GRAPH THEORY

A graph $G = (V, \mathcal{E})$ is made of a Vertex set $V = \{v_1, \dots, v_N\}$ and an Edge set $\mathcal{E} \subseteq [V^2] = \{(v_i, v_j)\},\ i \neq j$.

- **Undirected** graph: $(v_i, v_j) \in \mathcal{E} \Rightarrow (v_j, v_i) \in \mathcal{E}$
- **Directed** graph: $(v_i, v_j) \in \mathcal{E} \not\Rightarrow (v_j, v_i) \in \mathcal{E}$

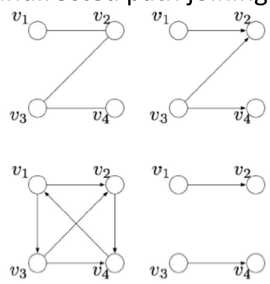
- **Degree** of a node = is the number of its neighbours (in-degree if directed graph)
- An <u>undirected</u> graph is **connected** if there exists a path joining any two vertexes in V.
- A <u>directed</u> graph is **strongly connected** if there exists a directed path joining any two vertexes in V. Or it's **weakly connected** if there exists an undirected path joining any two vertexes in V.

connected

weakly connected

strongly connected

disconnected

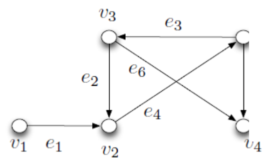


Def: <u>*Two graphs G1 and G2 are **isomorphic** if there exists a bijection $\varphi: V(G1) \rightarrow V(G2)$ such that if the vertexes x, y are adjacent in G1 the vertexes $\varphi(x), \varphi(y)$ are adjacent in G2.*</u>
Proposition: <u>*are **isomorphic** if there exists a permutation matrix P such that $P^T A(G1) P = A(G2)$*</u>

- **Adjacency Matrix** $A \in \mathbb{R}^{NxN}$:

  $A_{ij} = 0 \ if\ (v_j, v_i) \notin \mathcal{E}\ and\ A_{ij} = 1\ if\ (v_j, v_i) \in \mathcal{E}$

  - one can generalize to a positive weight $A_{ij} = w_i$
  - square and symmetric for undirected graph so $A = A^T$.



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

- **Degree Matrix $\boldsymbol{\Delta}$** $\in \mathbb{R}^{NxN}$: $D = diag(\sum_{j=1}^{N} A_{ij})$

$$\Delta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

- **Incidence matrix $\boldsymbol{E}$** $\in \mathbb{R}^{Nx|\mathcal{E}|}$ encode the incidence relationship

  $E_{ij} = -1\ if\ v_i\ is\ the\ tail\ of\ edge\ e_j$

  $E_{ij} = 1\ if\ v_i\ is\ the\ head\ of\ edge\ e_j$

  $E_{ij} = 0\ otherwise$

$$E = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 & -1 & 0 \end{bmatrix}$$

- **Laplacian Matrix $\boldsymbol{L}$** $\in \mathbb{R}^{NxN}$: $\boldsymbol{L = \Delta - A}$ or $\boldsymbol{L = EE^T}$

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}$$

Properties of the Laplacian:

- UNDIRECTED GRAPH: L is symmetric ($\mathbf{1}^T L = 0$), positive semi-definite ($L\mathbf{1} = 0$) → all the eigenvalues λ_i are <u>real</u> and <u>non-negative</u>: $0 \leq \lambda_1 \leq \dots \leq \lambda_N$

- **Def:** The graph is connected iff $\lambda_2(G) > 0 \iff rank(L) = N - 1$.
  λ_2 is the **connectivity eigenvalue** and **1** is the eigenvector associated to $\lambda_1 = 0$.
- Also $E^T \mathbf{1} = 0 \rightarrow rank(E) = N - 1$

CONSENSUS PROTOCOL

Problem: N agents with internal state x_i and internal dynamic for the state evolution $\dot{x}_i = u_i$. →Design the control inputs u_i so that all the states x_i agree on the same common value $\bar{x}$ by making use of only relative information w.r.t. the neighbours' state (decentralized approach).

$$\boxed{\lim_{t\to\infty} x_i(t) = \bar{x}}$$

A possible choice for u could be the sum of all the differences of the neighbours' states:

$$\boxed{u_i = \sum_{j\in\mathcal{N}_i} (x_j - x_i)}$$

• equivalent to $u = -Lx$ for all agents, and when closing the loop: $\dot{x} = -Lx$.

→ **Convergence** is related to the properties of the Laplacian (state-transition matrix of closed-loop dynamics).

- **UNDIRECTED GRAPH:** If the graph G is connected ($\lambda_2 > 0$) the consensus protocol converges to the average of the initial condition x_0:

$$\boxed{\lim_{t\to\infty} x(t) = \frac{(\mathbf{1}^T x_0)\mathbf{1}}{N}}$$

(proof: from the explicit closed loop dynamics x(t) when $\lambda_1 = 0$ second term ->0 if $\lambda_2 > 0$).

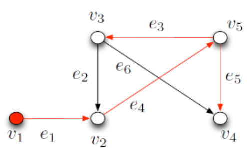
$$x(t) = \frac{(\mathbf{1}^T x_0)\mathbf{1}}{N} + \cancel{\sum_{i=2}^{N} u_i u_i^T e^{-\lambda_i t} x_0} \to 0$$

• consensus protocol makes the state $x(t) \to span(\mathbf{1}) = \{x | x_i = x_j\}$ (the null space of L) and the centroid of the states never changes over time $\mathbf{1}^T x(t) = \mathbf{1}^T x_0 = const$ (constant motion)

• λ_2 dictates **rate of convergence** of the consensus (⟷ rate of the asymptotic decay of the sum)
→ the more connected the graph, the faster the consensus convergence.

• If the graph is not connected, then the consensus will be achieved on each connected component (L is a block-diagonal matrix).

- **DIRECTED GRAPH:** (L is not symmetric)→ still $L\mathbf{1} = 0$ but $\mathbf{1}^T L \neq 0$



• **FACT 1:** the conditions for the consensus convergence $rank(L) = N - 1$ require the graph contains a **rooted out-branching** (a graph with no cycle and in which the root is connected with a directed path to all the other vertexes)
• **FACT 2:** (Gersgorin Theorem) L for directed graphs has all the eigenvalues with non-negative real part (and they cannot be imaginary pairs) $\Re(\lambda_i) \geq 0$

→In general, the consensus will not converge to the average of the initial condition

$$\boxed{\lim_{t\to\infty} x(t) = (q_1^T x_0) p_1 = (q_1^T x_0)\mathbf{1}}$$

For some $q_1 \neq 0, \lambda_1 = 0$ ($p_1 = \mathbf{1}$).
→In general, $q_1 \notin span(1)$.

• for a **balanced** directed graph (in-degree = out-degree.), it is also $\mathbf{1}^T L = 0$ (+ L**1**=0)→ we obtain the same limit analogously to the undirected graph case

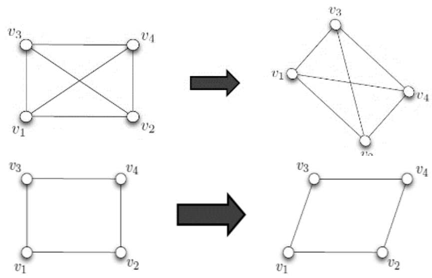
$$\boxed{\lim_{t\to\infty} x(t) = \frac{(\mathbf{1}^T x_0)\mathbf{1}}{N}}$$

Some remarks:

- We can take into account suitable gains $u_i = k_i(t) \sum_{j \in \mathcal{N}_i} (x_j - x_i)$
- One can generalize to a *stochastic settings*,
- or it is possible to consider *time-varying topologies* for the graph (e.g., considering the occlusion of visibility for robots);
- or we can consider more complex *linear* or *nonlinear dynamics*.
- It is possible to consider *time delays* and/or *asynchronous* communication

GRAPH RIGIDITY

Consider N agents and $M \leq N(N-1)/2$ pair-wise geometrical constraints (= edge). We can characterize the "flexibility". Ex: consider distance constraints



- If $M = N(N-1)/2$ (**complete graph**) the shape is determined up to a rototranslation in the plane (agents move as planar rigid body).
- If $M < N(N-1)/2$ (**not complete graph**) → depending on situation (edges) the shape is **preserved** (as complete graph) or **not**.

