Robot Kinematic Constraints

- Example: A Differential-Drive Robot
- Example: A Turning Bicycle
- Using the Forward Kinematic Equation
The kinematic constraints on a robot come from the combination of constraints from its wheels:

\[
\begin{align*}
\text{Rolling constraint:} & \quad \sin(\alpha + \beta) - \cos(\alpha + \beta) (-l) \cos(\beta) \dot{\xi}_R = r \dot{\phi} \\
\text{Sliding constraint:} & \quad \cos(\alpha + \beta) \sin(\alpha + \beta) l \sin(\beta) \dot{\xi}_R = 0
\end{align*}
\]

Castor, Swedish, and spherical wheels impose no constraints. Therefore, we consider only constraints from fixed and steerable standard wheels. For each wheel, we have both rolling and sliding constraints; therefore, two equations per wheel. We stack all of these constraints together into matrix form as illustrated in the examples that follow...
Robot Kinematic Constraints

- The kinematic constraints on a robot come from the combination of constraints from its wheels:
  - The rolling constraint:
    \[ [\sin(\alpha + \beta) - \cos(\alpha + \beta) (-l) \cos(\beta)] \dot{\xi}_R = r \dot{\phi} \]
The kinematic constraints on a robot come from the combination of constraints from its wheels:

- The rolling constraint:
  \[ [\sin(\alpha + \beta) - \cos(\alpha + \beta) (-l) \cos(\beta)] \dot{\xi}_R = r\dot{\phi} \]
The kinematic constraints on a robot come from the combination of constraints from its wheels:

- The rolling constraint:
  \[\begin{bmatrix} \sin(\alpha + \beta) - \cos(\alpha + \beta) (-l) \cos(\beta) \end{bmatrix} \dot{\xi}_R = r\dot{\phi}\]

- The sliding constraint:
  \[\begin{bmatrix} \cos(\alpha + \beta) \sin(\alpha + \beta) l \sin(\beta) \end{bmatrix} \dot{\xi}_R = 0\]
The kinematic constraints on a robot come from the combination of constraints from its wheels:

- The rolling constraint:
  \[
  \begin{bmatrix}
  \sin(\alpha + \beta) - \cos(\alpha + \beta) (-l \cos(\beta))
  \end{bmatrix}
  \dot{\xi}_R = r \dot{\phi}
  \]

- The sliding constraint:
  \[
  \begin{bmatrix}
  \cos(\alpha + \beta) \sin(\alpha + \beta) l \sin(\beta)
  \end{bmatrix}
  \dot{\xi}_R = 0
  \]
The kinematic constraints on a robot come from the combination of constraints from its wheels:

- The rolling constraint:
  \[
  [\sin(\alpha + \beta) - \cos(\alpha + \beta)(-l)\cos(\beta)] \dot{\xi}_R = r\dot{\phi}
  \]

- The sliding constraint:
  \[
  [\cos(\alpha + \beta) \sin(\alpha + \beta) l \sin(\beta)] \dot{\xi}_R = 0
  \]

- Castor, Swedish, and spherical wheels impose no constraints
The kinematic constraints on a robot come from the combination of constraints from its wheels:

- **The rolling constraint:**

\[
\begin{bmatrix}
\sin(\alpha + \beta) - \cos(\alpha + \beta) (-l) \cos(\beta)
\end{bmatrix} \cdot \dot{\xi}_R = r \phi
\]

- **The sliding constraint:**

\[
\begin{bmatrix}
\cos(\alpha + \beta) \sin(\alpha + \beta) l \sin(\beta)
\end{bmatrix} \cdot \dot{\xi}_R = 0
\]

- Castor, Swedish, and spherical wheels impose no constraints.
- Therefore we consider only constraints from fixed and steerable standard wheels.
The kinematic constraints on a robot come from the combination of constraints from its wheels:

- The rolling constraint:
  \[
  [\sin(\alpha + \beta) - \cos(\alpha + \beta) (-l) \cos(\beta)] \dot{\xi}_R = r\dot{\phi}
  \]

- The sliding constraint:
  \[
  [\cos(\alpha + \beta) \sin(\alpha + \beta) l \sin(\beta)] \dot{\xi}_R = 0
  \]

- Castor, Swedish, and spherical wheels impose no constraints
- Therefore we consider only constraints from fixed and steerable standard wheels
- For each wheel, we have both rolling and sliding constraints; Therefore, two equations per wheel
Robot Kinematic Constraints

