Introduction to Mobile Robotics

Jeff McGough
Larry Pyeatt

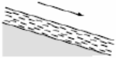
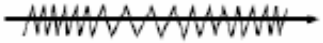
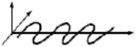
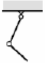
Department of Computer Science
South Dakota School of Mines and Technology
Rapid City, SD 57701, USA

September 12, 2012

Locomotion

Machine motion often has biological roots – except the wheel.

Locomotion

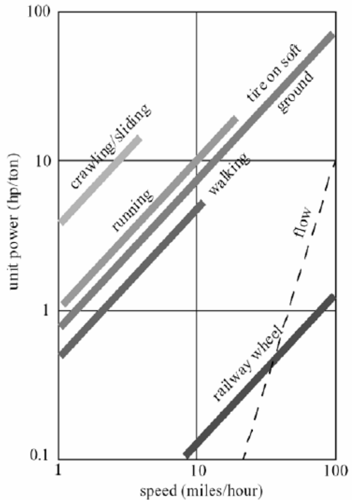
Type of motion	Resistance to motion	Basic kinematics of motion
Flow in a Channel 	Hydrodynamic forces	Eddies 
Crawl 	Friction forces	 Longitudinal vibration
Sliding 	Friction forces	 Transverse vibration
Running 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Jumping 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Walking 	Gravitational forces	Rolling of a polygon (see figure 2.2) 

Locomotion

- Found in nature
  Difficult to imitate
- Most mechanical systems use wheels or tracks
- Rolling is very efficient
  Nature does not use wheels
- Human movement can be modeled as a rolling polygon

Walk or Roll

- Number of actuators
- Structural complexity
- Control issues
- Energy expense
- Internal mass movement



Mobility Issues

Stability

- Number of contact points
- Center of gravity
- Static/Dynamic stabilization
- Terrain

Contact

- Contact area
- Angle of contact
- Friction

Environment

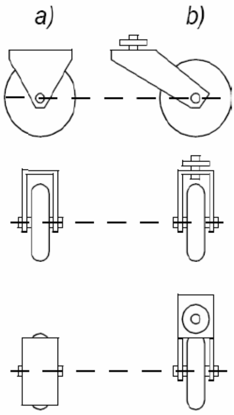
- Structure
- Medium

Wheels

- Wheels work well in many applications
- Three wheels give stability
- More than three require suspension
- Wheel type depends on application
- Steering may be required

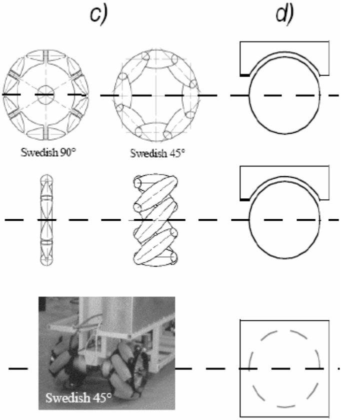
Wheel Types

- Standard Wheel: Two degrees of freedom. 1. Rotation about the axle. 2. Rotation about the contact point.
- Castor Wheel: Three degrees of freedom. 1. Rotation about the axle. 2. Rotation about the contact point. 3. Rotation about the castor axle.

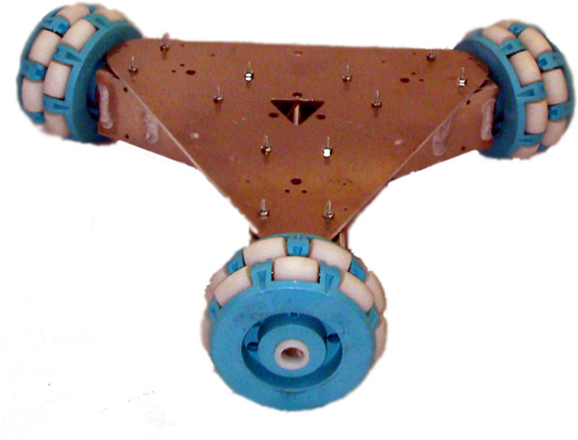


Wheel Types

- Swedish Wheel: Three degrees of freedom. 1. Rotation about the axle. 2. Rotation about the rollers. 3. Rotation about the contact point.
- Ball or Spherical Wheel: Suspension technically not solved. If implemented with Swedish wheels, one may obtain three degrees of freedom.