Bar-and-joint framework:

let $G = (V, \mathcal{E})$ and $p: V \to \mathbb{R}^d$ a function mapping each vertex to a point (a position associated to each node). Call $g_{ij}(p_i, p_j)$ a constraint function for each edge (i, j)

- In most cases, the constraint only depends on the relative positions/poses $g_{ij}(p_i - p_j)$. Let then $g_G = \{\dots g_{ij} \dots.\}$ be the *cumulative constraint* function over all the edges in G.

• A Framework is **rigid** if the only allowed motions satisfying the constraints are those of the complete graph K_N.

Or: A framework is **rigid** if there exists a neighbourhood U of p such that

$$\boxed{g_G^{-1}(g_G(p)) \cap \mathcal{U} = g_K^{-1}(g_K(p)) \cap \mathcal{U}}$$

(we need $N(N-1)/2$ edges, however, rigidity is often possible with a set of $2N-3$ edges properly placed). Then:

- we can solve **formation control** regulating the constraints→ each agent has to control its geometrical constraint (edges) to ensure that the desired global shape is realized (complete graph no needed).
- We can solve **relative localization** univocally from the measured value of the constraints. Each agent can only be at one specific location.

Two frameworks $(G, p_1), (G, p_2)$:

- are **equivalent** if they have the same constraints but not necessarily the same shape $g_G(p_1) = g_G(p_2)$,
- are **congruent** if they have the same constrains but also the same shape $g_K(p_1) = g_K(p_2)$

- a framework (G, p_1) is **globally rigid** if all the frameworks (G, p_2) are equivalent and also congruent to (G, p_1)
- A framework is **minimally rigid** if the removal of any edges yields a non-rigid framework.

Infinitesimal rigidity

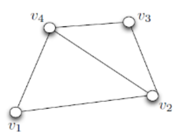
We want to find the instantaneous motions of $p(t)$ that preserves the constraints: $g_G\big(p(t)\big) = const$

$$\boxed{g_G\big(p(t)\big) = const \rightarrow \dot{g}_G\big(p(t)\big) = 0 \rightarrow \frac{\partial g_G(p)}{\partial p}\dot{p} = R_G(p)\dot{p} = 0}$$

To preserved constraints the **infinitesimal motions** consistent with the constraints are $\dot{\boldsymbol{p}} \in \mathbf{ker}\ (\boldsymbol{R_G}(\boldsymbol{p}))$ → a framework is **infinitesimally rigid** if $ker(R_G(p)) = ker(R_K(p))$ or the same $rank$ (same of complete graph)

- Infinitesimal rigidity implies rigidity, but NOT the opposite! (special alignment of agents can causes the rigidity matrix to lose rank).
- A point $\bar{p}$ is a **regular point** (= no special alignment) if $rank\big(R_G(\bar{p})\big) = \max\Big(rank\big(R_G(p)\big)\Big)$
- **Rigidity matrix** $\boxed{R_G(p) = \frac{\partial g_G(p)}{\partial p}}$ link between agent motion and constraint variations, Its null-space $\ker\big(R_G(p)\big)$ describes all the motions preserving the constraints.
  - Allows also to determine the minimum number of edges in a graph for being rigid: Let $rank\big(R_{K_N}(p)\big) = r$ → A framework is **rigid** if $rank\big(R_G(p)\big) = rank(R_{K_N}(p))$.

❖ **Distance constraints** in $\mathbb{R}^2$ of the form $g(p) = \big|\big|p_i - p_j\big|\big|^2$ the complete graph allows 3 collective motions: **2 translations** on the plane + **1 rotation**.



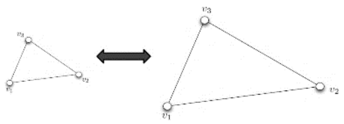
→ for a rigid graph $dim\ ker\ \big(R_G(p)\big) = 3,\ rank\big(R_G(p)\big) = 2N - 3 = 5$ so one need at least 5 edges.

In $\mathbb{R}^3$ we have vectors n_1, n_2 for planar translations along x and y, n_3 for rotation and a pivot point p^* → $dim\ ker\big(R_G(p)\big) = 6$ for a rigid graph. The constraint-preserving motions are the 3 translations and 3 rotations around an arbitrary p^*.

❖ **Bearing constraints**: we are constraining "relative angles" between pairs of agents

- Absolute bearing $\beta_{ij} = \frac{p_j - p_i}{||p_j - p_i||} \in S^{n-1}$ if the angles are expressed in a common frame
- Body-frame bearing $\beta_{ij} = R^i \frac{p_j - p_i}{||p_j - p_i||} \in S^{n-1}$ if the angles are expressed in the local frame of agent i.

Here, in case of absolute bearing the only allowed motions are the 2D translation and expansion/retraction (NO rotation since it would change the bearing angle!).



→ Same $rank$ as for the previous distance constraints, but different *kernel*!! (n_3 is different)

Rigidity is very important for **formation control** and **localization**:

- **Formation control:**
  Suppose that we have distance constraints and we want to stabilize the pose p of the agents to a desired pose p_d → $g_G(p) \rightarrow g_G(p_d) = g_d$. This means that we just care about the final shape, not where it is placed on the plane. → find a feedback controller which zeros the "constraint error" $g_d - g_G(p)$.

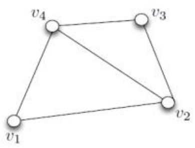
  If the framework is **rigid**, we are guaranteed that $g_d = g_G(p_d)$ implies **congruency** with the desired p_d. Define the error $e = \frac{1}{2}||g_d - g_G(p)||^2$ → it can be minimized by following its negative gradient, i.e.,

$$\boxed{\dot{p} = R_G(p)^T\left(g_d^T - g_G^T(p)\right)}$$

  Which is a **decentralized controller** (decentralized structure of the rigidity matrix) → each agent can regulate itself to achieve the desired shape.

  • Recall the example

$$\begin{array}{cc} & \begin{array}{cccc} & & \text{Agent 3} & \end{array} \\ R_G(p) = & \begin{bmatrix} p_1^T - p_2^T & p_2^T - p_1^T & 0 & 0 \\ p_1^T - p_4^T & 0 & 0 & p_4^T - p_1^T \\ 0 & p_2^T - p_3^T & p_3^T - p_2^T & 0 \\ 0 & p_2^T - p_4^T & 0 & p_4^T - p_2^T \\ 0 & 0 & p_3^T - p_4^T & p_4^T - p_3^T \end{bmatrix} \end{array}$$



  • The i-th column of $R_G(p)$ (associated to agent i) only depends on p_i and p_j, $j \in \mathcal{N}_i$

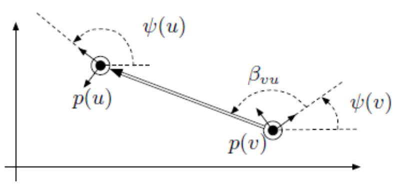
- **Localization problem**:
  assume N agents can measure the relative distances and we want to localize the agent positions in some common frame. Assume that (G, p) is **rigid** → if the estimate $\hat{p}$ agrees with the measurements i.e., if $g_G(\hat{p}) = g_G(p)$→ $\hat{p}$ can only be a rigid roto-translation of the real p and represents a correct localization of the agents in "some frame".