- The kinematic constraints on a robot come from the combination of constraints from its wheels:
  - The rolling constraint:
    \[ [\sin(\alpha + \beta) - \cos(\alpha + \beta)(-l)\cos(\beta)] \dot{\xi}_R = r\dot{\phi} \]
  - The sliding constraint:
    \[ [\cos(\alpha + \beta) \sin(\alpha + \beta) l\sin(\beta)] \dot{\xi}_R = 0 \]

- Castor, Swedish, and spherical wheels impose no constraints
- Therefore we consider only constraints from fixed and steerable standard wheels
- For each wheel, we have both rolling and sliding constraints; Therefore, two equations per wheel
- We stack all of these constraints together into matrix form as illustrated in the examples that follow...
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW’s are as follows:

- **Right wheel:**
  - \( \alpha_r = -\frac{\pi}{2} \)
  - \( \beta_r = \pi \) (+ve spin should cause movement in +X direction)

- **Left wheel:**
  - \( \alpha_l = \frac{\pi}{2} \)
  - \( \beta_l = 0 \)

Assume that the two wheels are equidistant from \( P \) at a distance of \( l \).
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis)
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW's are as follows:
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW’s are as follows:

- Right wheel:
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis). The parameters of the two FSW’s are as follows:

- Right wheel:
  - $\alpha_r = -\frac{\pi}{2}$
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW’s are as follows:
- Right wheel:
  - $\alpha_r = -\frac{\pi}{2}$
  - $\beta_r = \pi$ (+ve spin should cause movement in $+ X_R$ direction)
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW’s are as follows:

- Right wheel:
  - $\alpha_r = -\frac{\pi}{2}$
  - $\beta_r = \pi$ (+ve spin should cause movement in $+ X_R$ direction)

- Left wheel:
  - $\alpha_l = \frac{\pi}{2}$
  - $\beta_l = 0$
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW’s are as follows:

- Right wheel:
  - \( \alpha_r = -\frac{\pi}{2} \)
  - \( \beta_r = \pi \) (+ve spin should cause movement in + \( X_R \) direction)

- Left wheel:
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW’s are as follows:

- Right wheel:
  - $\alpha_r = -\frac{\pi}{2}$
  - $\beta_r = \pi$ (+ve spin should cause movement in $+X_R$ direction)

- Left wheel:
  - $\alpha_l = \frac{\pi}{2}$
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW’s are as follows:

- **Right wheel:**
  - $\alpha_r = -\frac{\pi}{2}$
  - $\beta_r = \pi$ (+ve spin should cause movement in $+X_R$ direction)

- **Left wheel:**
  - $\alpha_l = \frac{\pi}{2}$
  - $\beta_l = 0$
Our D-D robot has two fixed standard wheels, plus a Castor wheel for stability (plays no part in the analysis).

The parameters of the two FSW’s are as follows:

- **Right wheel:**
  - $\alpha_r = -\frac{\pi}{2}$
  - $\beta_r = \pi$ (positive spin should cause movement in $+X_R$ direction)

- **Left wheel:**
  - $\alpha_l = \frac{\pi}{2}$
  - $\beta_l = 0$

Assume that the two wheels are equidistant from $P$ at a distance of $l$
Here are the rolling constraints for both wheels:

\[
\begin{align*}
\dot{\xi}_R &= r \dot{\phi}_r \\
\dot{\xi}_R &= r \dot{\phi}_l 
\end{align*}
\]

Now the sliding constraints:

\[
\begin{align*}
\cos(\alpha_r + \beta_r) \sin(\alpha_r + \beta_r) l \sin(\beta_r) &= 0 \\
\cos(\alpha_l + \beta_l) \sin(\alpha_l + \beta_l) l \sin(\beta_l) &= 0
\end{align*}
\]

Combine all of the above into one big equation:

\[
\begin{pmatrix}
\sin(\alpha_r + \beta_r) - \cos(\alpha_r + \beta_r) (-l) \cos(\beta_r) \\
\sin(\alpha_l + \beta_l) - \cos(\alpha_l + \beta_l) (-l) \cos(\beta_l) \\
\cos(\alpha_r + \beta_r) \sin(\alpha_r + \beta_r) l \sin(\beta_r) \\
\cos(\alpha_l + \beta_l) \sin(\alpha_l + \beta_l) l \sin(\beta_l)
\end{pmatrix}
\dot{\xi}_R = r
\begin{pmatrix}
\dot{\phi}_r \\
\dot{\phi}_l \\
0 \\
0
\end{pmatrix}
\]
Here are the rolling constraints for both wheels:

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) & (-l)\cos(\beta_r) \\
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) & (-l)\cos(\beta_l)
\end{bmatrix} \dot{\xi}_R = r \dot{\phi}_r
\]

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) & (-l)\cos(\beta_r) \\
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) & (-l)\cos(\beta_l)
\end{bmatrix} \dot{\xi}_R = r \dot{\phi}_l
\]
Here are the rolling constraints for both wheels:

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) \\
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l)
\end{bmatrix}
\begin{bmatrix}
\xi_R \\
\dot{\xi}_R
\end{bmatrix}
= r \dot{\phi}_r
\]

\[
\begin{bmatrix}
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) \\
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r)
\end{bmatrix}
\begin{bmatrix}
\xi_R \\
\dot{\xi}_R
\end{bmatrix}
= r \dot{\phi}_l
\]

Now the sliding constraints:
Here are the rolling constraints for both wheels:

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) & (-l) \cos(\beta_r) \\
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) & (-l) \cos(\beta_l)
\end{bmatrix}
\dot{\xi}_R = r \dot{\phi}_r
\]

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) & (-l) \cos(\beta_r) \\
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) & (-l) \cos(\beta_l)
\end{bmatrix}
\dot{\xi}_R = r \dot{\phi}_l
\]

Now the sliding constraints:

\[
\begin{bmatrix}
\cos(\alpha_r + \beta_r) & \sin(\alpha_r + \beta_r) & l \sin(\beta_r) \\
\cos(\alpha_l + \beta_l) & \sin(\alpha_l + \beta_l) & l \sin(\beta_l)
\end{bmatrix}
\dot{\xi}_R = 0
\]

\[
\begin{bmatrix}
\cos(\alpha_r + \beta_r) & \sin(\alpha_r + \beta_r) & l \sin(\beta_r) \\
\cos(\alpha_l + \beta_l) & \sin(\alpha_l + \beta_l) & l \sin(\beta_l)
\end{bmatrix}
\dot{\xi}_R = 0
\]
Here are the rolling constraints for both wheels:

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) & (-l)\cos(\beta_r)
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_R
\end{bmatrix}
= r\dot{\phi}_r
\]

\[
\begin{bmatrix}
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) & (-l)\cos(\beta_l)
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_R
\end{bmatrix}
= r\dot{\phi}_l
\]

Now the sliding constraints:

\[
\begin{bmatrix}
\cos(\alpha_r + \beta_r) & \sin(\alpha_r + \beta_r) & l\sin(\beta_r)
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_R
\end{bmatrix}
= 0
\]

\[
\begin{bmatrix}
\cos(\alpha_l + \beta_l) & \sin(\alpha_l + \beta_l) & l\sin(\beta_l)
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_R
\end{bmatrix}
= 0
\]

Combine all of the above into one big equation:
Here are the rolling constraints for both wheels:

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) & (-l) \cos(\beta_r) \\
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) & (-l) \cos(\beta_l)
\end{bmatrix} \dot{\xi}_R = r \phi_r
\]

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) & (-l) \cos(\beta_r) \\
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) & (-l) \cos(\beta_l)
\end{bmatrix} \dot{\xi}_R = r \phi_l
\]

Now the sliding constraints:

\[
\begin{bmatrix}
\cos(\alpha_r + \beta_r) & \sin(\alpha_r + \beta_r) & l \sin(\beta_r) \\
\cos(\alpha_l + \beta_l) & \sin(\alpha_l + \beta_l) & l \sin(\beta_l)
\end{bmatrix} \dot{\xi}_R = 0
\]

Combine all of the above into one big equation:

\[
\begin{bmatrix}
\sin(\alpha_r + \beta_r) & -\cos(\alpha_r + \beta_r) & (-l) \cos(\beta_r) \\
\sin(\alpha_l + \beta_l) & -\cos(\alpha_l + \beta_l) & (-l) \cos(\beta_l) \\
\cos(\alpha_r + \beta_r) & \sin(\alpha_r + \beta_r) & l \sin(\beta_r) \\
\cos(\alpha_l + \beta_l) & \sin(\alpha_l + \beta_l) & l \sin(\beta_l)
\end{bmatrix} \dot{\xi}_R = r \begin{bmatrix} \phi_r \\ \phi_l \\ 0 \\ 0 \end{bmatrix}
\]
Our overall equation is,

