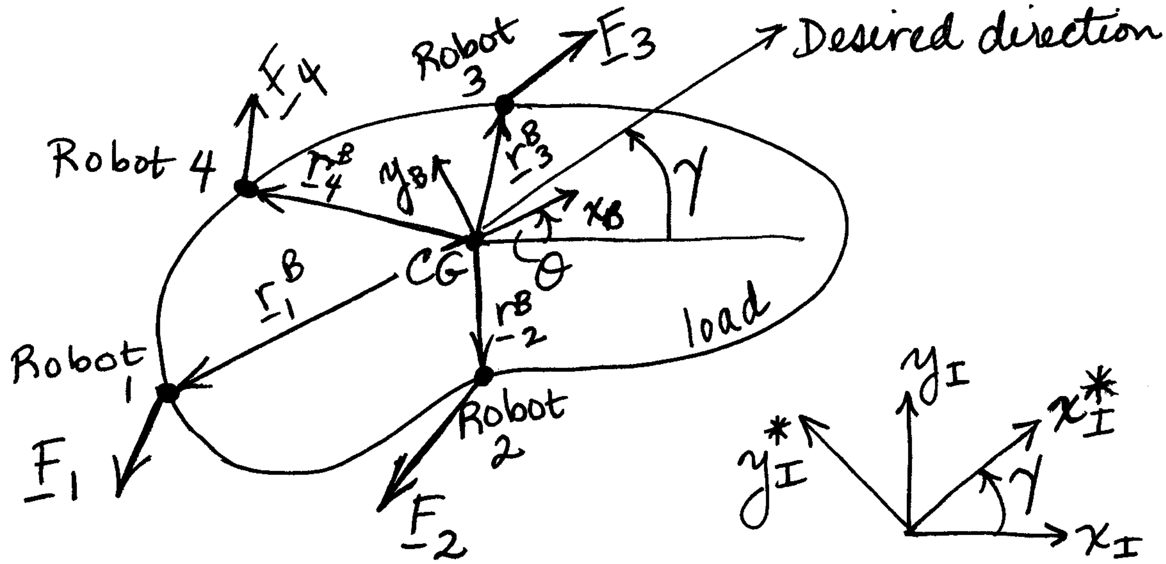
Decentralized Sliding Mode Approach to Collective Transport

(Farivarnejad, Wilson, + Berman, CDC 2016)

- Sliding mode control provides robust control of nonlinear systems; only requires bounds on uncertainties in the dynamics instead of their precise characterization.
  [Khalil, Nonlinear Systems; Slotine + Li, Applied Nonlinear Control]
- Objective: Design robot controllers that drive the transport team to move the load at a desired speed in a target direction. Load speed + direction should converge to desired values in finite time.
- Assumptions:
  - Fixed team of robots that are rigidly attached to the load and modeled as point agents
  - Robots grasp + lift the load off the ground
  - No obstacles in environment, which is planar
  - Robots are identical and arbitrarily positioned around the load
  - Robots do not have information about the environment, load, or transport team; no GPS or communication
  - Robots know the target load speed + direction of transport and can measure their own heading and speed.



N point-mass robots

$\underline{X}_i(t) \in \mathbb{R}^2$ = position of robot i at time t in inertial reference frame I

$\underline{u}_i \in \mathbb{R}^2$ = robot's actuating force

$\underline{F}_i \in \mathbb{R}^2$ = reaction force exerted by load on robot i

m_i = mass of robot i

Eq. of motion for robot i: $\quad m_i \ddot{\underline{X}}_i = \underline{u}_i - \underline{F}_i \quad$ (1)

To develop a sliding mode controller for robot i, need to write this in the form: $\quad \ddot{\underline{X}}_i = \underline{h} + \underline{G}\underline{u} \quad$ (2)

$\underline{G}$ = input matrix that is a function of the load dynamics

$\underline{h}$ = nonlinear term describing effects of load dynamics & forces applied by other robots

- Sliding mode controller only requires <u>bounds</u> on $\underline{h}$

$\underline{R}_B^I$ = rotation matrix from body-fixed frame B on the load to intertial ref. frame I

$\underline{\ddot{X}}_0^B$ = acceleration of load center of gravity (CG) in frame B

$\underline{r}_i^B$ = vector from load CG to attachment point of robot i

θ = load orientation in frame I

$\omega = \dot{\theta}$ = load angular velocity

$\alpha = \ddot{\theta}$ = load angular acceleration

Note that $\underline{a} \times \underline{b} = \underline{\hat{a}}\,\underline{b} = \underline{\hat{b}}^T \underline{a}$, where $\underline{\hat{a}} = -\underline{\hat{a}}^T$ is a skew-symmetric matrix.

ex) $\underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$, $\underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$: $\underline{a} \times \underline{b} = \underbrace{\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}}_{\underline{\hat{a}}} \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_{\underline{b}}$

- Each robot is rigidly attached to the load $\Rightarrow$

$$\underline{\ddot{X}}_i = \underline{R}_B^I \left(\underline{\ddot{X}}_0^B + \underline{\hat{\alpha}}\, \underline{r}_i^B + \underline{\hat{\omega}} \left(\underline{\hat{\omega}}\, \underline{r}_i^B \right) \right) \qquad (3)$$