Example



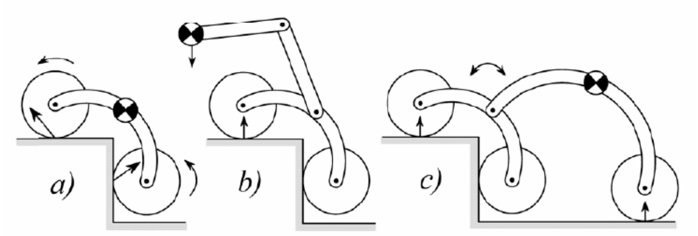
Characteristics

- Stability of a vehicle is guaranteed with three wheels: center of gravity is within the triangle is formed by the ground contact points of the wheels.
- Stability may be improved by 4 or more wheels: however, this requires a suspension system. More complicated but more robust.
- Bigger wheels: allow movement over larger obstacles (lower angle of incidence) but require higher torque or higher gear ratios.
- Most arrangements are non-holonomic meaning that there are no exact formulas for motion, so motion models must be solved using differential equation techniques.
- Combining drive and steering on one wheel increases complexity and reduces the accuracy of odometry.

Walking Wheels

- Passive locomotion concept
- Six wheels
  - two non-steered wheels on each side
  - rear steered wheel (single)
  - front steered wheel with suspension (single)
- length 60 cm
- height 20 cm
- Characterstics
  - very stable on rough terrain
  - can overcome obstacles 2x wheel diameter

Adaptive Suspension



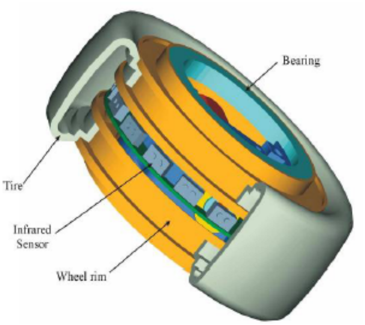
Purely friction based

Change of center of gravity (CoG)

Adapted suspension mechanism with passive or active joints

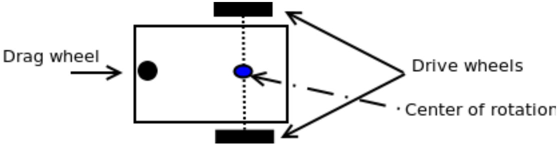
Flexible Wheel

- Better grip in loose ground
- Measure the contact force and points
- Active locomotion concept
  - Eight motorized wheels
  - Integrate sensor data with control
- Sensors
  - Inclinometer
  - Joint sensors
  - Tactile wheels

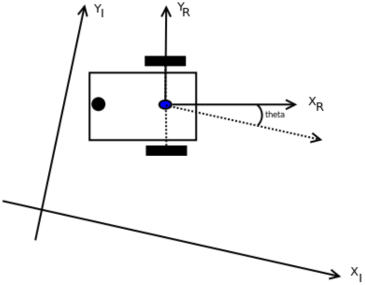


Kinematics

Begin with a small differential drive robot.



Need to establish local and global reference frames. Use the center of rotation to track the robot.



Kinematics

Inertial basis: X_I, Y_I
Relative basis: X_R, Y_R
In the local or relative basis, the X axis will be aligned in the direction of travel and the Y axis will be orthogonal (in this case along the axle direction).
Combine:

$$\xi_I = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

In the global or intertial reference frame, x, y is the location, θ is the orientation.
In the local or relative frame, the origin is placed at the center of the axle which is the center of rotation. We keep the rotation variable the same in both systems for simplicity.

