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

Odometry

ECSE 211: Design Principles and Methods

Outline

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

- Introduction
- Assumptions
- Derivation
- Error
- Summary

Introduction

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

- Requires the synthesis of many basic topics
  - Geometry
  - Trigonometry
  - Differential Calculus
- Grounded in theory
- Essential for the operation of wheeled robots

Assumptions

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

- The radius and distance-from-robot-centre-point of each wheel are *constant*, and the wheels are parallel.
- The tires contact the floor surface at a point and do not slip.
- Over a *sufficiently* short time interval, the angular velocity of the wheels is *constant*.
- The NXT brick is capable of sampling the wheel motor tachometers *sufficiently* quickly.

Derivation – Overview

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

1. The type of motion the robot will perform over a *sufficiently short* time interval must be determined
2. The change in the robot's position over said interval must be expressed as a function of what is known: the change in its wheel motors' tachometers
3. A recursive relationship will yield the computation of its position (relative to its starting point) for all time

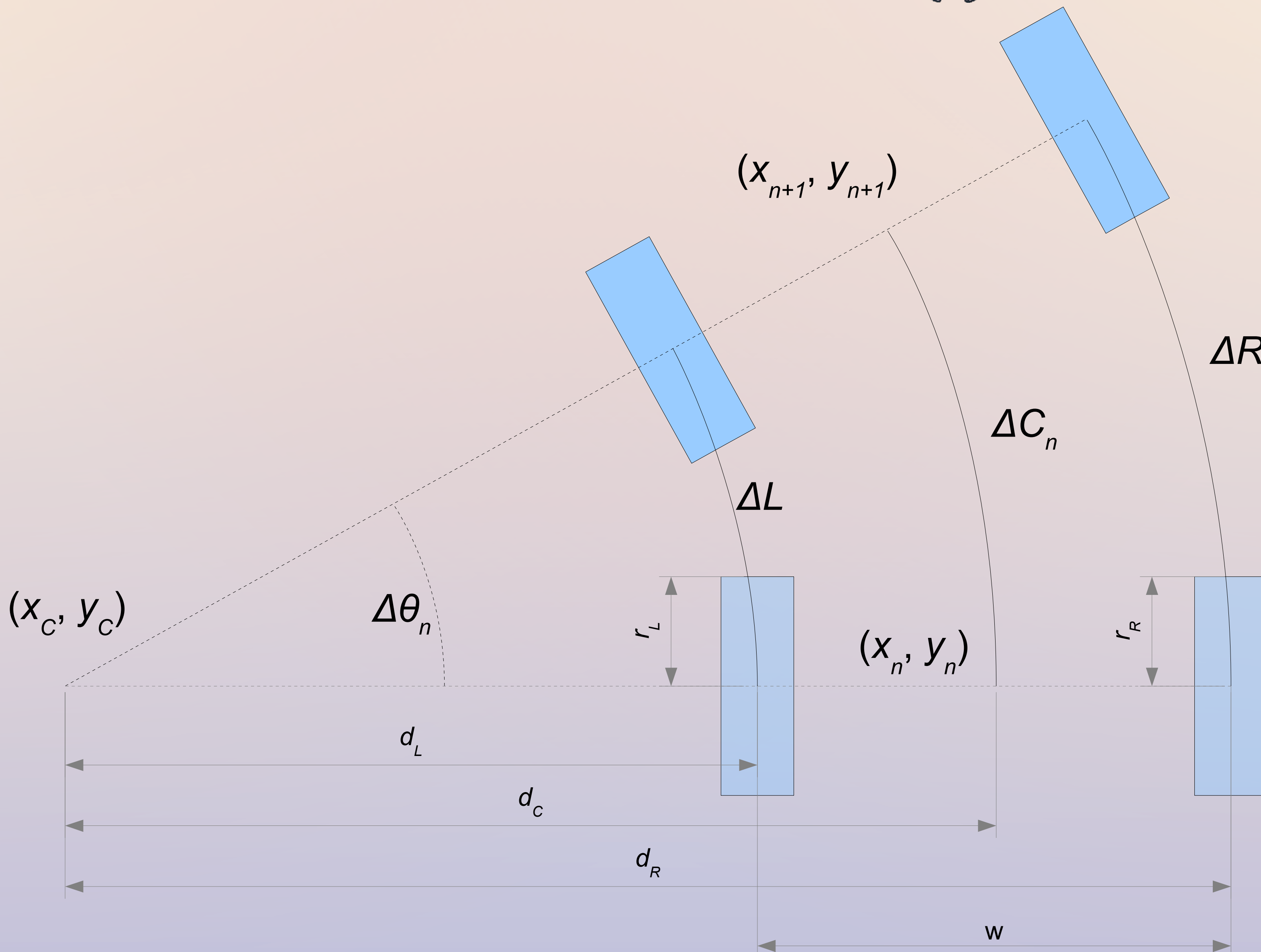
McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

Derivation – 1

- By assumptions 1 and 2, the motion of the wheels of the robot (that is, the time-derivative of their position vectors) must be perpendicular to the vector pointing from one wheel to the other
- This implies that the motion of the robot (again, over a *sufficiently short* period of time) is a circular arc

Derivation – 1

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



Derivation – 2

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

- The graphic on the previous slide represents the expected motion of the robot (as per Derivation – 1) over a *sufficiently short* time interval
- If we define the position of the robot's centre point at a timestep *n* as:

$$\vec{r}_n = (x_n, y_n)$$

we seek its change in position, $\Delta \vec{r}_n$, such that

$$\vec{r}_{n+1} = \vec{r}_n + \Delta \vec{r}_n$$

Derivation – 2

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

- Let's define λ_n and ρ_n as the tachometers (in radians) of the left and right wheel motors, respectively, at a timestep n, and $\Delta\lambda_n$ and $\Delta\rho_n$ as their change from timestep n to timestep $n+1$. The from the diagram on slide 7:

$$\Delta\lambda_n r_L = \Delta L = \Delta\theta_n d_L$$
$$\Delta\rho_n r_R = \Delta R = \Delta\theta_n d_R$$

where r_L and r_R are the left and right wheel radii, respectively, and ΔL and ΔR are the arc lengths travelled by the wheels.