  The problem can be solved as before, define $e = \frac{1}{2}||g_G(p) - g_G(\hat{p})||^2$ Therefore, an update law for $\hat{p}$ is:

$$\boxed{\dot{\hat{p}} = R_G^T(\hat{p})(g_G(p) - g_G(\hat{p}))}$$

  REMARK:
  It is also possible to remove the **translational ambiguity** enforcing additional constraints on the estimated positions $\hat{p}$ (e.g., fixing $\hat{p}_1 = 0$ the all the remaining $\hat{p}_i$ represent relative positions w.r.t. agent 1).
  Similarly, removes the **rotational ambiguity**: one could fix the orientation of the common frame by fixing the direction of one of its edges connecting two agents (one can enforce $\hat{p}_1 - \hat{p}_k = p_1 - p_k$).
  →All these constraints can be embedded in a single cost function, which lead to a (decentralized) update law $\dot{p}_i$.

- Consider the case of **body-frame bearings**:
  We consider a planar problem in which the vertexes of graph are mapped to a pose (p_i, ϕ_i). These form a directed graph since we have directed measurement between the agents. Moreover, each node consists of a position on the plane and an orientation w.r.t. some global frame → $R_G \in R^{|\mathcal{E}| x 3N}$



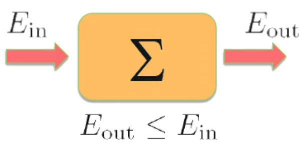
  For the complete graph, there are 4 allowed motions that are: 2D translation, expansion/contraction and rotation around a pivot point p^*. → therefore, $rank(R_G) = 3N - 4$, → we need at least 3N-4 edges properly placed for the frame to be bearing rigid.

PASSIVITY

Passivity is a I/O property of a dynamical system, related to the concept of "energy" flow inside a system.

Def: A memoryless static functions $y = h(u)$ is said to be **passive** if

- Power flowing into the system is never negative
- The system does not produce energy (can only absorb and dissipate)



Consider a generic **nonlinear** system $\begin{cases} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{cases}$

The system is **dissipative** if there exists a continuous (differentiable) lower bounded function of the state (*storage function*) $V(x) \in C^1$ and a function of the I/O pair (*supply rate*) $\omega(u, y)$ such that

$$\boxed{\begin{cases} V(x(t)) - V(x(t_0)) & \leq \int_{t_0}^{t} w(u(s), y(s))\mathrm{d}s \\ \dot{V}(x(t)) & \leq w(u(t), y(t)) \end{cases}}$$

- When the supply rate is $\omega(u, y) = y^T u - \delta u^T u - \epsilon y^T y, \quad \delta, \epsilon \geq 0$ the system is said **passive** w.r.t. to ω and V. In particular,
  - **Lossless** if $\delta = 0, \epsilon = 0, and\ \dot{V} = y^T u$
  - **input strictly passive** (ISP) if $\delta > 0$
  - **output strictly passive** (OSP) $\epsilon > 0$
  - **very strictly passive** (VSP) $\epsilon > 0, \delta > 0$

- If there exists a positive definite function $S(x): R^n \to R^+$ such that $\dot{V}(x) \leq y^T u - S(x)$ then the system is said **strictly passive**, and S(x) is the **dissipation rate.**

The **storage function** $V(x)$ represents the internal stored energy. The **supply rate** $y^T u$ is the power (energy flow) exchanged with the external world.
The basic passivity condition can be interpreted as:

$$V(x(t)) \leq V(x(t_0)) + \int_{t_0}^{t} y^T(s)u(s)\mathrm{d}s$$

Current energy is at most equal to the initial energy + supplied energy from outside

equivalent to "no internal generation of energy".

Another interpretation is that **exctractable energy** (net of the energy supplied from outside) is bounded from below by the **initial stored energy:**

$$\int_{t_0}^{t} y^T(s)u(s)\mathrm{d}s \geq V(x(t)) - V(x(t_0)) \boxed{\geq -V(x(t_0)) \geq -c^2,} \quad c \in \mathbb{R}$$

This yields another **passivity** condition:

$$\boxed{\int_{t_0}^{t} y^T(\tau)u(\tau)\mathrm{d}\tau \geq -c^2, \ c \in \mathbb{R}, \quad \forall u, \forall t \geq t_0}$$

(useful because no formal need of a storage function)

- **Stability** ↔linked to **Lyapunov stability**.

Lyapunov: • Given a system $\dot{x} = f(x) \quad f(0) = 0$ (■)

the equilibrium $x = 0$ is
- Stable if $\forall \epsilon > 0 \exists \delta(\epsilon) > 0 \mid \|x(t_0)\| \leq \delta \Rightarrow \|x(t)\| \leq \epsilon, \quad \forall t \geq t_0$
- Unstable if it is not stable
- Asymptotically stable if stable and $\|x(t_0)\| \leq \delta \Rightarrow \lim_{t\to\infty} x(t) = 0$

• If there exists a $V(x)$ such that
- $\dot{V}(x) \leq 0 \text{ in } D$ then the system is stable
- $\dot{V}(x) < 0 \text{ in } D - \{0\}$ then the system is (locally) asympt. stable (LAS)
- If $V(x)$ is radially unbounded, i.e., $D = \mathbb{R}^n$ and $\|x\| \to \infty \Rightarrow V(x) \to \infty$, and it still holds $\dot{V}(x) < 0 \text{ in } D - \{0\}$ the system is globally asympt. Stable (GAS)

LaSalle th: • LaSalle Th.:The system will converge towards M, the largest invariant set in $S = \{x \in D \mid \dot{V}(x) = 0\}$

• If $M = \{0\}$, i.e., only $x(t) \equiv 0$ can stay identically in S, then the system is LAS (GAS)

Passivity: given a system $\begin{cases} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{cases}$ and a storage function $V(x)$ such that $\dot{V}(x) \leq y^T u$

If we have that $V(0) = 0 \rightarrow$ V(x) is also a Lyapunov candidate around 0. In this case:
- if $u = 0 \rightarrow \dot{V} \leq 0$ → system is **stable**
- if $y = 0 \rightarrow \dot{V} \leq 0$ → the **zero-dynamics** of the system **is stable.**

It can be enforced by:

- **output feedback:** The system can be easily stabilized by a **static output feedback** (like $u = -ky$).

  $$\boxed{u = -\phi(y), \quad y^T\phi(y) > 0 \, \forall y \neq 0}$$

  we obtain:
  - Non increasing storage function $\dot{V} \leq -y^T\Phi(y) \leq 0$ → state trajectories bounded
  - The output (velocity) converges to 0:→ $y = h(x) = 0$

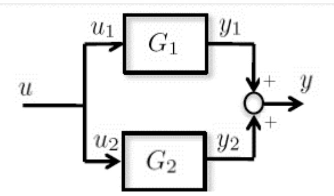
- if the system is **zero-state observable** **(LaSalle):** $y(t) = h\big(x(t)\big) = 0 \;\Rightarrow x(t) = 0$ so zeroing the output implies zeroing the complete state → $u = -\Phi(y)$ provides local asymptotic stability **LAS**. GAS if V(x) is also radially unbounded.