Kinematics

We will derive the Kinematics in three major steps:

1. Find the instantaneous velocities that describe the robot's motion: $\dot{x}$, $\dot{y}$, and $\dot{\theta}$.
2. Integrate the velocities to find the robot's pose at any given time: $x(t)$, $y(t)$, $\theta(t)$.
3. Invert the equations for pose and velocity to <u>achieve</u> a desired motion.

Kinematics

Movement of the robot traces a path: $x(t)$, $y(t)$. What does this look like?

We need to relate global and local frames. This is related via the rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The relation depends on the orientation of the robot which changes from instant to instant. We can relate orientation at an instantaneous time:

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I$$

Note for $\theta = 45^\circ$

$$R(\theta) = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Kinematics

We actually want

$$\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$$

Since R is an orthogonal matrix[1], the inverse is easy to compute.

$$R(\theta)^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To apply this we need to know the specifics of the robot:

- Wheel size
- Axle length
- Origin of local coordinate system

[1] Strang: Linear Algebra

Kinematics

We can undo the rotation easily. Since R is an orthogonal matrix[2], the inverse is easy to compute.

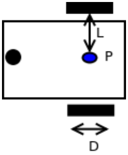
$$R(\theta)^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To apply this we need to know the specifics of the robot:

- Wheel size
- Axle length
- Origin of local coordinate system

[2] Strang: Linear Algebra

Kinematics



Wheel size $= D$, so the radius $r = D/2$
Axle length is $2L$ and the center, P, is placed on the midpoint of the axle.

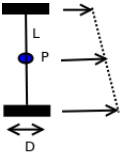
Let $\dot{\phi}_1$ and $\dot{\phi}_2$ be the right and left wheel rotational speeds (respectively).

Note: ϕ is an angle and measured in radians, $\dot{\phi}$ is measured in radians per unit time, and $\dot{\phi}/2\pi$ is the "rpm" (or rps, etc).

Kinematics

We can start by computing the linear velocity of each wheel with respect to the surface it is rolling on:

Right wheel: $\dot{x_1} = r\dot{\phi_1}$ Left wheel: $\dot{x_2} = r\dot{\phi_2}$



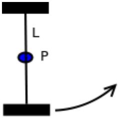
The velocity of point P is given by the moment of the velocities (weighted average based on distances of the wheels to P).

$$\dot{x_R} = \frac{1}{2}r\dot{\phi_1} + \frac{1}{2}r\dot{\phi_2} = \frac{r}{2}(\dot{\phi_1} + \dot{\phi_2})$$

For this model, there is no motion parallel to the axle so, $\dot{y_R} = 0$.

Kinematics

What about $\dot{\theta}$? Work in the local coordinates first.
If the right wheel moves faster than the left wheel, then we have positive rotation of the vehicle.
Each wheel will act like a lever arm rotating the craft.



Contribution from the right wheel: $2L\dot{\theta} = r\dot{\phi}_1$ or $\dot{\theta} = r\dot{\phi}_1/(2L)$
Contribution from the left wheel: $2L\dot{\theta} = -r\dot{\phi}_2$ or $\dot{\theta} = -r\dot{\phi}_2/(2L)$
The rotation about P is given by adding the contributions:

$$\dot{\theta} = \frac{r}{2L}(\dot{\phi}_1 - \dot{\phi}_2)$$

Differential Drive Model

To get the model in global coordinates we must map from local to global. We have

$$\dot{\xi}_I = R(\theta)\dot{\xi}_R = R(\theta) \begin{bmatrix} \frac{r}{2}(\dot{\phi}_1 + \dot{\phi}_2) \\ 0 \\ \frac{r}{2L}(\dot{\phi}_1 - \dot{\phi}_2) \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r}{2}(\dot{\phi}_1 + \dot{\phi}_2) \\ 0 \\ \frac{r}{2L}(\dot{\phi}_1 - \dot{\phi}_2) \end{bmatrix} = \begin{bmatrix} \frac{r}{2}(\dot{\phi}_1 + \dot{\phi}_2)\cos(\theta) \\ \frac{r}{2}(\dot{\phi}_1 + \dot{\phi}_2)\sin(\theta) \\ \frac{r}{2L}(\dot{\phi}_1 - \dot{\phi}_2) \end{bmatrix}$$