Derivation – 2

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

- Since, from the last slide:

$$\Delta\theta_n d_L = \Delta\lambda_n r_L$$
$$\Delta\theta_n d_R = \Delta\rho_n r_R$$

and:

$$d_R - d_L = w$$

- we solve for $\Delta\theta_n$ with:

$$\Delta\theta_n d_R - \Delta\theta_n d_L = \Delta\theta_n (d_R - d_L) = \Delta\rho_n r_R - \Delta\lambda_n r_L$$
$$\Delta\theta_n = \frac{\Delta\rho_n r_R - \Delta\lambda_n r_L}{d_R - d_L} = \frac{\Delta\rho_n r_R - \Delta\lambda_n r_L}{w}$$

Derivation – 2

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

- Similarly, we seek ΔC, the arc length travelled by the centre of the robot:

$$\Delta C_n = d_C \Delta\theta_n$$

$$= \frac{d_R + d_L}{2}\Delta\theta_n$$

$$= \frac{\Delta\theta_n d_R + \Delta\theta_n d_L}{2} = \frac{\Delta\rho_n r_R + \Delta\lambda_n r_L}{2}$$

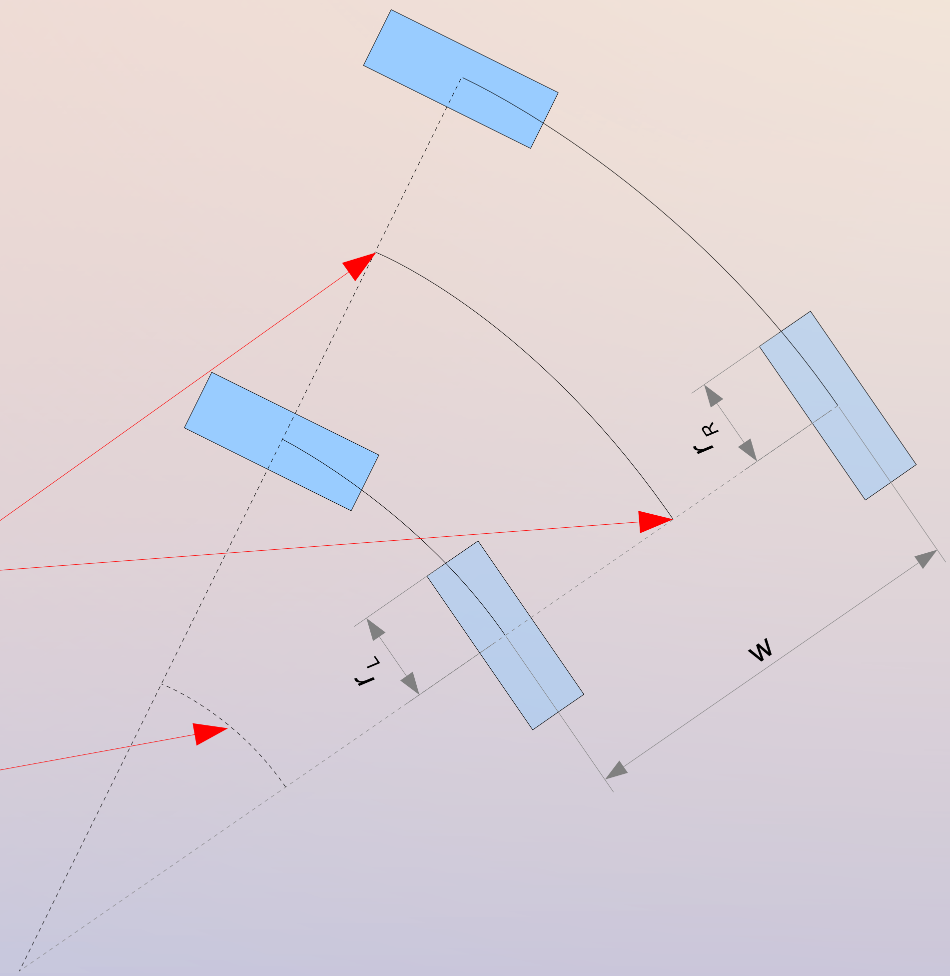
Derivation – 2

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

- We now have two equations, expressing ΔC and $\Delta\theta$ as a function of known values:

$$\Delta C = \frac{\Delta\rho_n r_R + \Delta\lambda_n r_L}{2}$$

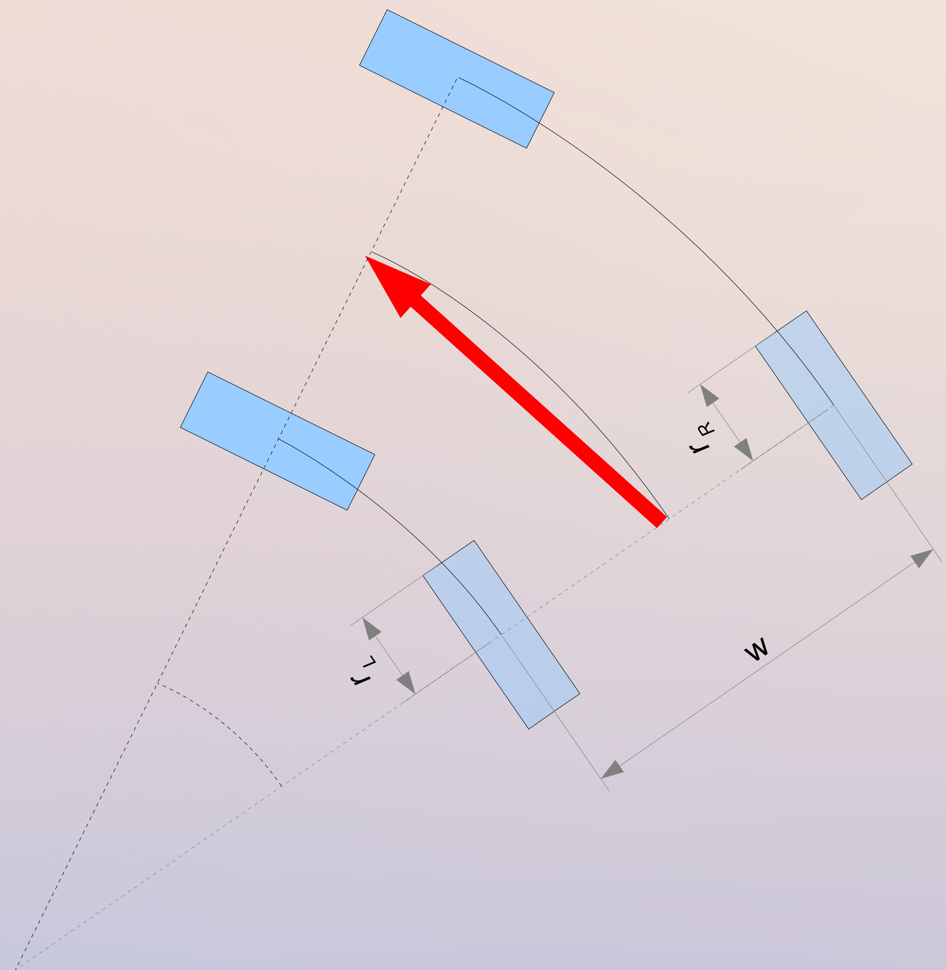
$$\Delta\theta = \frac{\Delta\rho_n r_R - \Delta\lambda_n r_L}{w}$$



