• We’ll start on the bottom right hand side.
Motion

- Two aspects:
  - Locomotion
  - Kinematics
- Locomotion: What kinds of motion are possible?
- Locomotion: What physical structures are there?
- Kinematics: Mathematical model of motion.
- Kinematics: Models make it possible to predict motion.

Locomotion in nature

<table>
<thead>
<tr>
<th>Type of motion</th>
<th>Resistance to motion</th>
<th>Basic kinematics of motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow in a Channel</td>
<td>Hydrodynamic forces</td>
<td>Eddies</td>
</tr>
<tr>
<td>Crawl</td>
<td>Friction forces</td>
<td>Longitudinal vibration</td>
</tr>
<tr>
<td>Sliding</td>
<td>Friction forces</td>
<td>Transverse vibration</td>
</tr>
<tr>
<td>Running</td>
<td>Loss of kinetic energy</td>
<td>Periodic bouncing on a spring</td>
</tr>
<tr>
<td>Walking</td>
<td>Loss of kinetic energy</td>
<td>Rolling of a polygon (see figure 2.2)</td>
</tr>
</tbody>
</table>
**Sliding**

- Snakes have four *gaits*.
- Lateral undulation (most common)
- Concertina
- Sidewinding
- Rectilinear

**Crawling**

- Concertina and Rectilinear motion can be considered crawling.

- Not directly implemented.
- The Makro (left) and Omnitread (right) robots crawl, but not exactly like real snakes do.
**Characterisation of locomotion**

- Locomotion:
  - Physical interaction between the vehicle and its environment.
- Locomotion is concerned with interaction forces, and the mechanisms and actuators that generate them.
- The most important issues in locomotion are:
  - Stability
  - Characteristics of contact
  - Nature of environment

<table>
<thead>
<tr>
<th>Stability</th>
<th>Characteristics of contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number and geometry of contact points</td>
<td>Contact point or contact area</td>
</tr>
<tr>
<td>Center of gravity</td>
<td>Angle of contact</td>
</tr>
<tr>
<td>Static/dynamic stabilization</td>
<td>Friction</td>
</tr>
<tr>
<td>Inclination of terrain</td>
<td>Nature of environment</td>
</tr>
<tr>
<td></td>
<td>Structure</td>
</tr>
<tr>
<td></td>
<td>Medium (water, air, soft or hard ground)</td>
</tr>
</tbody>
</table>
Legged motion – adaptability / maneuverability

Legged motion - Stability

- The fewer legs the more complicated locomotion becomes
  - at least three legs are required for static stability
- During walking some legs are lifted
- For static walking at least 6 legs are required
  - Babies have to learn for quite a while until they are able to stand or walk on their two legs.

mammal (2 or 4)  reptile  insect
**Leg joints**

- A minimum of two DOF is required to move a leg forward
  - A lift and a swing motion.
  - Sliding-free motion in more than one direction not possible
- Three DOF for each leg in most cases
- Fourth DOF for the ankle joint
  - Might improve walking
  - However, additional joint (DOF) increase the complexity of the design and especially of the locomotion control.

**Leg structure**

- How many degrees of freedom does this leg have?
Numbers of gaits

- Gait is characterized as the sequence of lift and release events of the individual legs.
  - Depends on the number of legs.
- The number of possible events $N$ for a walking machine with $k$ legs is:
  \[ N = (2k - 1)! \]
- For a biped walker ($k=2$) the number of possible events $N$ is:
  \[ N = (4 - 1)! = 3! = 3 \times 2 \times 1 = 6 \]
- The 6 different events are:
  - lift right leg / lift left leg / release right leg / release left leg / lift both legs together / release both legs together

- For a robot with 6 legs (hexapod), such as:

  ![Hexapod Robot]

  - $N$ is already
  \[ N = 11! = 39,916,800 \]
Obvious 6-legged gait

- Static stability— the robot is always stable.

(six-legged-crawl)

Obvious 4-legged gaits

<table>
<thead>
<tr>
<th>Changeover walk</th>
<th>Gallop</th>
</tr>
</thead>
</table>

Diagram of gaits with arrows indicating leg movements.
Titan VIII

- A family of 9 robots, developed from 1976, to explore gaits.

(Aibo)
**Big Dog, Little Dog**

- Work grew out of the MIT Leg Lab.

  *(leg-lab-spring-flamingo)*

**Humanoid Robots**

- Two-legged gaits are difficult to achieve — human gait, for example is very unstable.
• Aldebaran.