\[
\begin{bmatrix}
1 & 0 & l \\
0 & 1 & 0 \\
-1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix} \dot{\xi}_R = \begin{bmatrix}
\dot{\phi}_r \\
\dot{\phi}_l \\
0 \\
0 \\
\end{bmatrix}
\]
Our overall equation is,

\[
\begin{bmatrix}
1 & 0 & l \\
1 & 0 & -l \\
0 & 1 & 0 \\
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_R
\end{bmatrix}
= \begin{bmatrix}
\dot{\phi}_r \\
\dot{\phi}_l \\
0 \\
0
\end{bmatrix}
\]
Our overall equation is,

\[
\begin{bmatrix}
1 & 0 & l \\
1 & 0 & -l \\
0 & 1 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_R \\
\dot{\phi}_r \\
\phi_l \\
0 \\
\end{bmatrix} = r
\]

We solve for \(\dot{\xi}_R\).
Example: A Turning Bicycle

A bicycle with its front wheel locked in a left turn

\[ \alpha_2 = \frac{\pi}{2} \]
\[ \beta_2 = \frac{3\pi}{4} \]

\[ \alpha_1 = -\frac{\pi}{2} \]
\[ \beta_1 = -\frac{\pi}{2} \]
Example: A Turning Bicycle

A bicycle with its front wheel locked in a left turn

\[
\begin{bmatrix}
\sin(\alpha_1 + \beta_1) & -\cos(\alpha_1 + \beta_1) & (-l_1)\cos(\beta_1) \\
\sin(\alpha_2 + \beta_2) & -\cos(\alpha_2 + \beta_2) & (-l_2)\cos(\beta_2) \\
\cos(\alpha_1 + \beta_1) & \sin(\alpha_1 + \beta_1) & l_1\sin(\beta_1) \\
\cos(\alpha_2 + \beta_2) & \sin(\alpha_2 + \beta_2) & l_2\sin(\beta_2)
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_R \\
\dot{\phi}_1 \\
\dot{\phi}_2
\end{bmatrix}
\]

\[
= r
\begin{bmatrix}
\dot{\phi}_1 \\
\dot{\phi}_2 \\
0
\end{bmatrix}
\]
\[ \begin{bmatrix} 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -1 & 0 & -1 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \dot{\xi}_R = r \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ 0 \\ 0 \end{bmatrix} \]
We apply Gauss-Jordan elimination to determine both the solution, and the condition on the existence of the solution.
\[ \begin{bmatrix} 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -1 & 0 & -1 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \dot{\xi}_R = r \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ 0 \\ 0 \end{bmatrix} \]

We apply Gauss-Jordan elimination to determine both the solution, and the condition on the existence of the solution.

\[ \begin{bmatrix} 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -1 & 0 & -1 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \]

\[ \begin{bmatrix} r \dot{\phi}_1 \\ r \dot{\phi}_2 \\ 0 \\ 0 \end{bmatrix} \]
\[
\begin{bmatrix}
0 & 1 & 0 \\
-\sqrt{2}/2 & \sqrt{2}/2 & \sqrt{2}/2 \\
-1 & 0 & -1 \\
-\sqrt{2}/2 & -\sqrt{2}/2 & \sqrt{2}/2
\end{bmatrix}
\begin{bmatrix}
\dot{\xi}_R \\
\dot{\phi}_1 \\
\dot{\phi}_2
\end{bmatrix} =
\begin{bmatrix}
r \\
0 \\
0
\end{bmatrix}
\]

We apply Gauss-Jordan elimination to determine both the solution, and the condition on the existence of the solution

\[
\begin{bmatrix}
0 & 1 & 0 & r \dot{\phi}_1 \\
-\sqrt{2}/2 & \sqrt{2}/2 & \sqrt{2}/2 & r \dot{\phi}_2 \\
-1 & 0 & -1 & 0 \\
-\sqrt{2}/2 & -\sqrt{2}/2 & \sqrt{2}/2 & 0
\end{bmatrix}
\rightarrow \text{row exchanges and combinations}
\]
After a number of steps, we arrive at,

\[
\begin{bmatrix}
1 & 0 & 0 & -r\phi_1/2 \\
0 & 1 & 0 & r\phi_1 \\
0 & 0 & 1 & r\phi_1/2 \\
0 & 0 & 0 & r\left(\frac{\sqrt{2}}{2}\phi_2 - \phi_1\right)
\end{bmatrix}
\]