Derivation – 2

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

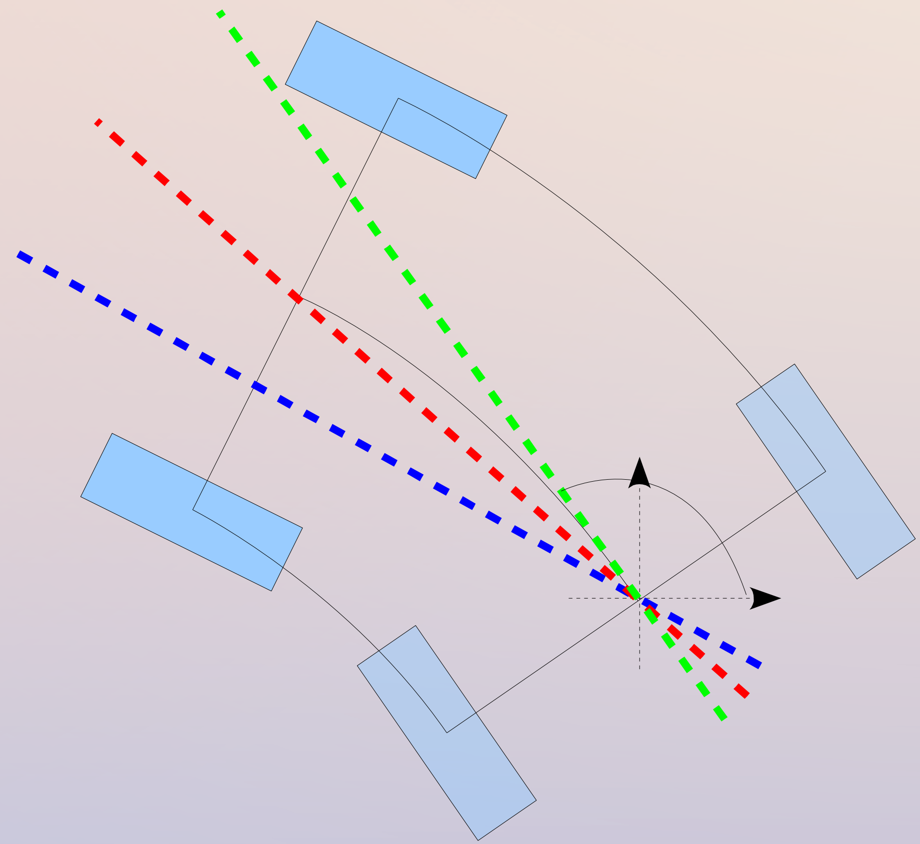
- However, what is sought is the vector $\Delta \vec{r}_n$, shown on the right in red
- We will need to do some math with ΔC and $\Delta\theta_n$ to get the magnitude and direction of $\Delta \vec{r}_n$



Derivation – 2

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

- In the diagram on the right, the green line represents the initial orientation of the robot, θ_n; the blue, the final orientation, θ_{n+1}; and the red, the direction of the $\Delta \vec{r}_n$ vector



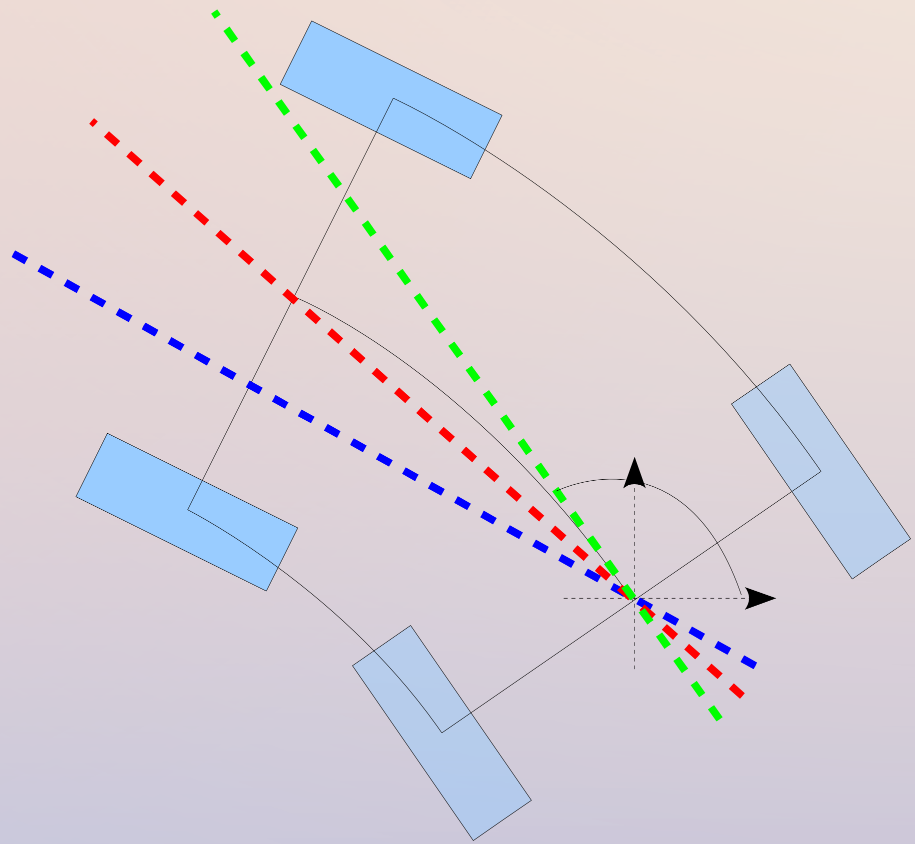
Derivation – 2

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

- This is a geometric argument (others exist) for the fact that:

$$\measuredangle \Delta \vec{r}_n = \frac{\theta_n + \theta_{n+1}}{2}$$

$$= \theta_n + \frac{\Delta \theta_n}{2}$$

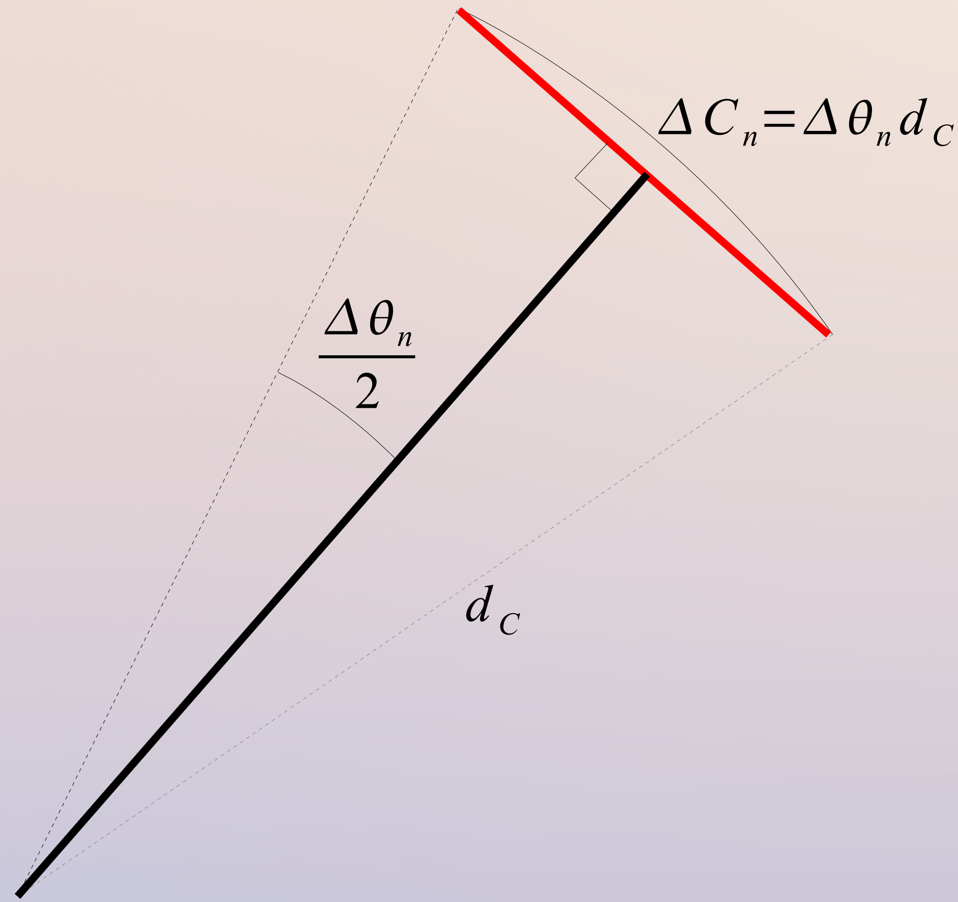


McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

Derivation – 2

- Similarly, from the diagram on the right we can derive an expression for $\|\Delta \vec{r}_n\|$ (shown in red):

$$\|\Delta \vec{r}_n\| = 2 d_C \sin\left(\frac{\Delta\theta_n}{2}\right)$$
$$= \frac{2\Delta C_n}{\Delta\theta_n}\sin\left(\frac{\Delta\theta_n}{2}\right)$$
$$= \Delta C_n\left(\frac{2}{\Delta\theta_n}\sin\left(\frac{\Delta\theta_n}{2}\right)\right)$$



Derivation – 2

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

- You should recognize the function $\frac{2}{\Delta\theta_n}\sin\left(\frac{\Delta\theta_n}{2}\right)$ as $\operatorname{sinc}\left(\frac{\Delta\theta_n}{2}\right)$, where the sinc function is defined as:

$$\operatorname{sinc}(x) = \frac{\sin(x)}{x}$$

But for small $\Delta\theta_n$,

$$\operatorname{sinc}\left(\frac{\Delta\theta_n}{2}\right) \approx 1$$

- So we approximate and say that:

$$\|\Delta\vec{r}_n\| = \Delta C_n \operatorname{sinc}\left(\frac{\Delta\theta_n}{2}\right) \approx \Delta C_n$$

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

Derivation – 2

- So we have determined $\Delta \vec{r}_n$ in polar coordinates as:

$$\|\Delta \vec{r}_n\| = \Delta \mathrm{C}_n$$

$$\sphericalangle \Delta \vec{r}_n = \theta_n + \frac{\Delta \theta_n}{2}$$

and convert to Cartesian coordinates in the usual manner:

$$\Delta \vec{r}_n = \left(\Delta C_n \cos\left(\theta_n + \frac{\Delta \theta_n}{2}\right), \Delta C_n \sin\left(\theta_n + \frac{\Delta \theta_n}{2}\right) \right)$$

Derivation – 3

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

- Finally, the recursive relationship for the x-coordinate, y-coordinate, and orientation of the robot is established:

$$\Delta C_n = \frac{\Delta \rho_n r_R + \Delta \lambda_n r_L}{2} \qquad \Delta \theta_n = \frac{\Delta \rho_n r_R - \Delta \lambda_n r_L}{w}$$

$$\vec{r}_{n+1} = \vec{r}_n + \Delta \vec{r}_n \longrightarrow \begin{aligned} x_{n+1} &= x_n + \Delta C_n \cos\left(\theta_n + \frac{\Delta \theta_n}{2}\right) \\ y_{n+1} &= y_n + \Delta C_n \sin\left(\theta_n + \frac{\Delta \theta_n}{2}\right) \\ \theta_{n+1} &= \theta_n + \Delta \theta_n \end{aligned}$$