*(NAO-playtime)*

**Walking or rolling?**

• Number of actuators
• Structural complexity
• Control expense
• Energy efficient
• Terrain (flat ground, soft ground, climbing..)
RHex

- Somewhere in between walking and rolling.

Wheeled robots

- Wheels are the most appropriate solution for many applications
  - Avoid the complexity of controlling legs
- Basic wheel layouts limited to easy terrain
  - Motivation for work on legged robots
  - Much work on adapting wheeled robots to hard terrain.
- Three wheels are sufficient to guarantee stability
  - With more than three wheels a flexible suspension is required
- Selection of wheels depends on the application
Four basic wheels

- Standard wheel: Two degrees of freedom; rotation around the (motorized) wheel axle and the contact point.
- Castor wheel: Two degrees of freedom; rotation around the wheel axle, and the offset castor axle.

Four standard wheels (II)

- Swedish wheel: Three degrees of freedom; rotation around the (motorized) wheel axle, around the rollers and around the contact point.
- Ball (also called spherical) wheel: Omnidirectional. Suspension is technically not solved.
Swedish wheel

- Invented in 1973 by Bengt Ilon, who was working for the Swedish company Mecanum AB.
- Also called a Mecanum or Ilon wheel.

Characteristics of wheeled vehicles

- Stability of a vehicle is guaranteed with 3 wheels.
  - Center of gravity is within the triangle which is formed by the ground contact points of the wheels.
- Stability is improved by 4 and more wheels.
  - However, such arrangements are hyperstatic and require a flexible suspension system.
- Bigger wheels allow robot to overcome higher obstacles.
  - But they require higher torque or reductions in the gear box.
- Most wheel arrangements are non-holonomic (see later)
  - Require high control effort
- Combining actuation and steering on one wheel makes the design complex and adds additional errors for odometry.
Two wheels

- Steering wheel at front, drive wheel at back.

- Differential drive. Center of mass above or below axle.

- The two wheel differential drive pattern was used in Cye.

- Cye was an early attempt at a robot for home use.
• The Segway RMP has its center of mass above the wheels

Three wheels

• Differential drive plus caster or omnidirectional wheel.

• Connected drive wheels at rear, steered wheel at front.

• Two free wheels in rear, steered drive wheel in front.
**Differential drive plus caster**

- Highly maneuverable, in moving forwards/backwards and rotating.

**Neptune**

- Neptune: an early experimental robot from CMU. Steered drive wheel in the front plus two free wheels in the back
Other three wheel drives

- Omnidirectional
- Three drive wheels.
- Swedish or spherical.

- Synchro drive
- Three drive wheels.
- Can’t control orientation.

Omnidirectional

- The Palm Pilot Robot Kit (left) and the Tribolo (right) (*Tribolo*)
**Synchro**

- All wheels are actuated synchronously by one motor
  - Defines the speed of the vehicle
- All wheels steered synchronously by a second motor
  - Sets the heading of the vehicle

(Borenstein)

- The orientation in space of the robot frame will always remain the same

**Four wheels**

- RWD car
- FWD car
- 4WD-4WS car
- 4 omnidirectional
- 2 drive wheels
- 2 omnidirectional
- 4 drive/steered castor wheels

- Various combinations of steered, driven wheels and omnidirectional wheels.
Four steering wheels

- Highly maneuverable, hard to control.

Nomad

- The Nomad series of robots from Nomadic Technologies had four casters, driven and steered.
**Uranus**

- Four omnidirectional driven wheels.
- Not a minimal arrangement, since only three degrees of freedom.
- Four wheels for stability.

---

**Tracked robots**

- Large contact area means good traction.
- Use slip/skid steering.
Slip/skid steering

- Also used on ATV versions of differential drive platforms.

Aerial robots

- Lots of interest in the last few years
- Aim is to use them for search and surveillance tasks.

- Starmac (left) and YARB (right)
Fish robots

- Incorporating various kinds of motion.

Kinematics

- So far we have looked at different kinds of motion in a qualitative way.
- One way to program robots to move is trial-and-error.
- A somewhat better way is to establish mathematically how the robot should move, this is kinematics.
- Rather kinematics is the business of figuring out how a robot will move if its motors work in a given way.
- Inverse-kinematics then tells us how to move the motors to get the robot to do what we want.
- We’ll look at two tiny bits of the kinematics world.
A formal model

- We will assume, as people usually do, that the robot’s location is fixed in terms of three coordinates:

$$(x_I, y_I, \theta_I)$$

- Given that the robot needs to navigate to a location $(x_G, y_G, \theta_G)$, it can determine how $x$, $y$ and $\theta$ need to change.
  - BUT it can’t control these directly.