Note that $\underline{\hat{\alpha}}\, \underline{r}_i^B = -\underline{\hat{r}}_i^B \underline{\alpha} = (\underline{\hat{r}}_i^B)^T \underline{\alpha}$

$$\Rightarrow \underline{\ddot{X}}_i = \underline{R}_B^I \left(\begin{bmatrix} \underline{I} & (\underline{\hat{r}}_i^B)^T \end{bmatrix} \begin{bmatrix} \underline{\ddot{X}}_0^B \\ \ddot{\theta} \end{bmatrix} + \underline{\hat{\omega}} \left(\underline{\hat{\omega}}\, \underline{r}_i^B \right) \right) \qquad (4)$$

Translational and rotational dynamics of the load:

$$\begin{bmatrix} m_0 \underline{I} & 0 \\ 0 \quad 0 & I_0 \end{bmatrix} \begin{bmatrix} \ddot{\underline{X}}_0^B \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \underline{I} & \cdots & \underline{I} \\ \hat{\underline{r}}_1^B & \cdots & \hat{\underline{r}}_N^B \end{bmatrix} \begin{bmatrix} \underline{F}_1^B \\ \vdots \\ \underline{F}_N^B \end{bmatrix} \qquad (5)$$

m_0 = load mass

I_0 = load moment of inertia about axis normal to the plane of motion + passing through the load's CG

$\underline{F}_i^B = \underline{F}_i$ in frame B

- Solve Eq. (5) for $\ddot{\underline{X}}_0^B$ and substitute $\ddot{\underline{X}}_0^B$ into Eq. (4) to obtain an equation of the form:

$$\ddot{\underline{X}}_i = \underline{Q} + \underline{P}\,\underline{F}_i$$

- $\underline{Q}$ contains the forces applied by robots $j \neq i$
- $\underline{P}\,\underline{F}_i$ contains the force applied by robot i

$$\Rightarrow \underline{F}_i = \underline{P}^{-1}(\ddot{\underline{X}}_i - \underline{Q})$$

↳ Substitute into Eq. (1):

$$m_i \ddot{\underline{X}}_i = \underline{u}_i - \underline{P}^{-1}(\ddot{\underline{X}}_i - \underline{Q})$$

$$\Rightarrow \ddot{\underline{X}}_i = \underbrace{(m_i \underline{I} + \underline{P}^{-1})^{-1} \underline{P}^{-1} \underline{Q}}_{\underline{h}} + \underbrace{(m_i \underline{I} + \underline{P}^{-1})^{-1}}_{\underline{G}} \underline{u}_i$$

Controller design

V_{des} = target load speed

γ = target load direction

State vector of a robot: $\underline{x} = \begin{bmatrix} \underline{X}^T \\ \dot{\underline{X}}^T \end{bmatrix} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}$
(in frame I)

$\underline{R}_I^{I\gamma}$ = rotation matrix from frame I to frame $I\gamma$

$\underline{x}^* = \underline{R}_I^{I\gamma}\,\underline{x} = [x^*\ y^*\ \dot{x}^*\ \dot{y}^*]^T$ state vector in frame $I\gamma$

$\underline{u}^* = \underline{R}_I^{I\gamma}\,\underline{u} = \begin{bmatrix} u_1^* \\ u_2^* \end{bmatrix}$

u_1^* ← robot's actuating force in desired direction γ

u_2^* ← robot's actuating force normal to this direction

Robot dynamics in frame $I\gamma$:

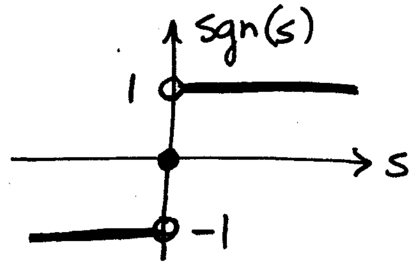
$$\underline{X}^* = \underline{h}^* + \underline{G}^*\underline{u}^* \qquad \underline{h}^* = \underline{R}_I^{I\gamma}\underline{h}, \quad \underline{G}^* = \underline{R}_I^{I\gamma}\underline{G}\,\underline{R}_{I\gamma}^{I}$$

- Design u_1^*, u_2^* as sliding mode controllers that drive all $\underline{x}^*(t)$ to a <u>sliding manifold</u> in the robot's state space in finite time and remain there afterward.
  - The robot exhibits a desired dynamical behavior when its state evolves along the manifold, s_i.