Derivation – 3

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

- We can thus relate the position of the robot at a timestep *N* to its initial position, providing proof that our algorithm works:

$$x_N = x_0 + \sum_{n=0}^{N-1} \Delta C_n \cos\left(\theta_n + \frac{\Delta \theta_n}{2}\right)$$

$$y_N = y_0 + \sum_{n=0}^{N-1} \Delta C_n \sin\left(\theta_n + \frac{\Delta \theta_n}{2}\right)$$

$$\theta_N = \theta_0 + \sum_{n=0}^{N-1} \Delta \theta_n$$

Error – 1

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

- Errors in mathematical models (such as the one just derived) result directly from the assumptions made
- The first assumption, that the wheels are parallel, equidistant from the robot's centre point, and of constant radius, causes errors because:
  - The wheels' axles can flex, modifying their distance to the robot's centre point, as well as their parallelism
  - The rubber tire can stretch and contract, modifying the wheels' (effective) radii

Error – 2

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

- The second assumption causes errors because the wheels have a contact *area*, not point, with the floor, that changes with time
- The third and fourth assumptions, that the robot is capable of sampling the wheel motors' tachometers sufficiently quickly, and that the wheels' motion between samples is of constant velocity, will cause a varying amount of error depending on the definition of 'sufficiently' (i.e. a larger sample period will reduce the odometer's accuracy)

Error Correction

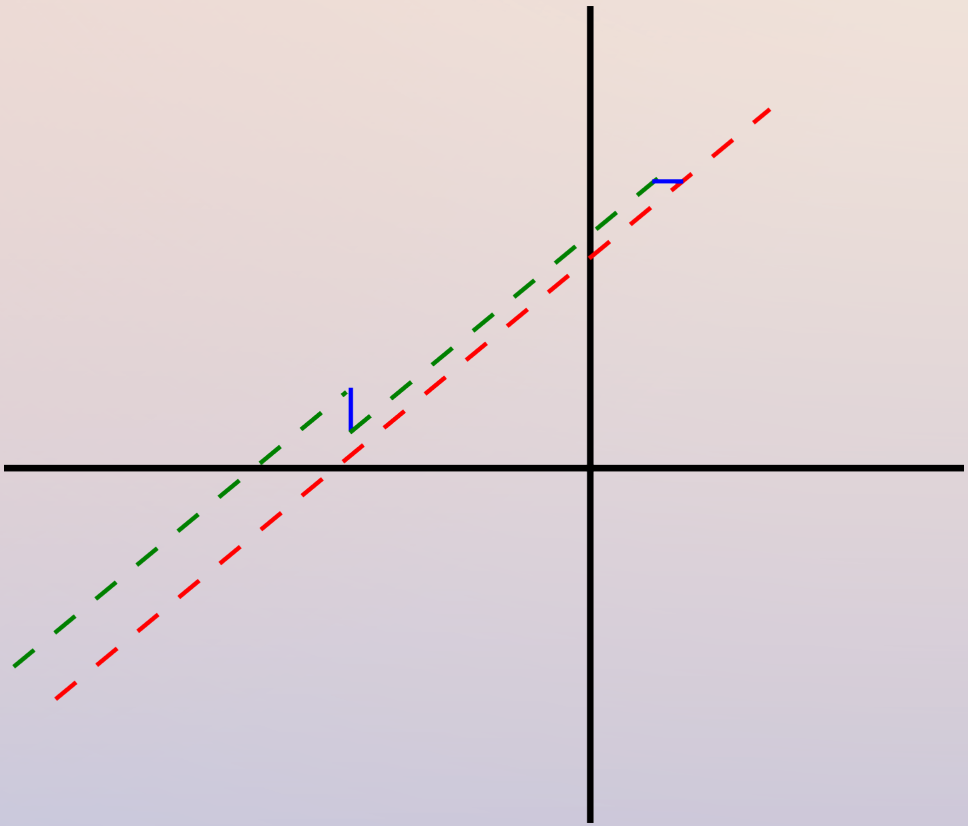
McGill University
Department of Electrical and Computer Engineering
Patrick Diez, U4 Undergraduate
Sean Lawlor, U4 Undergraduate

- Over time, these errors will accrue, resulting in unusable position information from the odometer
- Grid lines, detectable by the light sensor, are provided for the purpose of stopping (or, more precisely, limiting) this accrual
- Correction of the odometer's position data is done by 'snapping' to grid lines

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

Error Correction

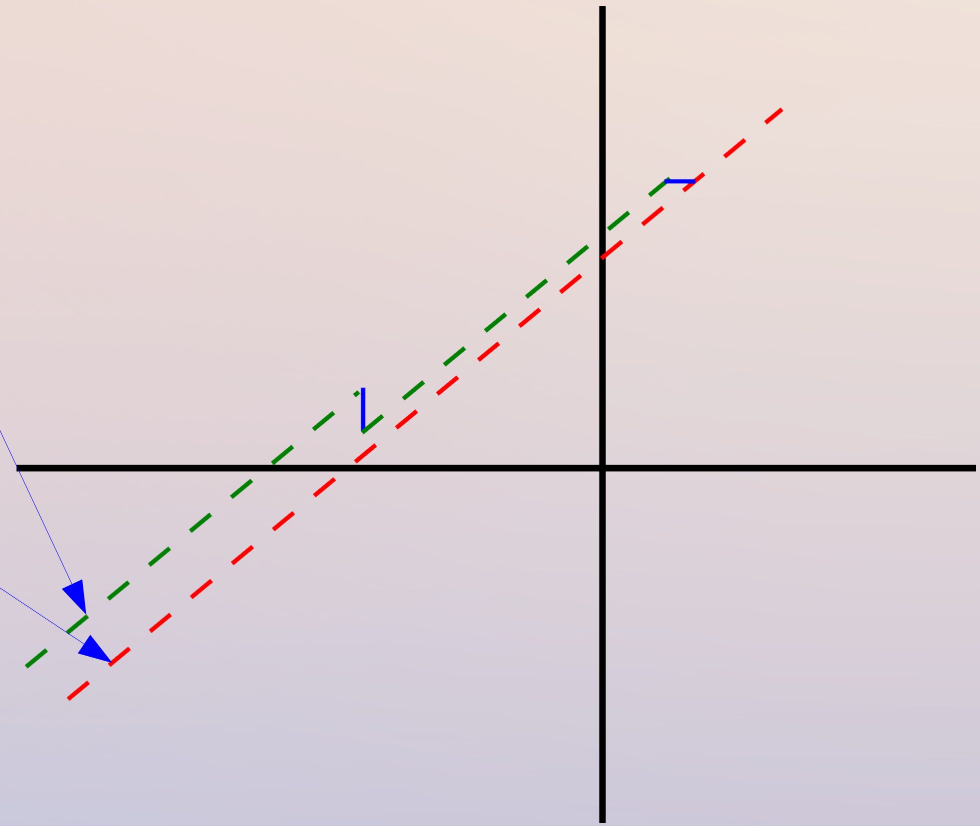
- Assume the robot is travelling along the red path shown on the right, and that its odometer is reporting the path in green
- Furthermore, assume that its light sensor is located behind its centre of rotation