- **Finding the "correct" output** : consider the state evolution $\dot{x} = f(x) + g(x)u$, assume we can find a $V(x)$ such that $\frac{\partial V}{\partial x} f(x) \leq 0$ i.e., a **stable free evolution** $u = 0$. → the system is **passive** w.r.t. the output $y = \left[\frac{\partial V}{\partial x} g(x)\right]^T$ → the feedback $u = -ky = -k\left[\frac{\partial V}{\partial x} g(x)\right]^T$ makes the system **LAS** (GAS).

- **Modular property:** proper interconnections of passive systems are again passive!
  So we can consider subnetworks, make them passive and interconnect them resulting in a passive system, stable etc.
  Consider two passive systems with proper I/O dimensions and storage functions $V_1(x_1), V_2(x_2)$

$$\begin{cases} \dot{x}_1 &= f_1(x_1) + g_1(x_1)u_1 \\ y_1 &= h_1(x_1) \end{cases} \qquad \begin{cases} \dot{x}_2 &= f_2(x_2) + g_2(x_2)u_2 \\ y_2 &= h_2(x_2) \end{cases}$$

  - **Parallel interconnection:**



set $u_1 = u_2 = u$ and $y = y_1 + y_2$.
Let $x = (x_1, x_2)$ and let $V(x) = V_1 + V_2$. Then:

$$\boxed{\dot{V} = \dot{V}_1 + \dot{V}_2 \leq y_1^T u_1 + y_2^T u_2 = (y_1 + y_2)^T u = y^T u}$$

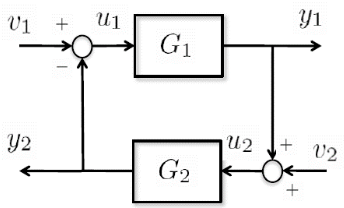
The new system is passive w.r.t. the pair $(y_1 + y_2, u) = (y, u)$.

  - **Feedback interconnection:**

They can be interconnected as:

$$\begin{cases} u_1 &= \pm y_2 + v_1 \\ u_2 &= \mp y_1 + v_2 \end{cases}$$

New (optional) inputs



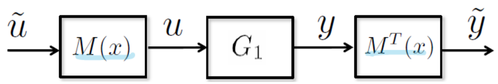
→ the interconnected system is passive with storage function $V(x)$ w.r.t. the (composed) input/output pair $([y_1^T \;\; y_2^T]^T, [v_1^T \;\; v_2^T]^T)$

In this case:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 & \pm 1 \\ \mp 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

The coupling matrix is ***skew-symmetric***→ This is a fundamental property that allows to retain passivity of the composed system (energy can only be transfer from a system to the other).
(this is an example of a power-preserving interconnection)

- **pre-post multiplication:**



  Assume G_1 is a passive system with storage function $V(x)$ w.r.t. the pair (u, y). Let $M(x)$ be a (possibly state-dependent) matrix, and let $u = M(x)\,\tilde{u}$ and $\tilde{y} = M^T(x)y$

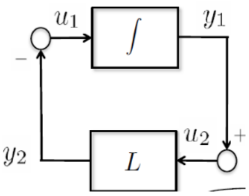
→ Passivity is preserved by a pre-multiplication of the input by $M(x)$ and a postmultiplication of the output $M(x)^T$.

Review of consensus protocol:

Take a passive (lossless) system: *single integrators* $\Sigma : \begin{cases} \dot{x} &= u_1 \\ y_1 &= x \end{cases}$

And consider a static function $y_2(u_2) = L\,u_2$ → this is a passive static function $u_2^T y_2 = u_2^T L u_2 \geq 0$.

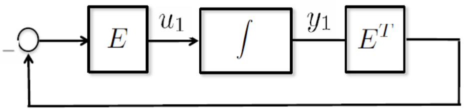
Interconnect these two passive systems by means of a "feedback interconnection" $\begin{cases} u_2 &= y_1 \\ u_1 &= -y_2 \end{cases}$



→The resulting system is passive, and it's the consensus closed-loop dynamics

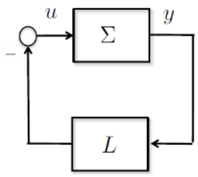
$$\dot{x} = -Lx$$

Recall that $L = EE^T$ then we can also consider it as:



Since the single integrator is passive and a pre-/post-multiplication preserves passivity, we are just closing the loop of a passive with a negative unitary output feedback.

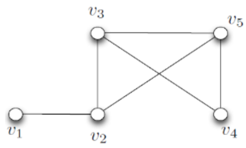
This can be also extend to any passive **nonlinear system**: $\Sigma : \begin{cases} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{cases}$



Then keeps being a (closed-loop) passive system.

Edge Laplacian

It is the matrix: $\boxed{L_\mathcal{E} = E^T E}$ and can be seen as an edge adjacency matrix, where 2 edges are adjacent if the share a common vertex (every edge is considered adjacent to itself so I have 2 on the diagonal) (recall that $L = EE^T$, where E is the incidence matrix of the graph) → the nonzero eigenvalues of L and $L_\mathcal{E}$ are the same.



consider a graph with states defined over the edges $x_\mathcal{E}(t) = E^T x(t)$ rather than over the vertexes (as in the standard Consensus). We can define the **edge agreement protocol:**

$$\boxed{\dot{x}_\mathcal{E}(t) = -L_\mathcal{E} x_\mathcal{E}(t)}$$

In this case, agreement is obtained when $x_\mathcal{E}(t) = 0$ → when the graph is **connected.**

Moreover, if graph $\mathcal{G}$ contains a spanning tree $\mathcal{G}_\tau$ → the incidence matrix can be decomposed as

$$E = [E_\tau \;\; E_c]$$

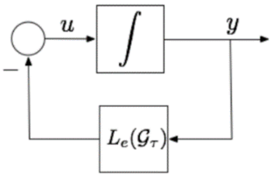
(E_τ for the spanning tree and E_c for the remaining edges) Accordingly, the Edge Laplacian can be decomposed as

$$L_\mathcal{E} = \begin{bmatrix} L_{\mathcal{E}_\tau} & E_\tau^T E_c \\ E_c^T E_\tau & L_{\mathcal{E}_c} \end{bmatrix}$$

- the Edge Laplacian of a spanning tree $L_{\mathcal{E}_\tau}$ is always positive definite → we can find a matrix R such that $L_\mathcal{E} = R^T L_{\mathcal{E}_\tau} R$.
- Furthermore, the edge agreement protocol can be reduced by considering the restriction over the spanning tree $\dot{x}_\tau(t) = -L_{\mathcal{E}_\tau} R R^T x_\tau(t)$

Passivity and the Edge Laplacian

If we interconnect an integrator with an edge Laplacian law over a spanning tree, then we get a strictly passive system.



Storage Function

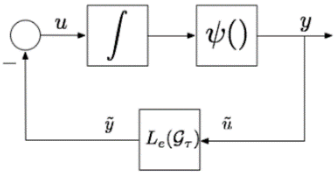
$$V(x) = \frac{1}{2}x(t)^T x(t)$$

A strictly passive system!

Positive-definite

$$u^T y = \dot{V} + y^T L_e(\mathcal{G}_\tau) y$$

So we can apply the same thing but with a **nonlinear function**: if the system is passive it can be stabilized through $u = -\Phi(y)$



Assume

$$\exists V(x) \text{ s.t. } \frac{\partial V(x)}{\partial X} f(x) < 0$$

➡ "forward path" is passive w.r.t output

Let

$$\phi(v) = \lambda_2(\mathcal{G})v$$

$$\tilde{u}^T \tilde{y} \geq \tilde{u}^T \phi(\tilde{u}) = \lambda_2(\mathcal{G})\tilde{u}^T \tilde{u} > 0$$

➡ The closed loop map is asymptotically stable!