Differential Drive Model

Thus we have the following equations of motion

$$\dot{x} = \tfrac{r}{2}(\dot{\phi_1} + \dot{\phi_2})\cos(\theta)$$

$$\dot{y} = \tfrac{r}{2}(\dot{\phi_1} + \dot{\phi_2})\sin(\theta)$$

$$\dot{\theta} = \tfrac{r}{2L}(\dot{\phi_1} - \dot{\phi_2})$$

If you know $\dot{\phi_1}$ and $\dot{\phi_2}$, position may be found by integration.

Integration

Given this equation

$$\dot{\theta} = \frac{d\theta}{dt} = \frac{r}{2L}(\dot{\phi}_1 - \dot{\phi}_2)$$

integrate from 0 to t (and be careful about integration variables)

$$\int_0^t \frac{d\theta}{d\tau}\, d\tau = \int_0^t \frac{r}{2L}(\dot{\phi}_1 - \dot{\phi}_2)\, d\tau$$

and we have

$$\theta(t) - \theta(0) = \int_0^t \frac{r}{2L}(\dot{\phi}_1 - \dot{\phi}_2)\, d\tau$$

Solution to the differential drive model

Assume that you know $\phi_i(t)$ (or if you know $\dot{\phi}_i(t)$), then what can you say?

From $\dot{\phi}_i(t)$ we can compute θ by integrating the last equation:

$$\theta(t) = \theta(0) + \int_0^t \frac{r}{2L}\left(\frac{d\phi_1}{d\tau} - \frac{d\phi_2}{d\tau}\right) d\tau$$

Using this result we can write down formulas for x and y

$$x(t) = x(0) + \int_0^t \frac{r}{2}\left(\frac{d\phi_1}{d\tau} + \frac{d\phi_2}{d\tau}\right)\cos(\theta(\tau))d\tau$$

$$y(t) = y(0) + \int_0^t \frac{r}{2}\left(\frac{d\phi_1}{d\tau} + \frac{d\phi_2}{d\tau}\right)\sin(\theta(\tau))d\tau$$

Example

It is possible to solve these equations if you know the form of ϕ_1 and ϕ_2.

Assume that the wheels are 18cm in diameter and L is 12cm.

Find an analytic solution and compute the position of the robot starting at t=0, x=0, y=0, theta=0, after the following sequence of moves:

$t = 0 \rightarrow 5$: $\phi_1 = \phi_2 = 3.0$,
$t = 5 \rightarrow 6$: $\phi_1 = -\phi_2 = 2.0$,
$t = 6 \rightarrow 10$: $\phi_1 = \phi_2 = 3.0$,
$t = 10 \rightarrow 11$: $-\phi_1 = \phi_2 = 2.0$,
$t = 11 \rightarrow 16$: $\phi_1 = \phi_2 = 3.0$,

Example

These equations are easy to integrate if you know the wheel velocities are constants. First integrate the θ equation:

$$\theta(t) = (r/2L)(\phi_1 - \phi_2)t + c_1$$

This can be plugged into the x and y equations and then integrated:

$$x(t) = \frac{L(\phi_1 + \phi_2)}{(\phi_1 - \phi_2)} \sin((r/2L)(\phi_1 - \phi_2)t + c_1) + c_2$$

$$y(t) = -\frac{L(\phi_1 + \phi_2)}{(\phi_1 - \phi_2)} \cos((r/2L)(\phi_1 - \phi_2)t + c_1) + c_3$$

Thus the solution is a sequence of circular arcs.