Error Correction

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

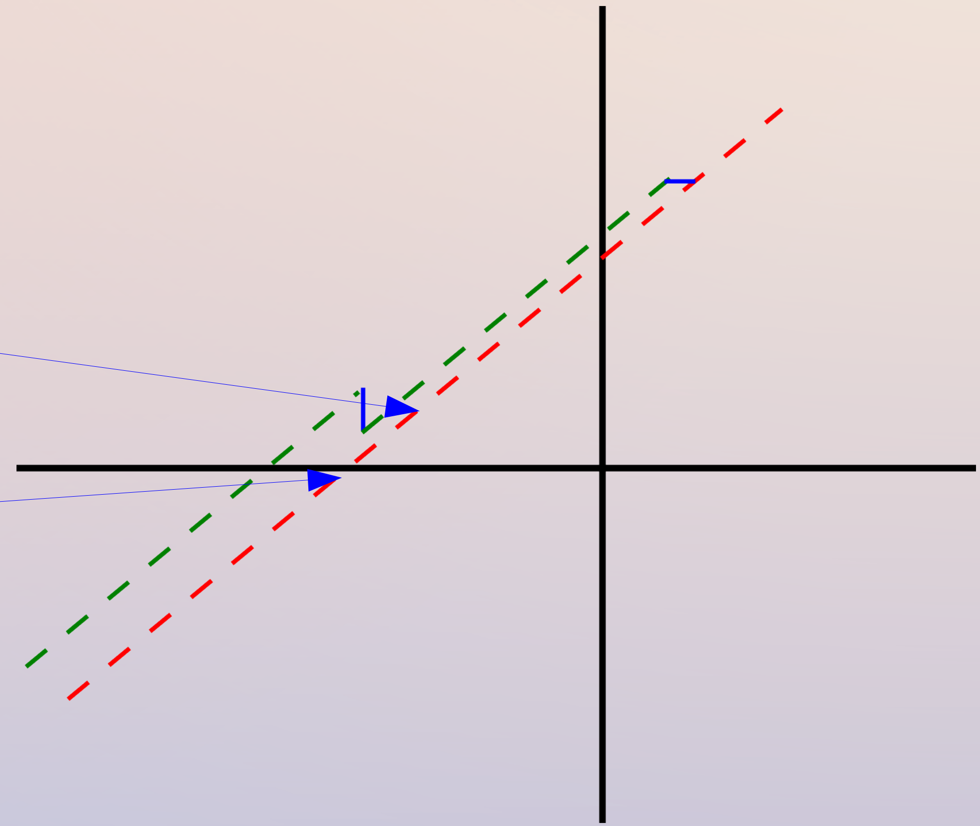
- At first, the robot's odometer is incorrectly reporting its location
- The robot is in fact located here



Error Correction

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

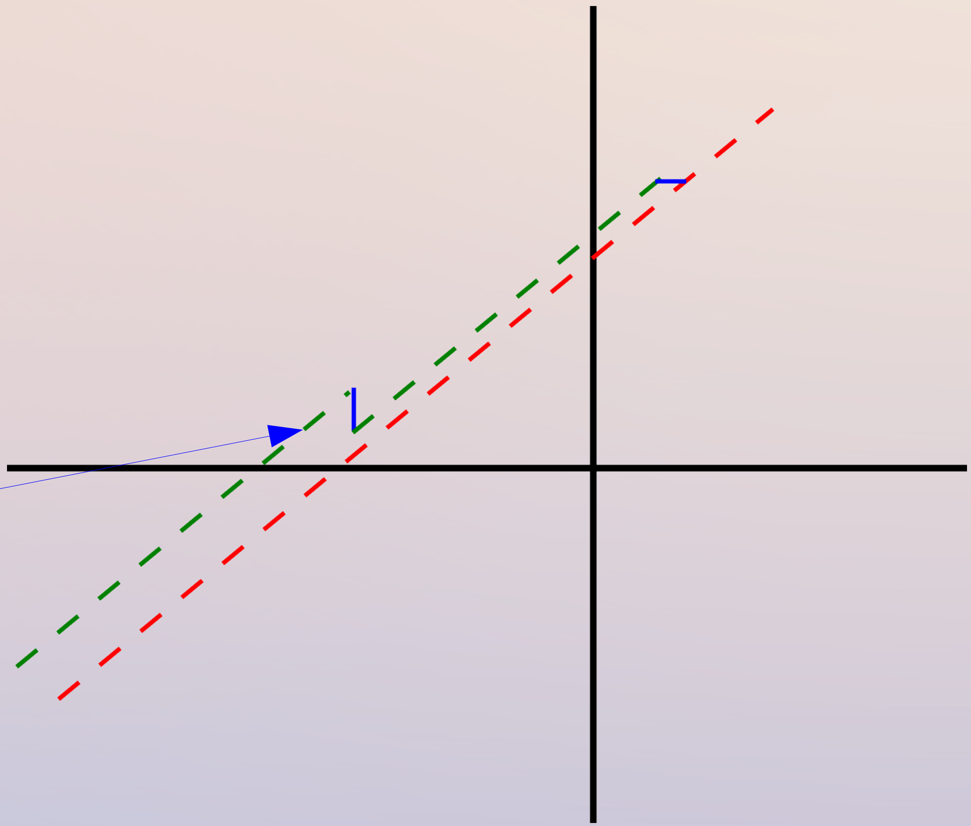
- The robot's centre of rotation (the point the odometer should report) is located here when the light sensor crosses the grid line here



