**Motors**

**Brushed DC Motors**

\[ \text{DC = Direct Current} \]

- gaps btw commutators are slightly larger than a brush so each brush is only touching one commutator at a time.

- permanent and electromagnets cause spinning.
- magnetism imparts rotation to the shaft.
- continuously making and breaking connections causing changing magnetic fields.
- brush touches commutator and energizes armature.

**Motor Properties:**

1) can model electrically

\[ V = \frac{e}{K} \]

- voltage drop
- speed
- back EMF constant

\[ V = 1R + e = 1R + wKe \]

2) Torque and Current are proportional from reading

\[ T = KtI \leftarrow I = \frac{T}{Kt} \]

\[ V = \frac{1}{K}R + wKe \]

\[ Kt = Ke \]

\[ V = \frac{I}{K}R + wK \]

\[ \omega = \frac{V}{K} - \frac{T}{K^2R} \Rightarrow \text{linear torque/speed relationship (assuming constant V)} \]
3) Power

\[ P_M = T \omega = T \left( -\frac{R}{k^2} T + \frac{V}{k} \right) \]

\[ = -\frac{T^2 R}{k^2} + \frac{V}{k} T \Rightarrow \text{power is quadratic in torque} \]

4) Problem: Peak efficiency at high speed, low torque
Most applications are high torque, low speed

\[ \omega_2 = \frac{r_1}{r_2} \omega_1 \quad T_2 = \frac{r_2}{r_1} T_1 \]

Can shift curve

\[ G = \frac{r_1}{r_2} \]
Digital Control of Motors

1) Pulse Width Modulation (PWM)

-a way to fake an analog output with a digital output

Problem: digital outputs are only 0s and 1s
Solution: modulate signal in time

\[ \text{Duty Cycle} = \frac{T_{on}}{T_{period}} \]

- have output on for 8 ms and off for 2 ms
  \[ \Rightarrow \text{average voltage is then less than 5 V} \]
- by controlling the pulse width, changes \( V_{\text{average}} \)
- period must be small enough for motor's momentum to carry through the off section
  \( \text{Duty Cycle} = \frac{T_{on}}{T_{period}} \)

2) H-Bridge

digital circuit that controls motors

Problem: want to control direction
Solution: H-Bridge circuit

\[ \begin{array}{c|ccccc}
\text{state} & 1 & 2 & 3 & 4 \\
\hline
\text{freewheel} & & & & \checkmark \\
\text{forward} & & & \checkmark & \checkmark \\
\text{backward} & \checkmark & \checkmark & & \\
\text{brake} & \checkmark & & \checkmark & \\
\text{short circuit} & \checkmark & \checkmark & \checkmark & \checkmark \\
\end{array} \]
9/25 Kinematics not on next week's exam

**Configuration Spaces**

A configuration \( q \) specifies the state of the robot.

Example: Turtlebot \( q = (x, y, \theta) \)

Configuration space \( \mathcal{X} \)

\( \mathcal{X} = \{ \text{all possible } q \text{'s} \} \)

Planar Arm

\( q = (\theta_1, \theta_2) \) relative to fixed frame = consistent zero

- Configuration space is specific to a robot
- not just a vector in \( \mathbb{R}^n \)
  \( \theta = \theta + 2\pi = \theta + 4\pi \ldots \)

\( \Rightarrow \) Topology matters a lot

wrap into tube

\( \mathbb{R}^2 = \text{long} \rightarrow \theta \)

wrap around short

wrap into torus
Topological Spaces Common in Robotics

\( \mathbb{R}^n = \) good ol' Cartesian coordinates

\( S^1 = \) planar angles (unit circle)

\( \mathbb{SO}(3) = \) space of 3D rotations (roll, pitch, yaw or rotation vector, etc.)

Ex. \( \mathbb{R}^2 \times S^1 = \) Rigid Transformation in 2D (\( x, y, \theta \))

\( \mathbb{R}^3 \times \mathbb{SO}(3) = \) Rigid Transform in 3D (\( x, y, z, r, \theta, \phi \))

Planar Arm: \( q \in S^1 \times S^1 = S^2 \)

Kinematic System

=a mapping \( \dot{q} = \sum_{i=1}^{n} V_i(q)w_i = \begin{bmatrix} V'_1(q) & \cdots & V'_n(q) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \)

vector field

Derivative (change in state) state/configuration ith control maps to direction in configuration space example of controls: wheel velocities, inputs

Differential Drive System

\[
\begin{pmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{pmatrix} =
\begin{bmatrix}
\frac{r}{2} \cos \theta \\
\frac{r}{2} \sin \theta \\
-r \frac{d}{2}
\end{bmatrix}
\begin{bmatrix}
V_L \\
V_R \\
0
\end{bmatrix}
\]

from other class:

\[
\dot{x} = \frac{r}{2} (V_L + V_R) \\
\dot{y} = \frac{r}{2} (V_L + V_R) \\
\dot{\theta} = \frac{(V_R - V_L)}{r} d
\]

\( x = x_w, y = y_w, \theta = \theta_w \)

plug these into matrix
Constraints on Kinematic Systems

one or more functions of the form \( f(q, \dot{q}) = 0 \) or
\[
\begin{align*}
\mathbf{b}_1 &= \begin{bmatrix}
\cos \theta \\
\sin \theta
\end{bmatrix}, \\
\mathbf{b}_2 &= \begin{bmatrix}
-sin \theta \\
cos \theta
\end{bmatrix}
\end{align*}
\]

Example for Diff. Drive
\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} \cdot \begin{bmatrix}
-sin \theta \\
cos \theta
\end{bmatrix} = 0
\]
shows that movement of robot is orthogonal to \( \mathbf{b}_2 \)
\( y \cos \theta - x \sin \theta = 0 \)

\( x = K \cos \theta, \ y = K \sin \theta, \ K = x' \)
plug in:
\( K \sin \theta \cos \theta - K \cos \theta \sin \theta = 0 \)

Holonomic Constraint
= of the form \( f(q) = 0 \)
living in subset of conf. space
= does not depend on configuration derivatives
Ex. \( q = (x, y) \in \mathbb{R}^2 \)
\( f(x, y) = x^2 + y^2 - 1 = 0 \)

s can't get everywhere
Holonomic Constraint
= can be written as \( f(q) = 0 \)
= remove degrees of freedom from configuration space
Non-Holonomic Constraint
can eventually get to any point in config. space
\( y' = \frac{w \sin \theta}{r} \)
\( x' = \frac{w \cos \theta}{r} \)
\( y \cos \theta - x \sin \theta = 0 \)

\( \Rightarrow \) wagon can only move along \( \theta \) line so
actually holonomic constraint despite its form
Instantaneous Center of Curvature (ICC)

- The motion of a wheeled vehicle is only valid if all wheels are traveling on concentric circles.
- The shared center is called the ICC.

*straight line is a circle of infinite radius

Diff. Drive Robot

\[ \dot{\theta} = \frac{V_L R}{(R-d)} = \frac{V_R R}{(R+d)} \]

\[ \Rightarrow R = \frac{V_R + V_L}{V_R - V_L} \text{ d} \]  
blows up if \( V_L = V_R \)  \( \Rightarrow R = \infty \)

\[ \kappa = \frac{V_R - V_L}{V_R + V_L} \cdot \frac{1}{d} \]  
blows up if \( V_L = 0 \) or \( V_L = -V_R \)

\( \kappa \) curvature