/s}$$

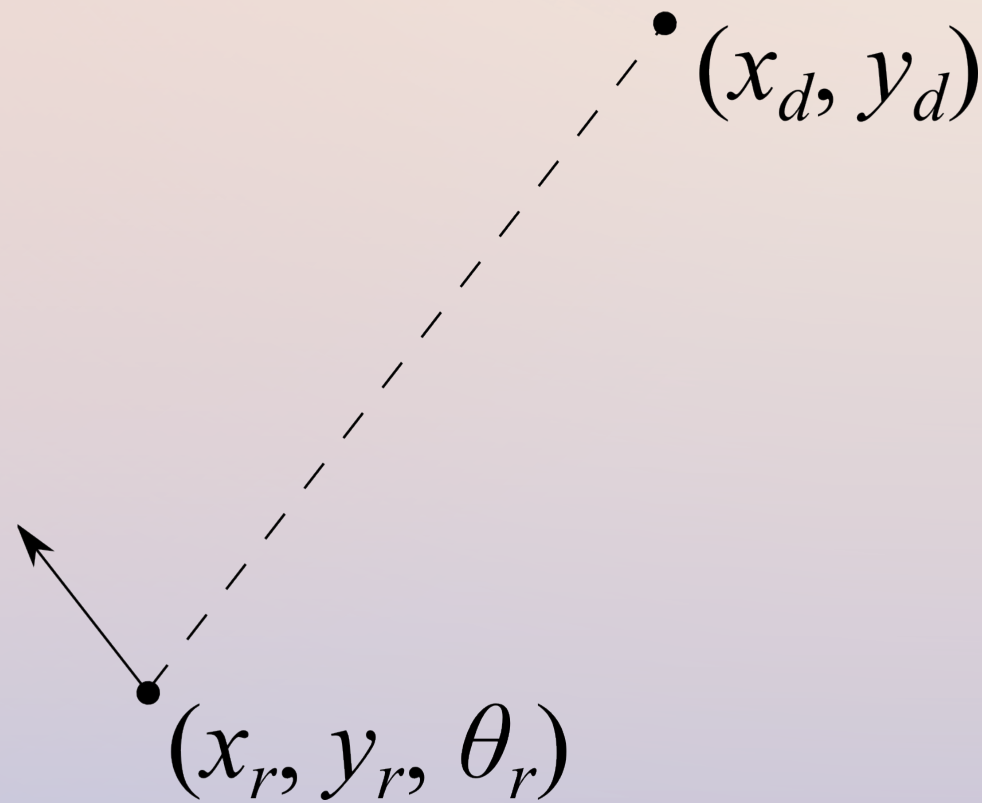
or 219 and 190 degrees per second, respectively.

McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

Derivation

- Now let us assume that we want to move the robot from its current position, (x_r, y_r, θ), to a new position (x_d, y_d)
- The robot clearly must rotate to face its destination, which is at a heading of:

$$\theta_d = \tan^{-1}(y_d - y_r, x_d - x_r)$$



Derivation

McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- The $\tan^{-1}(y, x)$ function is a binary form of the unary arctan function, which, unlike its unary cousin, provides the correct result when x is negative:

$$\tan^{-1}(y,x)=\begin{cases}\tan^{-1}\left(\dfrac{y}{x}\right) & x>0\\ \tan^{-1}\left(\dfrac{y}{x}\right)+\pi & x<0,\ y>0\\ \tan^{-1}\left(\dfrac{y}{x}\right)-\pi & x<0,\ y<0\end{cases}$$

Derivation

McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- The *error* in the robot's heading is $\theta_d - \theta_r$, however this value may not represent the minimal rotation of the robot (e.g. the robot may rotate 359º clockwise instead of 1º counter-clockwise)
- To correct this flaw, consider possible values of $\theta_d - \theta_r$
  - Assume θ_d and θ_r are each angles between 0 and 359
  - Then $\theta_d - \theta_r$ cannot be smaller than -359, nor larger than +359
  - Since the interval (-180, 180) will result in minimal rotation, only the intervals (-359, -180) and (180, 359) need consideration

Derivation

McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- When $\theta_d - \theta_r$ is less than -180, the robot will attempt to rotate counter-clockwise by more than 180º, when it should instead rotate clockwise by $360^\circ + (\theta_d - \theta_r)$
- When $\theta_d - \theta_r$ is greater than 180, the robot will attempt to rotate clockwise by more than 180º, when it should instead rotate counter-clockwise by $(\theta_d - \theta_r) - 360^\circ$
- Thus we will define (see next slide) a function $f(x)$ such that:

$$f(\theta_d - \theta_r) \in (-180, 180)$$

Derivation

McGill University
Department of Electrical and
Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- As is seen below, a simple piecewise function will correctly round the error in the heading of the robot to the interval (-180º, 180º)

$$f(\theta_d-\theta_r)=\begin{cases}\theta_d-\theta_r & (\theta_d-\theta_r)\in(-180,180)\\(\theta_d-\theta_r)+360 & (\theta_d-\theta_r)<-180\\(\theta_d-\theta_r)-360 & (\theta_d-\theta_r)>180\end{cases}$$