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

Error Correction

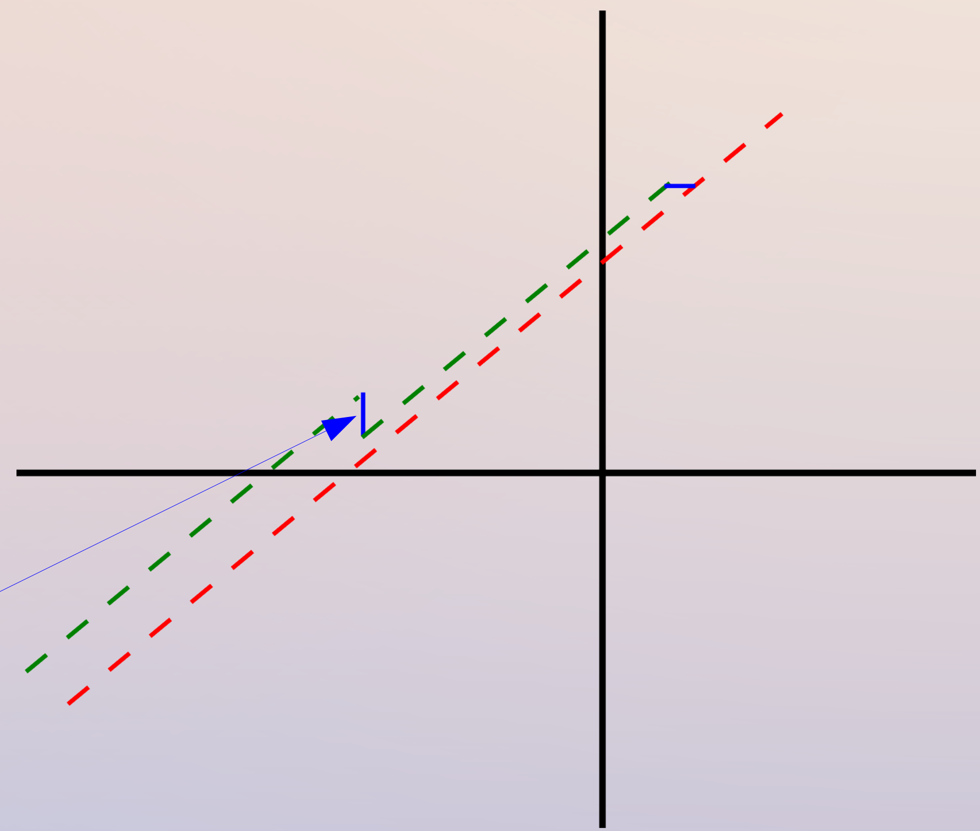
- However, if we calculate the position of the light sensor based on the odometer's (erroneous) information, we find it to be here, off the grid line



Error Correction

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

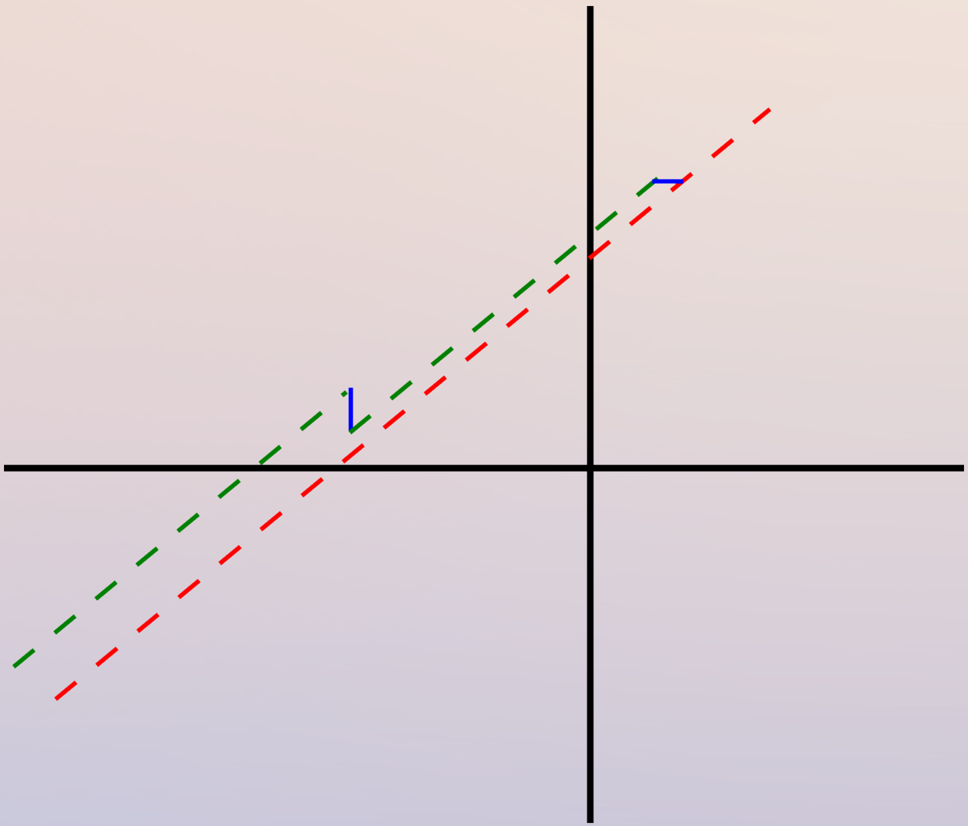
- Thus, the odometer's *y position* is corrected by the difference between the light sensor's calculated position and the nearest grid line (seen in blue)



Error Correction

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

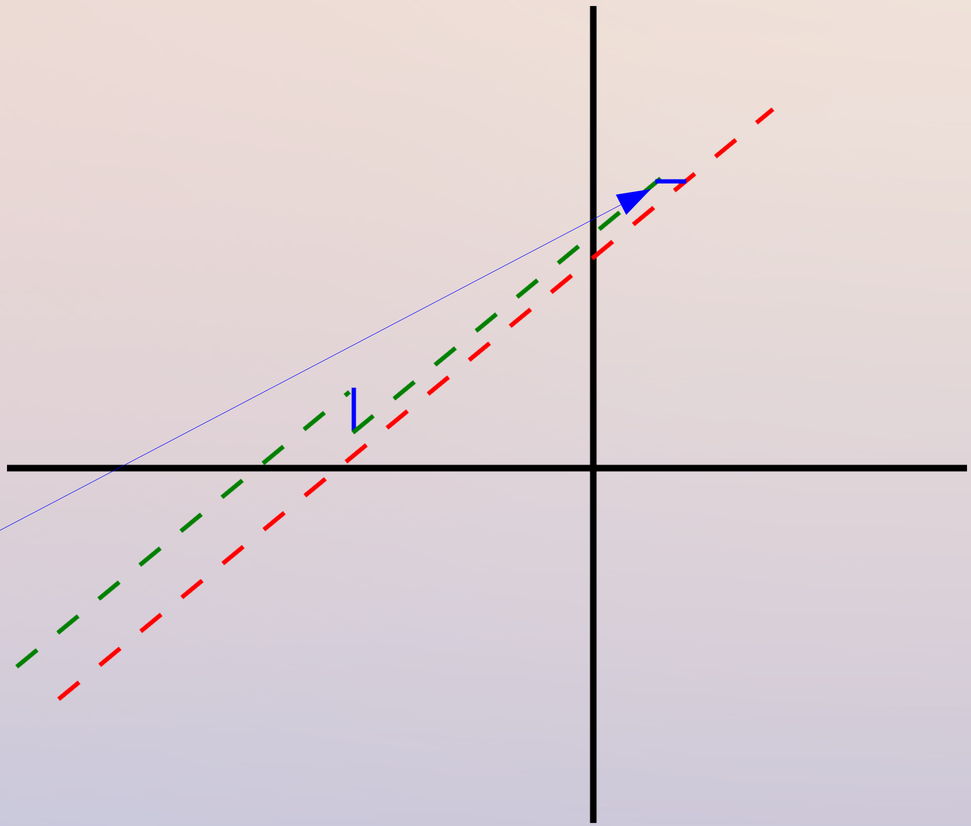
- During this process, the odometer is interrupted only momentarily to perform the correction
- It continues to update the robot's position as the robot moves