- All it has access to are the speeds of its wheels:
  $\phi_1, \ldots, \phi_n$  
  n: # of driving wheels

The steering angles of the steerable wheels:

$\beta_1, \ldots, \beta_m$  
  m: # of steering wheels

and the speeds with which those steering angles are changing.

$\dot{\beta}_1, \ldots, \dot{\beta}_m$

- Together these determine the motion of the robot:

$$f(\phi_1, \ldots, \phi_n, \beta_1, \ldots, \beta_m, \dot{\beta}_1, \ldots, \dot{\beta}_m) = \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta}_I \end{bmatrix}$$

- **Forward kinematics**: given motion of wheels in robot’s reference frame, derive motion in the world reference frame (right side of equation)
• This is usually not what we want.
• But we can get what we want from this:

\[
\begin{bmatrix}
\phi_1 \\
\vdots \\
\phi_n \\
\beta_1 \\
\vdots \\
\beta_m \\
\beta_1 \\
\vdots \\
\beta_m
\end{bmatrix} = f(x_I, y_I, \dot{\theta}_I)
\]

• Inverse kinematics: given motion in world reference frame, derive motion of wheels in robot's reference frame (left side of equation)

---

**Representing robot position: two reference frames**

• The robot knows how it moves relative to its center of rotation.
• This is not the same as knowing how it moves relative to the world.

Two systems of coordinates:

- Inertial Frame: \( \{X_I, Y_I\} \)  
  (World or global coordinates)
- Robot Frame: \( \{X_R, Y_R\} \)  
  (Local coordinates, relative to own center)
• Robot position:

\[ \xi_I = [x_I, y_I, \theta_I]^T \]

• Mapping between frames:

\[ \dot{\xi}_R = R(\theta) \dot{\xi}_I \]
\[ = R(\theta)[x_I, y_I, \dot{\theta}_I]^T \]

where

\[ R(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \]

• In other words

\[
\begin{pmatrix} 
\dot{x}_R \\
\dot{y}_R \\
\dot{\theta}_R 
\end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\
-\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 
\dot{x}_I \\
\dot{y}_I \\
\dot{\theta}_I 
\end{pmatrix}
\]

meaning that:

\[
\begin{align*}
\dot{x}_R &= \dot{x}_I \cos(\theta) + \dot{y}_I \sin(\theta) + \dot{\theta}_I.0 \\
\dot{y}_R &= -\dot{x}_I \sin(\theta) + \dot{y}_I \cos(\theta) + \dot{\theta}_I.0 \\
\dot{\theta}_R &= \dot{x}_I.0 + \dot{y}_I.0 + \dot{\theta}_I.1
\end{align*}
\]
• That is the inverse kinematic model. But if we want the forward kinematic model, then:

\[
\begin{bmatrix}
\dot{x}_I \\
\dot{y}_I \\
\dot{\theta}_I
\end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} 
\dot{x}_R \\
\dot{y}_R \\
\dot{\theta}_R
\end{bmatrix}
\]

where \(R(\theta)^{-1}\) is the inverse of \(R(\theta)\).

• Often \(R(\theta)^{-1}\) is hard to compute, but luckily for us in this case it isn’t. It’s simply \(R(-\theta)\).

• We have:

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

which we can use to establish \(\dot{x}_I, \dot{y}_I, \dot{\theta}_I\)

• In other words

\[
\begin{bmatrix}
\dot{x}_I \\
\dot{y}_I \\
\dot{\theta}_I
\end{bmatrix} = \begin{bmatrix}
\cos(\theta) & -\sin(\theta) & 0 \\
\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix} 
\dot{x}_R \\
\dot{y}_R \\
\dot{\theta}_R
\end{bmatrix}
\]

meaning that:

\[
\begin{align*}
\dot{x}_I &= \dot{x}_R \cos(\theta) - \dot{y}_R \sin(\theta) + \dot{\theta}_R \cdot 0 \\
\dot{y}_I &= \dot{x}_R \sin(\theta) + \dot{y}_R \cos(\theta) + \dot{\theta}_R \cdot 0 \\
\dot{\theta}_I &= \dot{x}_R \cdot 0 + \dot{y}_R \cdot 0 + \dot{\theta}_R \cdot 1
\end{align*}
\]
Down to the structure of the robot

- We can now identify the motion of the robot, in the global frame, if we know:

\[
\dot{x}_R, \dot{y}_R, \dot{\theta}
\]

but how do we tell what these are?