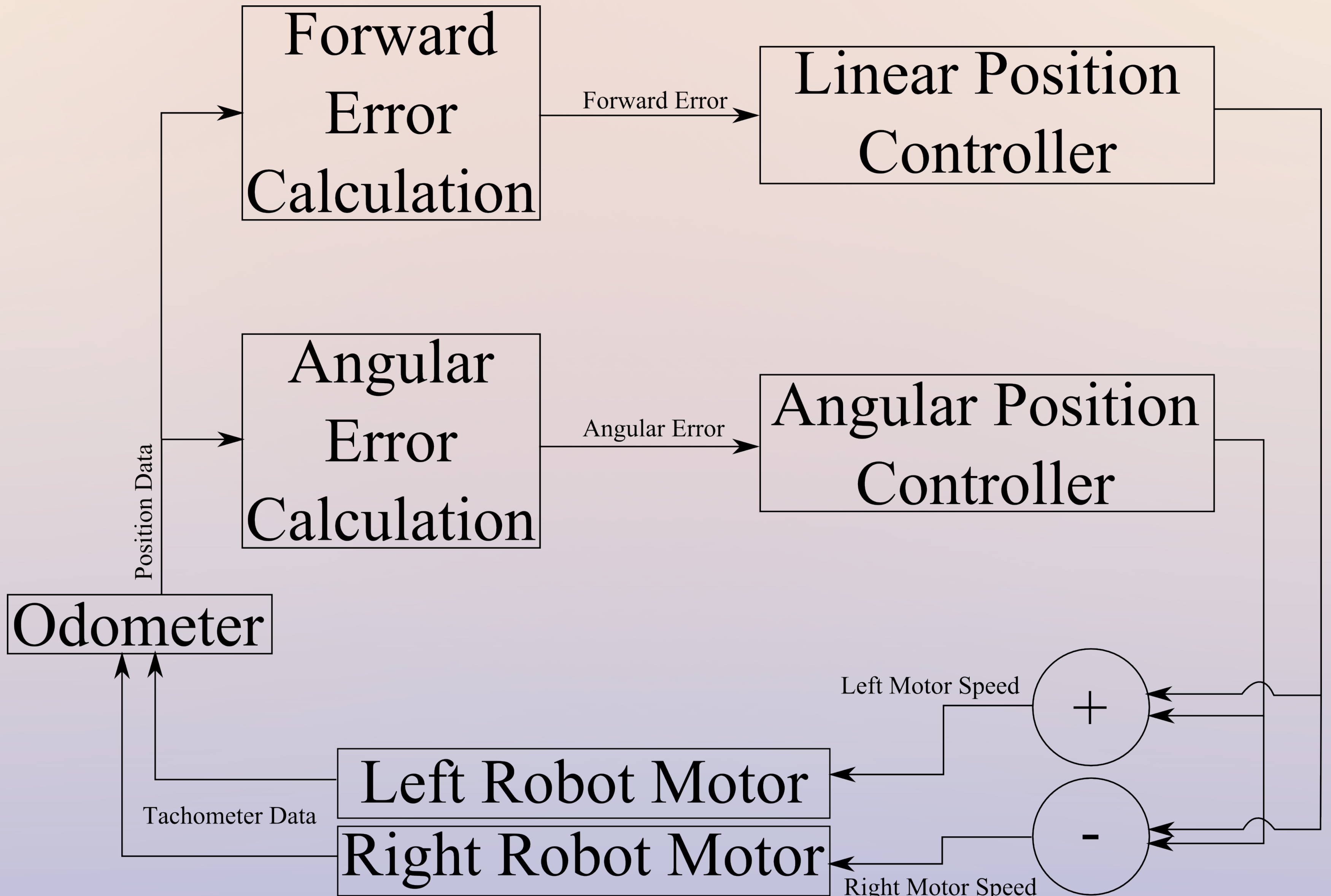
Derivation

McGill University
Department of Electrical and
Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- For the forward position error, several options exist
  - Euclidean distance from the robot to the destination: Easily calculated, but will only work for small angular error.
  - Dot product of vector connecting robot to destination and robot's unit direction vector: Harder to calculate, but will work for large angular error
- Once the forward and angular errors are computed, a *controller* uses these values to determine the speeds of the motors (see block diagram on next slide)

Block Diagram

McGill University
Department of Electrical and
Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate



Summary

McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- The robot chooses a (possibly hard-coded) destination point on the field
- This point's coordinates are used in the forward and angular error calculations to determine how much the robot needs to advance or turn
- The controllers then compute the linear and angular *velocities* which are added and subtracted for the left and right wheel motors respectively
- This in turn causes the motors to spin, changing the robot's coordinates, and causing new values to be calculated when this process start all over