Error Correction

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

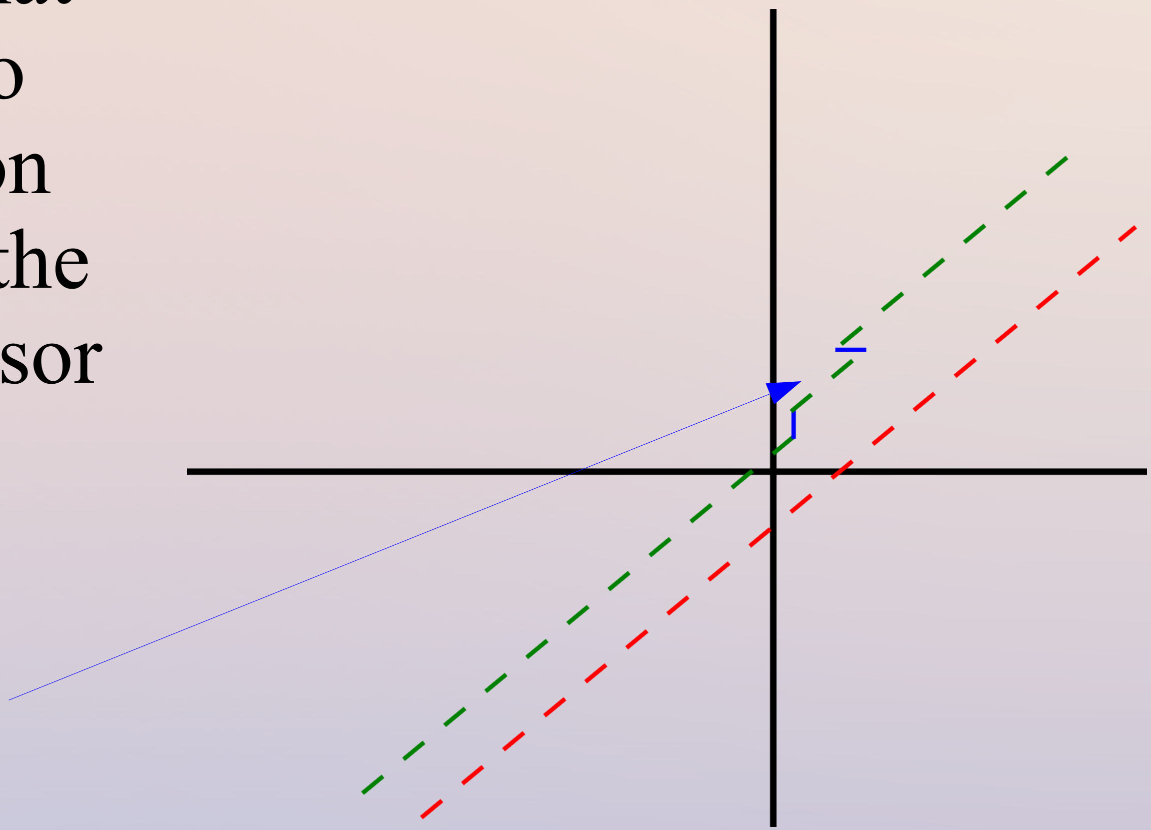
- Similarly, when the robot crosses a *vertical* grid line, its *x position* can be corrected



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

Error Correction

- Note that these corrections assume that the grid line closest to the *calculated* position of the light sensor is the grid line the light sensor has crossed
- Near grid line intersections, this assumption does not hold



Error Correction

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

- There are three important characteristics of this correction method to note:
  1. It is path-dependent. If the robot only travels parallel to grid lines, only one of the two components of its position will be corrected. Choosing paths that cross grid lines of alternating orientation is thus advantageous.
  2. It assumes perfect knowledge of the heading of the robot. The error caused by this assumption is reduced if the light sensor is located close to the centre of rotation of the robot.
  3. It does not work near grid line intersections. This is resolved by foregoing corrections when the robot is near grid line intersections.

Summary

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

- This tutorial has presented the basics of determining the robot's position in 2D space using the knowledge of the motion of its wheels
- This determination relied on assumptions, which overlooked realistic aspects of the robot's operation
- These oversights, in turn, resulted in inaccuracies in the position reported by the odometer
- With additional information (the light sensor's detection of grid lines), the inaccuracies were corrected, and their consequences mitigated

Afterword

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

- The process used here to develop the odometer can be generalized. It is, in essence, a four-step process:
  i. A goal is established: to track the robot's position
  ii. Observations are made about the robot's design: its wheel's tachometers provide position information (though not in Cartesian coordinates)
  iii. Assumptions are made about the robot's design. These, in turn, simplify the fourth step -
  iv. Modelling (mathematically) the robot's behaviour

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

Afterword

- If too few assumptions are made, modelling may become difficult if not impossible
- Conversely, if too many assumptions are made, the resulting model may be insufficiently accurate to be of use
  - Example: Assuming that both wheels are of the same radius
- Note that through this entire process, *no code was written*. This "pen and paper" preparation *prior* to writing code is *essential* for efficient work.