- We compute it from what we can measure like the speed of the wheels:

- Some assumptions — *constraints* on the motion of the robot:
  - Movement on a horizontal plane
  - Point contact of the wheels; wheels not deformable
  - Pure rolling, so \( v = 0 \) at contact point; no slipping, skidding or sliding
  - No friction for rotation around contact point
  - Steering axes orthogonal to the surface
  - Wheels connected by rigid frame (chassis)
• Consider differential drive.

• Wheels rotate at $\phi$ (in radians/sec)

• Each wheel contributes:

  \[ \frac{r\phi}{2} \quad \text{(in meters/sec)} \]

  to motion of center of rotation.

• Total speed is the sum of two contributions.

• Rotation due to right wheel is

  \[ \omega_r = \frac{r\phi}{2l} \quad \text{(in radians/sec)} \]

  counterclockwise about the left wheel.

  \(2l\) is the distance between wheels. (in meters)

• Rotation due to the left wheel is:

  \[ \omega_l = \frac{-r\phi}{2l} \]

  counterclockwise about the right wheel.

• Combining these components we have:

  \[
  \begin{pmatrix}
  \dot{x}_R \\
  \dot{y}_R \\
  \dot{\theta}_R
  \end{pmatrix}
  =
  \begin{pmatrix}
  \frac{r\phi_r}{2} & 0 & \frac{r\phi_l}{2} \\
  0 & \frac{r\phi_r}{2l} & -\frac{r\phi_l}{2l}
  \end{pmatrix}
  \]

• And we can combine these with $R(\theta)^{-1}$ to find motion in the global frame.
Wheel geometry

- Making sure the assumptions hold imposes constraints.
- For example, ensuring that the wheel doesn’t slip says something about motion in the direction of the wheel.

More complex scenarios

- Steered standard wheel
  - More parameters.
  - Only fixed and steerable standard wheels impose constraints.

- Caster
Robot mobility

• The sliding constraint means that a standard wheel has no lateral motion.
  – Zero motion line through the axis.

![Diagram showing instantaneous center of rotation](image)

• Has to move along a circle whose center is on the zero motion line.

• A differential drive robot has just one line of zero motion.

![Robot](image)

• Thus its rotation is not constrained
  – It can move in any circle it wants.
• Makes it very easy to move around.
• This depends on the number of *independent* kinematic constraints.
Formally we have the notion of a *degree of mobility*

\[ \delta_m = 3 - \text{number of independent kinematic constraints} \]

- The independence is important.
- Differential drive has two standard wheels, but they are on the same axis.
  - So not independent.
- So \( \delta_m = 2 \) for a differential drive robot
  - Can alter \( \dot{x} \) and \( \dot{\theta} \) just through wheel velocity.
- A bicycle has two independent wheels, so \( \delta_m = 1 \)
  - Can only alter \( \dot{x} \) using wheel velocity.

---

**Steerability and Maneuverability**

- Steering has an impact on how the robot moves.
- The *degree of steerability* \( \delta_s \) is then the number of independent steerable wheels.
- Note that a steerable standard wheel will both reduce the degree of mobility and increase the degree of steerability.
- The *degree of maneuverability* is:
  \[ \delta_M = \delta_m + \delta_s \]
- \( \delta_M \) tells us how many degrees of freedom a robot can manipulate.
- Two robots with the same \( \delta_M \) aren’t necessarily equivalent (see on).
Common configurations for 3-wheelers

Holonomy

- Consider a bicycle, it has a $\delta_M = 2$ yet can position itself anywhere in the plane.
  - Workspace DOF is 3 in the general case.
  - They include dimensions in $x$, $y$ and $\theta$ (yaw)
- **Differential** DOF
  - Number of independently achievable velocities
  - DDOF is always equal to the *degree of mobility*, $\delta_m$
  - A bicycle only has one DDOF (the forward velocity)
- A general inequality:
  \[
  DDOF \leq \delta_M \leq \text{Workspace DOF}
  \]
- A robot with $DDOF = \delta_M = DOF$ is called *holonomic*
If we were interested

- We could compute forward and inverse kinematics for robot arms, and use these to decide how to rotate motors to move the hand.

Legged robots

- Of course, this is how we get robot legs to do what we want also.
Summary

- This lecture started by looking at locomotion.
- It discussed many of the kinds of motion that robots use, giving examples.
- We then looked a bit at kinematics
  - The business of relating what robots do in the world to what their motors need to be told to do.
- We did a little math, but most of the discussion was qualitative.
- The book goes more into the mathematical detail of establishing kinematic constraints.