This last equation is a condition on the existence of solutions; Unlike a differential drive robot, the two wheel speeds here cannot be set arbitrarily.
After a number of steps, we arrive at,

\[
\begin{bmatrix}
1 & 0 & 0 & -r\dot{\phi}_1/2 \\
0 & 1 & 0 & r\dot{\phi}_1 \\
0 & 0 & 1 & r\dot{\phi}_1/2 \\
0 & 0 & 0 & r\left(\frac{\sqrt{2}}{2}\dot{\phi}_2 - \dot{\phi}_1\right)
\end{bmatrix}
\]

Yielding the following
After a number of steps, we arrive at,
\[
\begin{bmatrix}
1 & 0 & 0 & -r\dot{\phi}_1/2 \\
0 & 1 & 0 & r\dot{\phi}_1 \\
0 & 0 & 1 & r\dot{\phi}_1/2 \\
0 & 0 & 0 & r\left(\frac{\sqrt{2}}{2}\dot{\phi}_2 - \dot{\phi}_1\right)
\end{bmatrix}
\]
Yielding the following
\[
\dot{x}_R = -r\dot{\phi}_1/2, \quad \dot{y}_R = r\dot{\phi}_1, \quad \dot{\theta} = r\dot{\phi}_1/2,
\]
After a number of steps, we arrive at,

\[
\begin{bmatrix}
1 & 0 & 0 & -r\dot{\phi}_1/2 \\
0 & 1 & 0 & r\dot{\phi}_1 \\
0 & 0 & 1 & r\dot{\phi}_1/2 \\
0 & 0 & 0 & r \left( \frac{\sqrt{2}}{2} \dot{\phi}_2 - \dot{\phi}_1 \right)
\end{bmatrix}
\]

Yielding the following

\[
\begin{align*}
\dot{x}_R &= -r\dot{\phi}_1/2, \\
\dot{y}_R &= r\dot{\phi}_1, \\
\dot{\theta} &= r\dot{\phi}_1/2,
\end{align*}
\]

\[
\dot{\phi}_1 = \frac{\sqrt{2}}{2} \dot{\phi}_2
\]
After a number of steps, we arrive at,

\[
\begin{bmatrix}
1 & 0 & 0 & -r\dot{\phi}/2 \\
0 & 1 & 0 & \dot{\phi} \\
0 & 0 & 1 & \dot{\phi}/2 \\
0 & 0 & 0 & r\left(\frac{\sqrt{2}}{2}\dot{\phi}_2 - \dot{\phi}_1\right)
\end{bmatrix}
\]

Yielding the following

\[
\dot{x}_R = -r\dot{\phi}/2, \quad \dot{y}_R = r\dot{\phi}, \quad \dot{\theta} = r\dot{\phi}/2,
\]

\[
\dot{\phi}_1 = \frac{\sqrt{2}}{2}\dot{\phi}_2
\]

This last equation is a condition on the existence of solutions; Unlike a differential drive robot, the two wheel speeds here cannot be set arbitrarily
Using the Forward Kinematic Equation

- Odometry:

Let us say we know $\xi_I(t)$ for the previous time step and we wish to determine the pose for current time $t'$. From the motors' optical encoders we can get an estimate of the wheels' current roll speeds: $\dot{\phi}_1, \dot{\phi}_2, ...$ Using the forward kinematic equation we obtain: $\dot{\xi}_R$. We can obtain $\dot{\xi}_I$ using our current estimate for $\theta$. We apply a first-order Taylor series expansion:

$$\xi_I(t') = \xi_I(t) + (t' - t) \dot{\xi}_I(t) + \cdots$$

(The "\cdots" represents higher-order terms that we don’t bother to include in a first-order approximation)

This equation can be applied iteratively to localize the robot over time. However, it will certainly drift as time passes.
Odometry:

One use for the forward kinematic equation is to allow a robot’s current pose to be tracked (known as odometry or *dead reckoning*).
Using the Forward Kinematic Equation

- **Odometry:**
  - One use for the forward kinematic equation is to allow a robot’s current pose to be tracked (known as odometry or *dead reckoning*).
  - Let us say we know $\xi_I(t)$ for the previous time step and we wish to determine the pose for current time $t'$.
Using the Forward Kinematic Equation

- Odometry:
  - One use for the forward kinematic equation is to allow a robot’s current pose to be tracked (known as odometry or dead reckoning).
  - Let us say we know $\xi_I(t)$ for the previous time step and we wish to determine the pose for current time $t'$. 
  - From the motors’ optical encoders we can get an estimate of the wheels’ current roll speeds: $\dot{\phi}_1, \dot{\phi}_2, \ldots$
Using the Forward Kinematic Equation

- **Odometry:**
  - One use for the forward kinematic equation is to allow a robot’s current pose to be tracked (known as odometry or *dead reckoning*).
  - Let us say we know $\xi_I(t)$ for the previous time step and we wish to determine the pose for current time $t'$.
  - From the motors’ optical encoders we can get an estimate of the wheels’ current roll speeds: $\dot{\phi}_1, \dot{\phi}_2, \ldots$
  - Using the forward kinematic equation we obtain: $\dot{\xi}_R$. 