Example

From these solutions (and from the differential equations as well), you can see that there is a problem when $\phi_1 = \phi_2$ or when $\phi_1 = -\phi_2$.

These are exactly the cases we are given. However, the equations are trivial for the special cases.

When $\phi_1 = \phi_2$,

$$d\theta/dt = 0$$

and when $\phi_1 = -\phi_2$

$$dx/dt = 0 \text{ and } dy/dt = 0.$$

So, you can just work out the solution from rate-time-distance formulas.

Example

Begin at $(x, y, \theta) = (0, 0, 0)$
$t = 0 \to 5$: $\phi_1 = \phi_2 = 3.0, \Rightarrow (0, 0, 0) + (135, 0, 0) = (135, 0, 0)$

$t = 5 \to 6$: $\phi_1 = -\phi_2 = 2.0, \Rightarrow (135, 0, 0) + (0, 0, 3/2) = (135, 0, 3/2)$

$t = 6 \to 10$: $\phi_1 = \phi_2 = 3.0, \Rightarrow$
$(135, 0, 3/2) + (108 \cos 3/2, 108 \sin 3/2, 0) \approx (142.6, 107.7, 1.5)$

$t = 10 \to 11$: $-\phi_1 = \phi_2 = 2.0, \Rightarrow$
$(142.6, 107.7, 1.5) + (0, 0, -1.5) = (142.6, 107.7, 0)$

$t = 11 \to 16$: $\phi_1 = \phi_2 = 3.0, \Rightarrow$
$(142.6, 107.7, 0) + (135, 0, 0) = (277.6, 107.7, 0)$

Problems...

Problems:

1. What if you can't integrate the function to get θ?
2. With θ determined, what if you can't integrate the x and y formulas?

Example: Let $\dot{\phi}_1 = e^{-t^2}$ and $\dot{\phi}_2 = t$

$$\theta(t) = \theta(0) + \int_0^t \frac{r}{2L}\left(e^{-\tau^2} - \tau\right) d\tau = ???$$

Now what does one do? There is another problem as well.

Discrete approximation

What if you don't have $\phi(t)$ and are measuring the wheel angular velocity during runtime, AND it is clearly not constant?

We will use Euler's method for solving the differential equations.
The Euler method is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. It is the most basic explicit method for numerical integration of ordinary differential equations and is the simplest Runge–Kutta method.

The Euler method is a first-order method, which means that the local error (error per step) is proportional to the square of the step size, and the global error (error at a given time) is proportional to the step size. It also suffers from stability problems. For these reasons, the Euler method is not often used in practice. It serves as the basis to construct more complicated methods, and should be good enough for our needs.

Discrete approximation

Let the time between measurements (the step size) be denoted by Δt.

The discretized variables are

$$t_k \equiv k\Delta t, \quad t_{k+1} = (k+1)\Delta t$$

$$x_k \equiv x(t_k), \qquad y_k \equiv y(t_k)$$

$$\phi_{1,k} \equiv \dot{\phi}_1(t_k), \qquad \phi_{2,k} \equiv \dot{\phi}_2(t_k)$$

Discrete approximation

Recall that if x is position then $\dot{x}$ is velocity (and $\ddot{x}$ is acceleration).

One may approximate a derivative by

$$\dot{x} \approx \frac{x(t + \Delta t) - x(t)}{\Delta t}$$

Take a time step of Δt (meaning $t_{k+1} = t_k + \Delta t$), Euler's ("Oil-ler's") method is

$$x(t_{k+1}) = x(t_k) + (\Delta t)x'(t_k) \quad \text{and} \quad y(t_{k+1}) = y(t_k) + (\Delta t)y'(t_k).$$

And so we can write our differential equations as difference equations...