In the **kuromoto model** we consider a multi-agent system of n coupled oscillators interacting over a network G. The state dynamics is:

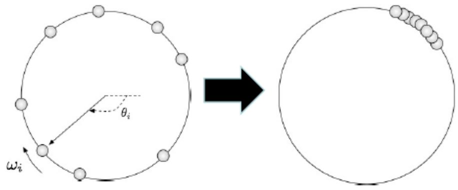
$$\dot{\theta}_i(t) = k \sum_{j \in N(i)} \sin(\theta_j(t) - \theta_i(t)), \quad i = 1, 2, \ldots, n.$$

We can see it also as

$$\dot{\theta}(t) = -\frac{K}{n_{\text{oscillators}}} E(\mathcal{G}) \sin(E(\mathcal{G})^T \theta(t))$$

Theorem: for any arbitrary connected graph G, coupling strength K>0 and almost all initial conditions in $(-\pi, \pi)^n$, the Kuramoto model will synchronize.

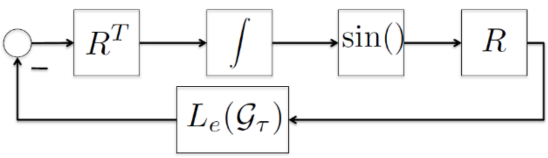
Moreover the rate of approach to synchronization is bounded by $\frac{2K}{\pi n}\lambda_2(G)$



In this we have that the nonlinear function is the sine. We can transform to edge states: $\theta_e(t) = E(\mathcal{G})^T \theta(t)$

Then the system will be:

$$\dot{\theta}_e(t) = -kR^T L_e(\mathcal{G}_\tau) R \sin(\theta_e(t))$$



Strictly Passive Element

← pre/post multiplication does not change passivity properties of forward loop → <u>using passivity and edge Laplacian we can say that the synchronization state is asymptotically stable.</u>

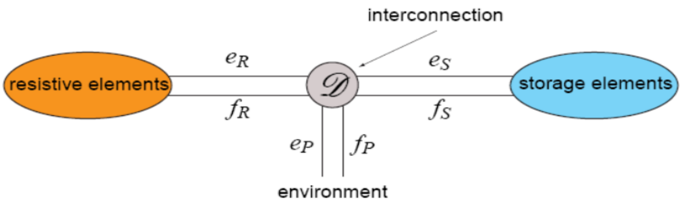
$$V(\theta_e(t)) = \mathbf{1}^T(\mathbf{1} - \cos(\theta_e(t))) \qquad \psi(\theta_e(t)) = \sin(\theta_e(t))$$

Port-Hamiltonian Systems

PHS are a powerful way to model physical systems which deals with the concept of Energy storage and Energy flows → The Dynamical behavior comes from the exchange of energy. **The power flows define the internal model structure.**

→ PHS ↔ strong link with passivity but focus on the structure behind passive systems.

Passive systems are made of **power-preserving interconnection** (modularity) that either: Store energy, Dissipate energy and Exchange energy (internally or with the external world) through power ports.
The Role of energy and the interconnections between subsystems provide the basis for various control techniques (Easily address complex nonlinear systems, especially when related to real "physical" ones).



- A set of energy storage elements (with their power ports (e_s, f_s))
- A set of resistive/dissipative elements (with their power ports (e_R, f_R))
- A set of "external world" power-ports (with their power ports (e_P, f_P))
- An internal **power-preserving interconnection** $\boldsymbol{D}$ (**Dirac structure**) , among the internal power ports ("pattern" of energy flow)

All the ports (included the "external ones"(e_P, f_P)) are constrained to belong to the Dirac structure $f_S, e_S, f_R, e_R, f_P, e_P) \in D$. That is $e_S^T f_s + e_R^T f_R + e_P^T f_P = 0$. → the **implicit definition of a Port-Hamiltonian system**

$$\left(-\dot{x},\ \frac{\partial H}{\partial x},\ f_R,\ e_R,\ f_P,\ e_P\right) \in \mathcal{D}$$

which implies $\dot{H} = e_R^T f_R + e_P^T f_P \leq e_P^T f_P$ as $e_R^T f_R \leq 0$

- In **nonlinear passive systems** can be rewritten as

$$\begin{cases} \dot{x} &= [J(x) - R(x)]\dfrac{\partial H}{\partial x} + g(x)u, \quad J(x) = -J^T(x),\ R(x) \geq 0 \\ y &= g^T(x)\dfrac{\partial H}{\partial x} \end{cases}$$

←**explicit Port-Hamiltonian form** (input-state-output) (when no algebraic constraints)

And the **passivity** condition naturally embedded in the same structure:

$$\dot{H} = -\frac{\partial H^T}{\partial x}R(x)\frac{\partial H}{\partial x} + \frac{\partial H^T}{\partial x}g(x)u \leq y^T u$$

- $H(x)$ = energy stored by the system,
- $R(x)$ = internal dissipation
- $J(x)$ = internal power-preserving interconnection among different components
- (u, y)= power-port, allowing energy exchange (in/out) with the external world

- The total energy of a PHS is called Hamiltonian.
- If the Hamiltonian is bounded from below, a PHS is passive w.r.t. its external ports. → $H(x) \geq c$

In the **linear time-invariant** case $\begin{cases} \dot{x} &= Ax + Bu \\ y &= Cx \end{cases}$

passivity implies existence of a storage function $H(x) = \frac{1}{2}x^T Q x,\ \ Q = Q^T \geq 0$ such that $A^T Q + QA \leq 0$ and $C = B^T Q$. Then we can rewrite

$$\boxed{\begin{cases} \dot{x} &= (J - R)Qx + Bu, \quad J = -J^T, \quad R = R^T \geq 0 \\ y &= B^T Q x \end{cases}}$$

And the energy balance $\boxed{\dot{H} = -x^T QRQx + x^T QBu \leq y^T u}$

Example: **Mass-spring-damper** $m\ddot{x} + b\dot{x} + kx = f$

we know the system is passive w.r.t. (u, y) with $u = f, y = \dot{x}$ and storage function $V = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2$, $\dot{V} \leq yu$. *But why is passive*? Let's see the internal structure:

it is composed of 2 components: kinetic energy K + elastic energy V

$$\mathcal{K}: \begin{cases} \dot{p} &= f_p \\ v_p &= \frac{\partial K}{\partial p} = \frac{p}{m} (= \dot{x}) \end{cases}$$

Kinetic energy storing

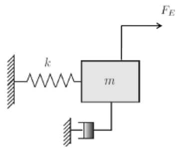
$$\mathcal{V}: \begin{cases} \dot{x} &= v_x \\ f_x &= \frac{\partial V}{\partial x} = kx \end{cases}$$

Potential energy storing

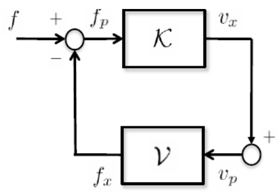
Both are integrators with linear outputs and so are passive w.r.t. $(v_p, f_p), (v_x, f_x)$.