(The "···" represents higher-order terms that we don’t bother to include in a first-order approximation.)
Using the Forward Kinematic Equation

**Odometry:**

- One use for the forward kinematic equation is to allow a robot’s current pose to be tracked (known as odometry or *dead reckoning*).
- Let us say we know $\xi_I(t)$ for the previous time step and we wish to determine the pose for current time $t'$.
- From the motors’ optical encoders we can get an estimate of the wheels’ current roll speeds: $\dot{\phi}_1, \dot{\phi}_2, \ldots$.
- Using the forward kinematic equation we obtain: $\dot{\xi}_R$.
- We can obtain $\dot{\xi}_I$ using our current estimate for $\theta$.
Using the Forward Kinematic Equation

- **Odometry:**
  - One use for the forward kinematic equation is to allow a robot’s current pose to be tracked (known as odometry or *dead reckoning*).
  - Let us say we know $\xi_I(t)$ for the previous time step and we wish to determine the pose for current time $t'$.
  - From the motors’ optical encoders we can get an estimate of the wheels’ current roll speeds: $\dot{\phi}_1, \dot{\phi}_2, \ldots$
  - Using the forward kinematic equation we obtain: $\dot{\xi}_R$.
  - We can obtain $\dot{\xi}_I$ using our current estimate for $\theta$.
  - We apply a first-order Taylor series expansion:
Using the Forward Kinematic Equation

- **Odometry:**
  - One use for the forward kinematic equation is to allow a robot’s current pose to be tracked (known as odometry or *dead reckoning*).
  - Let us say we know $\xi_I(t)$ for the previous time step and we wish to determine the pose for current time $t'$.
  - From the motors’ optical encoders we can get an estimate of the wheels’ current roll speeds: $\dot{\phi}_1, \dot{\phi}_2, ...$
  - Using the forward kinematic equation we obtain: $\dot{\xi}_R$
  - We can obtain $\dot{\xi}_I$ using our current estimate for $\theta$.
  - We apply a first-order Taylor series expansion:
Using the Forward Kinematic Equation

- **Odometry:**
  - One use for the forward kinematic equation is to allow a robot’s current pose to be tracked (known as odometry or *dead reckoning*).
  - Let us say we know $\xi_I(t)$ for the previous time step and we wish to determine the pose for current time $t'$.
  - From the motors’ optical encoders we can get an estimate of the wheels’ current roll speeds: $\dot{\phi}_1, \dot{\phi}_2, \ldots$
  - Using the forward kinematic equation we obtain: $\dot{\xi}_R$.
  - We can obtain $\dot{\xi}_I$ using our current estimate for $\dot{\theta}$.
  - We apply a first-order Taylor series expansion:
    $$\xi_I(t') = \xi_I(t) + (t' - t)\dot{\xi}_I(t) + \cdots$$

  - (The “$\cdots$” represents higher-order terms that we don’t bother to include in a first-order approximation.)
Using the Forward Kinematic Equation

- Odometry:
  - One use for the forward kinematic equation is to allow a robot's current pose to be tracked (known as odometry or dead reckoning)
  - Let us say we know $\xi_t(t)$ for the previous time step and we wish to determine the pose for current time $t'$
  - From the motors' optical encoders we can get an estimate of the wheels' current roll speeds: $\dot{\phi}_1, \dot{\phi}_2, \ldots$
  - Using the forward kinematic equation we obtain: $\dot{\xi}_R$
  - We can obtain $\dot{\xi}_i$ using our current estimate for $\theta$
  - We apply a first-order Taylor series expansion:
    $$\xi_i(t') = \xi_i(t) + (t' - t)\dot{\xi}_i(t) + \cdots$$
    (The “\cdots” represents higher-order terms that we don’t bother to include in a first-order approximation)
  - This equation can be applied iteratively to localize the robot over time. However, it will certainly drift as time passes.