Discrete approximation

$$\frac{x(t+\Delta t)-x(t)}{\Delta t} \approx \dot{x} = \frac{r}{2}(\dot{\phi}_1+\dot{\phi}_2)\cos(\theta)$$

$$\frac{y(t+\Delta t)-y(t)}{\Delta t} \approx \dot{y} = \frac{r}{2}(\dot{\phi}_1+\dot{\phi}_2)\sin(\theta)$$

$$\frac{\theta(t+\Delta t)-\theta(t)}{\Delta t} \approx \dot{\theta} = \frac{r}{2L}(\dot{\phi}_1-\dot{\phi}_2)$$

Algebra:

$$x(t+\Delta t) \approx x(t) + \tfrac{r\Delta t}{2}(\dot{\phi}_1+\dot{\phi}_2)\cos(\theta)$$

$$y(t+\Delta t) \approx y(t) + \tfrac{r\Delta t}{2}(\dot{\phi}_1+\dot{\phi}_2)\sin(\theta)$$

$$\theta(t+\Delta t) \approx \theta(t) + \tfrac{r\Delta t}{2L}(\dot{\phi}_1-\dot{\phi}_2)$$

Discrete Differential Drive Model

Using the discrete (sample) variables, $x(t) \to x_k$, etc, we can rewrite the expression in terms of the discrete variables.

Given starting configuration and wheel velocity measurements, we have:

$$x_{k+1} = x_k + \frac{r\Delta t}{2}(\phi_{1,k} + \phi_{2,k})\cos(\theta_k)$$

$$y_{k+1} = y_k + \frac{r\Delta t}{2}(\phi_{1,k} + \phi_{2,k})\sin(\theta_k)$$

$$\theta_{k+1} = \theta_k + \frac{r\Delta t}{2L}(\phi_{1,k} - \phi_{2,k})$$

Euler Code Example

In C one may only return a single object. To get around this, we bundle up our pose (three vector) into a single item, named here `triple_t`:

```
typedef struct
{
  double x;
  double y;
  double th;
} triple_t;
```

You can use this to type data:

```
 triple_t delta;
```

Euler Code

We can build a function that returns the special datatype and can return the velocities:

```
triple_t vel(double w1, double w2, double t, double th)
{
   triple_t delta;

   delta.x = 4.5*(w1 + w2)*cos(th);
   delta.y = 4.5*(w1 + w2)*sin(th);
   delta.th = 0.375*(w1-w2);
   return delta;
}
```

Euler Code

Then one can call this in a loop - known as time stepping.

```
for(i=0;i<LOOPS;i++)
 {
     t += dt;
     delta = vel(w1, w2, t,th);
     x += dt*delta.x;
     y += dt*delta.y;
     th += dt*delta.th;
  }
```

Euler Code output

```
t = 0.200000,  x = 5.400000,  y = 0.000000,  theta = 0.000000
t = 0.400000,  x = 10.800000,  y = 0.000000,  theta = 0.000000
t = 0.600000,  x = 16.200000,  y = 0.000000,  theta = 0.000000
t = 0.800000,  x = 21.600000,  y = 0.000000,  theta = 0.000000
t = 1.000000,  x = 27.000000,  y = 0.000000,  theta = 0.000000
t = 1.200000,  x = 32.400000,  y = 0.000000,  theta = 0.000000
t = 1.400000,  x = 37.800000,  y = 0.000000,  theta = 0.000000
<cut>
t = 14.800000,  x = 245.239618,  y = 107.729459,  theta = 0.000000
t = 15.000000,  x = 250.639618,  y = 107.729459,  theta = 0.000000
t = 15.200000,  x = 256.039618,  y = 107.729459,  theta = 0.000000
t = 15.400000,  x = 261.439618,  y = 107.729459,  theta = 0.000000
t = 15.600000,  x = 266.839618,  y = 107.729459,  theta = 0.000000
t = 15.800000,  x = 272.239618,  y = 107.729459,  theta = 0.000000
t = 16.000000,  x = 277.639618,  y = 107.729459,  theta = 0.000000
```