Their power-preserving interconnection is a feedback interconnection (thus, preserves passivity): $v_x = v_p m\ f_p = -f_x + f$ the resulting system is

$$\begin{cases} \dot{x} &= [J(x) - R(x)]\frac{\partial H}{\partial x} + g(x)u, \quad J(x) = -J^T(x),\ R(x) \geq 0 \\ y &= g^T(x)\frac{\partial H}{\partial x} \end{cases}$$



specializes into $J(x) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},\ R(x) = \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix},\ g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

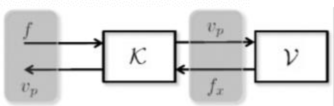


$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \left(\underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\text{Skew-symmetric}} - \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix}}_{\text{Positive semi-def.}} \right) \begin{bmatrix} \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f$$

where $H(x, p) = K(p) + V(x)$ is the Hamiltonian (total energy). The energy balance:

$$\dot{H} = -\underbrace{\begin{bmatrix} \frac{\partial H^T}{\partial x} & \frac{\partial H^T}{\partial p} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial p} \end{bmatrix}}_{\leq 0} + \begin{bmatrix} \frac{\partial H^T}{\partial x} & \frac{\partial H^T}{\partial p} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} f \leq \frac{\partial H^T}{\partial p} f = f^T v_p$$

We find again passivity w.r.t. (f, v_p).



The subsets K and V exchange energy in a *power-preserving way* (skew-symmetric matrix J(x)) → no energy is created or destroyed.

The total energy H can vary because of the exchange energy with the "external world" through the **power-port** (f, v_p), or decrease because of internal dissipation because of b.

- Any **mechanical system** described by the **Euler-Lagrange** equations can be recast in PHS:
  From $L = K(q,\dot q) - V(q)$, (kinetic + potential) by a change coord: $(q,\dot q) \to (q,p)$ where $p = M(q)\dot p$ → now the Hamiltonian is: $H(q,p) = K(q,p) + V(q)$. The Euler-Lagrange equations for the system are

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot q}(q,\dot q)\right) - \frac{\partial L}{\partial q}(q,\dot q) = \tau$$

  that can be rewritten as follows since $p = \partial L/\partial \dot q = \partial K/\partial \dot q$:

$$\begin{cases} \dot q &= \dfrac{\partial H}{\partial p} \\ \dot p &= -\dfrac{\partial H}{\partial q} + \tau \end{cases} \Rightarrow \left\{ \begin{bmatrix} \dot q \\ \dot p \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau \right. \qquad \boxed{\dot H = \frac{\partial H^T}{\partial p}\tau = \dot q^T \tau}$$

If $H(q,p)$ is bounded from below, the system is **passive** w.r.t. the **power port** $(\dot q, \tau)$.

- **Modularity**: Proper interconnection (through a Dirac structure D) of PHS are again PHS with

  - Hamiltonian $H = H_1 + \dots H_k$
  - State manifold $\mathcal{M} = \mathcal{M}_1 \times \dots \mathcal{M}_k$
  - Dirac structure $\mathcal{D} = \mathcal{D}_1, \dots, \mathcal{D}_k, \mathcal{D}_I$

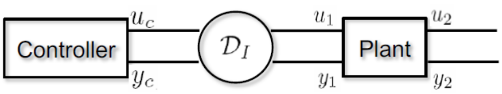
  - This allows for modularity and scalability.

Control of PHS

$$\begin{cases} \dot x &= [J(x) - R(x)]\dfrac{\partial H}{\partial x} + g(x)u \\ y &= g^T(x)\dfrac{\partial H}{\partial x} \end{cases} \longleftrightarrow \begin{cases} \dot x &= f(x) + g(x)u \\ y &= h(x) \end{cases}$$

Still a nonlinear dynamical system → so we can consider the usual control techniques. However, in closed-loop, we want to retain and to exploit the PHS structure! Then assume plant and controller in PHS form interconnected through a suitable D_1

$$\begin{cases} \dot x &= [J(x) - R(x)]\dfrac{\partial H}{\partial x} + g(x)u \\ y &= g^T(x)\dfrac{\partial H}{\partial x} \end{cases} \qquad \begin{cases} \dot x_c &= (J_c(x_c) - R_c(x_c))\dfrac{\partial H_c}{\partial x_c} + g_c(x_c)u_c \\ y_c &= g_c^T(x_c)\dfrac{\partial H_c}{\partial x_c} \end{cases}$$



where we split the plant port (u,y) into (u_1,y_1) for the interconnection with the controller and (u_2,y_2) with external. Moreover assume $H(x)$ is bounded from below→ with $u = -ky,\ \ k > 0$ will yield a (asympt.) stable closed-loop system ("damping injection").

We will cover three important subclasses of the fundamental **Energy Shaping problem:**

1. Energy Transfer Control
2. Energy Balancing
3. Interconnection and Damping assignment

1. Energy Transfer Control

Consider two PHS
$$\begin{cases} \dot{x}_1 &= J_1(x_1)\dfrac{\partial H_1}{\partial x_1} + g_1(x_1)u_1 \\ y_1 &= g_1^T(x_1)\dfrac{\partial H_1}{\partial x_1} \end{cases} \qquad \begin{cases} \dot{x}_2 &= J_2(x_2)\dfrac{\partial H_2}{\partial x_2} + g_2(x_2)u_2 \\ y_2 &= g_2^T(x_2)\dfrac{\partial H_2}{\partial x_2} \end{cases}$$

we want to transfer energy from a PHS to the other in a lossless way. → this can be done by means of state-modulated power preserving interconnection:

$$\boxed{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -\alpha y_1(x_1)y_2^T(x_2) \\ \alpha y_2(x_2)y_1^T(x_1) & 0 \end{bmatrix}}_{\text{Skew-symmetric}} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \alpha \in \mathbb{R}}$$

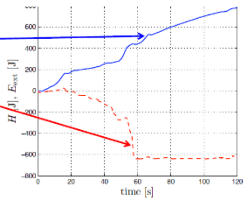
Then $H(x_1, x_2) = H(x_1) + H(x_2)$ constant. $\rightarrow \dot{H}(x_1, x_2) = 0$ The total energy doesn't change, but the individual energies may increase/decrease depending on the parameter α (no energy transfer with $\alpha = 0$).

$$\dot{H}_1(x_1) = -\alpha\|y_1\|^2\|y_2\|^2 \quad \dot{H}_2(x_2) = \alpha\|y_1\|^2\|y_2\|^2$$

NB: in a PHS there is an inherent **passivity margin** due to the internal dissipation:
from the usual PHS representation we know it is passive since $\dot{H} = \cdots \leq y^T u \rightarrow$ integral form

$$\underbrace{H(t) - H(t_0)}_{E_{in}} = \underbrace{\int_{t_0}^{t} y^T u \, d\tau}_{E_{ext}} \underbrace{\int_{t_0}^{t} \frac{\partial H^T}{\partial x} R(x) \frac{\partial H}{\partial x} d\tau}_{\leq 0}$$

dissipation



over time $E_{in} \leq E_{ext}$ ← The gap between them is because of the integral of the **dissipation term.** However, we would to have $E_{in} = E_{ext}$ to ensure lossless energy balance.
→ Dissipation term: **passivity margin** of the system.

IDEA: store back the dissipated energy and use it to passively implement whatever action w (without violating passivity).
Which is the basis of the **Energy Tank**:

Energy Tank: an atomic energy storing element with state $x_t \in R$ and energy function $T(x_t) = \frac{1}{2}x_t^2 \geq 0$

We can use it for
$$\boxed{\begin{cases} \dot{x}_t &= u_t \\ y_t &= \dfrac{\partial T}{\partial x_t}(= x_t) \end{cases}}$$

We want to exploit the tank for storing back the natural dissipation of a PHS and so using it for implementing some actions (this tank-based action will necessarily meet the passivity constraint!)