$s_1 = \dot{x}^* - V_{des} = 0$ ← regulates robot's speed along x_I^* axis

$s_2 = \dot{y}^* = 0$ ← regulates component of robot's velocity along the y_I^* axis to 0

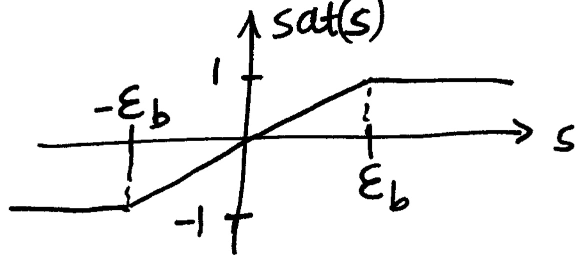
$$u_1^* = -k_1 \,\mathrm{sgn}(s_1)$$
$$u_2^* = -k_2 \,\mathrm{sgn}(s_2)$$



Signum function

k_1, k_2 must be high enough to stabilize the system on the sliding manifolds.

- To eliminate chattering on sliding manifolds, replace $\mathrm{sgn}(s_i)$ by the saturation function $\mathrm{sat}\left(\frac{s_i}{\varepsilon_{bi}}\right)$:



ε_b = boundary layer parameter

- Chattering: finite-frequency, finite-amplitude oscillations caused by high-freq. switching of a sl. mode controller exciting unmodeled dynamics in the closed loop.

Stability proof

1. Show that ω remains bounded under u_1^*, u_2^*.
   - Use the comparison lemma for scalar differential eq's: [Khalil, Nonlinear Systems]

Consider the equation $\dot{z} = f(t, z)$, $z(t_0) = z_0$, where $f(t, z)$ is continuous in t and locally Lipschitz in z for all $t \geq 0$ and all $z \in J \subset \mathbb{R}$.

Let $[t_0, T)$ (T could be ∞) be the maximal interval of existence of the solution $z(t)$, and suppose that $z \in J$ for all $t \in [t_0, T)$.

Let v be a continuous function for which:

$$\dot{v}(t) \leq f(t, v), \quad v(t_0) \leq z_0, \text{ where } v \in J \text{ for all } t \in [t_0, T).$$

[Note: If v is not differentiable, this could be replaced by its upper right hand derivative $D^+ v$.]

Then, $\boxed{v(t) \leq z(t)}$ for all $t \in [t_0, T)$.

- This lemma is applied to $v(t) = \frac{1}{2}\dot{\phi}(t)^2$, where $\phi = \theta - \gamma \Rightarrow \dot{\phi} = \dot{\theta} = \omega$. Thus, ω is bounded.
- Since ω is bounded, $\underline{h}$ is bounded because it is a function of ω and constant properties (load mass, dimensions), as well as forces applied by robots (which are bounded).

w is bounded $\Rightarrow \underline{h}$ is bounded since it is a function of w and constant properties (load's mass, geometry)

② Use Lyapunov stability analysis to show that the system is asymptotically stable, i.e. all state trajectories reach the intersection of s_1, s_2 in finite time and remain there for all time.

- Lyapunov functions: $V_1 = \frac{1}{2}s_1^2$, $V_2 = \frac{1}{2}s_2^2$

$$\dot{V}_1 = s_1 \dot{s}_1 = s_1 \ddot{x}^*$$

$$\dot{V}_2 = s_2 \dot{s}_2 = s_2 \ddot{y}^*$$

Show that $\dot{V}_1, \dot{V}_2 < 0$ when $|s_1|, |s_2| \neq 0$.

Implementation on differential-drive robots

$\dot{\theta}_R, \dot{\theta}_L$ = angular velocities of right + left wheels

τ_R, τ_L = actuation torques on wheels (control inputs)

Define: $\dot{\theta}_H = \frac{1}{2}(\dot{\theta}_R - \dot{\theta}_L)$, $\tau_H = \frac{1}{2}(\tau_R - \tau_L)$

$\dot{\theta}_V = \frac{1}{2}(\dot{\theta}_R + \dot{\theta}_L)$, $\tau_V = \frac{1}{2}(\tau_R + \tau_L)$

$$\begin{bmatrix} A-B & 0 \\ 0 & A+B \end{bmatrix} \begin{bmatrix} \ddot{\theta}_H \\ \ddot{\theta}_V \end{bmatrix} + \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} \dot{\theta}_H \\ \dot{\theta}_V \end{bmatrix} = \begin{bmatrix} \tau_H \\ \tau_V \end{bmatrix}$$

[Ivanjko et al, "Modelling of mobile robot dynamics," EUROSIM 2010]

- A, B depend on geometry + mass properties of robot
- K is the damping in the wheels.

$\Rightarrow$ 2 eq's: one governs robot's heading angle $\varphi \overset{\text{proportional to}}{\propto} (\theta_R - \theta_L)$
one governs robot's speed $V \propto (\dot{\theta}_R + \dot{\theta}_L)$

$$S_H = \varphi - \gamma \qquad S_V = V - V_{des}$$

$$\tau_H = -k_H \, \mathrm{sat}\left(\frac{S_H}{\varepsilon_{bH}}\right), \quad \tau_V = -k_V \, \mathrm{sat}\left(\frac{S_V}{\varepsilon_{bV}}\right)$$ sliding mode controllers for robot

Can control $\ddot{\theta}_H$, $\ddot{\theta}_V$ directly in actual robot (the motor accelerations) – requires measurements of wheel velocities + robot's heading.