Let $D(x) = \frac{\partial H^T}{\partial x} R(x) \frac{\partial H}{\partial x}$ be the **dissipation rate** of the PHS. And choose $u_t = \frac{1}{x_t} D(x) + \tilde{u}_t$ in the Tank dynamics, then

$$\dot{T}(x_t) = x_t \left(\frac{1}{x_t} D(x) + \tilde{u}_t \right) = D(x) + x_t \tilde{u}_t$$

In order to exploit this stored energy to implement an action on the PHS system, we must design a suitable **(power-preserving) interconnection** among the PHS and Tank element (this preserve passivity by construction):

Assume we want to implement the action $w \in R^m$ on the PHS

$$\begin{cases} \dot{x} &= [J(x) - R(x)]\dfrac{\partial H}{\partial x} + g(x)u \\ y &= g^T(x)\dfrac{\partial H}{\partial x} \end{cases} \quad \mathcal{D}_I \quad \begin{cases} \dot{x}_t &= \dfrac{1}{x_t}D(x) + \tilde{u}_t \\ y_t &= x_t \end{cases}$$

then:

$$\begin{bmatrix} u \\ \tilde{u}_t \end{bmatrix} = \begin{bmatrix} 0 & \dfrac{w}{x_t} \\ -\dfrac{w^T}{x_t} & 0 \end{bmatrix} \begin{bmatrix} y \\ y_t \end{bmatrix}$$

← since the coupling is skew-symmetric no energy is created/destroyed.

In this way

Fact 1: action w is correctly implemented on the original PHS $\quad \dot{x} = [J(x) - R(x)]\dfrac{\partial H}{\partial x} + g(x)w$

Fact 2: the composite system is (altogether) a **passive (lossless) system** whatever the expression of w

$$\begin{bmatrix} \dot{x} \\ \dot{x}_t \end{bmatrix} = \left(\begin{bmatrix} J(x) & \dfrac{w}{x_t} \\ -\dfrac{w^T}{x_t} & 0 \end{bmatrix} - \begin{bmatrix} R(x) & 0 \\ -\dfrac{1}{x_t}\dfrac{\partial \mathcal{H}^T}{\partial x}R(x) & 0 \end{bmatrix} \right) \begin{bmatrix} \dfrac{\partial \mathcal{H}}{\partial x} \\ \dfrac{\partial \mathcal{H}}{\partial x_t} \end{bmatrix}$$

$$\dot{\mathcal{H}} = -\frac{\partial \mathcal{H}^T}{\partial x}R(x)\frac{\partial \mathcal{H}}{\partial x} + \frac{\partial \mathcal{H}^T}{\partial x_t}\frac{1}{x_t}\frac{\partial \mathcal{H}}{\partial x}R(x)\frac{\partial \mathcal{H}}{\partial x} = 0$$

Fact 3: Singularity when $x_t = 0$ → represents the impossibility of passively perform the desired action w.

However, one can consider a switching parameter $\alpha(t)$ and implement $\alpha(t)w$ instead of w (Idea is if cannot implement w wait the tank gets replenished).

In fact, the Tank is:

- Continuously refilled due to the dissipation $D(x)$ (I)
- Possibly refilled by the action w (II)

$$\dot{x}_t = \boxed{\frac{1}{x_t}D(x)} - \boxed{\frac{w^T}{x_t}g^T(x)\frac{\partial H}{\partial x}}$$

Moreover we can choose any $x_t(t_0) > 0$ → we have complete freedom in choosing the initial amount of energy in the tank $T(x_t(t_0))$

- **Passivity**: bounded amount of extractable energy, but for <u>whatever initial energy</u> in the system (only needs to be <u>finite</u>)

2. Energy balancing

In case $H(x)$ has a minimum at $x = 0 \rightarrow$ by applying the output feedback $u = -ky$, and assume $R(x) > 0$ and/or zero-state observability then $H(x)$ will asymptotically reach its minimum at $x = 0$.

If we want stabilization at a different x^****?*** **Energy shaping passivity:**

Given a passive system with storage function $H(x)$ → find a control action $u = \beta(x) + v$ which enforces in closed-loop passivity of the pair (v, y) w.r.t. the new (shaped) storage function $H_d(x) = H(x) + H_c(x)$:

$$\boxed{H_d(x(t)) - H_d(x(t_0)) \leq \int_{t_0}^{t} v^T(\tau)y(\tau)\mathrm{d}\tau}$$

$H_d(x)$ will encode the desired behaviour, having a minimum at $x = x^*$.

and along the system trajectories $-\int_{t_0}^{t} \beta^T(x(\tau))y(\tau)\mathrm{d}\tau = H_c(x(t)) + k$
then we speak about "**energy balancing**": the integral term is the energy supplied by the controller to the plant, represented by the state function $H_c(x)$; this energy modifies the total energy (of the closed-loop) into $H_d(x) = H(x) + H_c(x)$ = original energy + energy supplied by the controller.

Furthermore, $H_d(x)$ is lower-bounded and if $H_d(x)$ has a minimum at $x = x^*$, then the system can be stabilized by the usual $v = -ky$.

• Energy Balancing with a **passive (PHS) controller**

- Energy provided by the controller is limited (passivity)
- Cannot shape $H(x)$ in those coordinates affected by internal dissipation
- Mechanical systems: can shape the potential energy, not the kinetic energy (dissipation in kinetic energy)

• Passive PHS controller + feedback interconnection + PHS plant

- Energy Balancing can be solved by using the "**Casimir**" method
- Look for **motion invariants** of the closed-loop system in the form $C(x, x_c) = x_c - F(x)$
- Only determined by the **internal structure** $J(x)$ and $R(x)$
- "Easy" conditions, but **dissipation obstacle**

• Abandon the Casimir method and passive PHS controller

- Can overcome the dissipation obstacle with a state-modulated interconnection
- Must solve an additional PDE

• Look for a feedback $u = \beta(x) + v$ which also assigns desired and $J_d(x)$ and $R_d(x)$

$$\dot{x} = [J_d(x) - R_d(x)]\frac{\partial H_d}{\partial x} \quad \text{instead of} \quad \dot{x} = [J(x) - R(x)]\frac{\partial H_d}{\partial x}$$

- **Interconnection and Damping assignment**
- Much more **freedom** (overcome the dissipation obstacle)
- Must solve an additional (and harder) **PDE**

Casimir Method

Casimir functions represent an "**invariant of motion**" of the system, i.e., a **conserved quantity** along the open-loop trajectories, evolving in free evolution ($u(t) = 0$) and independently of the Hamiltonian of the system $H(x)$

$$\boxed{C(x(t)) = const \rightarrow \dot{C}(x(t)) = 0}$$

Formally, given a PHS
$$\begin{cases} \dot{x} &= [J(x) - R(x)]\dfrac{\partial H}{\partial x} + g(x)u \\ y &= g^T(x)\dfrac{\partial H}{\partial x} \end{cases}$$

Casimir functions must satisfy: $\boxed{\dfrac{\partial C^T}{\partial x}(J(x) - R(x)) = 0}$ so that $\dot{C}(x) = \frac{\partial C^T}{\partial x}\dot{x}$ whatever $H(x)$.

Therefore, Casimir functions are determined only by the internal structure of the system (interconnection structure J(x) or dissipation structure R(x)).
Then to find it we can check if $J(x) - R(x)$ singular (det =0) → If YES, let $a(x)$ so that $a^T(x)[J(x) - R(x)] = 0$ → one can hope to solve the PDE $\frac{\partial C(x)}{\partial x} = a(x)$ and determine $C(x(t))$.

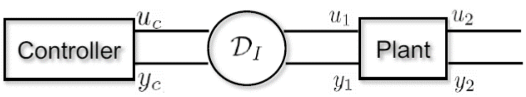
And... defining $H_d(x) = H(x) + C(x)$ it follows $\dot{H}_d(x) = \dot{H}(x)$ → we can use $C(x)$ to shape $H(x)$ while retaining the same "convergence properties"!!

We can reformulate the approach as an interconnection between a **PHS plant** and a **PHS controller** (to be determined):

$$\begin{cases} \dot{x} &= [J(x) - R(x)]\dfrac{\partial H}{\partial x} + g(x)u \\ y &= g^T(x)\dfrac{\partial H}{\partial x} \end{cases} \qquad \begin{cases} \dot{x}_c &= (J_c(x_c) - R_c(x_c))\dfrac{\partial H_c}{\partial x_c} + g_c(x_c)u_c \\ y_c &= g_c^T(x_c)\dfrac{\partial H_c}{\partial x_c} \end{cases}$$

and the interconnection structure D_1 taken for simplicity as

$$u = -y_c + v_1,\ u_c = y + v_2$$



Now look for **Casimir functions** $C(x, x_c)$ satisfying:

$$\begin{bmatrix} \dfrac{\partial C^T}{\partial x} & \dfrac{\partial C^T}{\partial x_c} \end{bmatrix} \left(\begin{bmatrix} J(x) & -g(x)g_x^T(x_c) \\ g_c(x_c)g^T(x) & J_c(x_c) \end{bmatrix} - \begin{bmatrix} R(x) & 0 \\ 0 & R_c(x_c) \end{bmatrix} \right) = 0$$

We seek stabilization at some $(x, x_c) = (x^*, \forall\,)$. Note that $\dot{C} = 0$ constraints x_c to be function of the plant state $x_c = \Gamma(x)$ → the total Hamiltonian is function of the plant state only: $H_d(x) = H(x) + H_x(\Gamma(x)) + c$

We look for a Casimir function $C(x, x_c) = x_c - F(x)$ and we can show that if the following 4 conditions are satisfied then $x_c = F(x) + c$. In this way the plant dynamics becomes $\dot{x} = [J(x) - R(x)]\frac{\partial H_d}{\partial x}$ where $H_d(x) = H(x) + H_c(F(x) + c)$.

1) $\dfrac{\partial F^T}{\partial x}J(x)\dfrac{\partial F}{\partial x} = J_c(x_c)$ $\quad -\dfrac{\partial F^T}{\partial x}R(x)\dfrac{\partial F}{\partial x} = R_c(x_c)$

with the latter further implying

$R_c(x_c) = 0$, $\quad R(x)\dfrac{\partial F}{\partial x} = 0$, $\quad \dfrac{\partial F^T}{\partial x}J(x) = g_c(x_c)g^T(x)$

2) **3)** **4)**

For the controller, since $R_c(x_c) = 0$, it is $\dot{H}_c = y_c^T u_c$ → All the energy flowing through (u_c, y_c) is stored (released) in $H_c(x_c)$.
The shaped (closed-loop) Hamiltonian evolves as $\dot{H}_d = \dot{H} + \dot{H}_c = \dot{H} - y^T u$ because of the interconnection $u = -y_c, u_c = y$ Therefore, as expected: **Energy Balancing:**

$$H_d(x(t)) = H(x(t)) - \int_{t_0}^{t} y^T(\tau)u(\tau)\mathrm{d}\tau + c$$

The shaped energy is the difference between: the energy stored in the plant $H(x(t))$ and the energy supplied by the controller $\int_{t_0}^{t} y^T(\tau)u(\tau)\mathrm{d}\tau = -H_c(x)$

NB: constraints **2)** and **3)** are <u>**dissipation obstacle:**</u>

- The controller cannot dissipate energy because of **2)**
- The energy shaping cannot be performed on those coordinates affected by plant dissipation because of **3)**

Physical reason: a passive controller can stabilize equilibria (x^*, x_c^*) where <u>**no energy dissipation takes place**</u>. (in mechanical systems it is a problem for the kinetic energy part only)

Is it possible to overcome the dissipation obstacle? → yes but we need to solve the PDE!

Consider the usual PHS controller $\begin{cases} \dot{x}_c &= u_c \\ y_c &= \dfrac{\partial H_c}{\partial x_c} \end{cases}$ but characterized by an <u>unbounded</u> Hamiltonian $H_c(x_c) = -x_c$ →This controller can provide an infinite amount of energy (it is not passive).

Consider a state-modulated interconnection $\begin{bmatrix} u \\ u_c \end{bmatrix} = \begin{bmatrix} 0 & -\beta(x) \\ \beta^T(x) & 0 \end{bmatrix} \begin{bmatrix} y \\ y_c \end{bmatrix}$

The closed-loop becomes:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} J(x) - R(x) & -g(x)\beta(x) \\ \beta^T(x)g^T(x) & 0 \end{bmatrix} \begin{bmatrix} \dfrac{\partial H}{\partial x} \\ \dfrac{\partial H_c}{\partial x_c} \end{bmatrix}$$

Note that, in closed-loop, the plant dynamics does not depend on x_c: $\dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + g(x)\beta(x)$

Assume that one can find a function $H_a(x)$ which solves the PDE $g(x)\beta(x) = [J(x) - R(x)]\frac{\partial H_a}{\partial x}$
→Then, under the feedback $u = \beta(x)$, the plant closed-loop becomes

$$\dot{x} = [J(x) - R(x)]\frac{\partial H_d}{\partial x}$$

with a new (shaped) energy function $H_d(x) = H(x) + H_a(x)$.

However, the structural matrixes $J(x), R(x)$ stay the same! We can actually go even further: if we find a controller that also shapes $J_d(x)$and $R_d(x)$ → more degree of freedom!

↓

3. Interconnection and Damping assignment

It can be shown that if a system $\dot{x} = f(x)$ is asymptotically stable at $x = x^*$ then

- $\boxed{f(x) = [J_d(x) - R_d(x)]\frac{\partial H_d}{\partial x}}$ with positive def. $H_d(x)$ having a minimum in $x = x^*$.
- If there exists $u = \beta(x)$ such that $\boxed{\dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + g(x)\beta(x)}$ is asymptotically stable at $x = x^*$ then the closed-loop is equivalent to:

$$\boxed{\dot{x} = [J_d(x) - R_d(x)]\frac{\partial H_d}{\partial x}}$$

With desired $H_d(x)$, $J_d(x)$and $R_d(x)$.

To find the controller we need to solve the PDE which may be <u>very hard</u>, but many degrees of freedom!

$$g^{\perp}(x)[J(x) - R(x)]\frac{\partial H}{\partial x} = g^{\perp}(x)[J_d(x) - R_d(x)]\frac{\partial H_d}{\partial x}$$

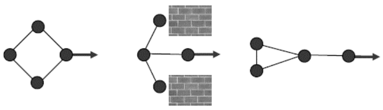
Where $g^{\perp}g = 0$. If this is possible, then the controller is

$$u = \beta(x) = [g^T(x)g(x)]^{-1}g^T(x)\left([J_d(x) - R_d(x)]\frac{\partial H_d}{\partial x} - [J(x) - R(x)]\frac{\partial H}{\partial x}\right)$$

Formation Control of Multiple Robots

Formation Control with Time-varying graph topology:

- Robots (e.g., UAVs) are loosely coupled together: can gain/lose neighbors, but must show some form of cohesive behavior
- **decentralized** design (local and 1-hop communication/sensing)
- Overall motion controlled by **leaders**
- **flexible formation**: splits/joins due to sensing/communication constraints, need to temporarily split for better maneuvering, execution of extra tasks in parallel to the collective motion
- Autonomy in **avoiding obstacles** and **collisions**



Anyway, we have to face a **Time-varying topology** → we need to ensure stability and guarantee passivity of the overall group.

- Consider the **agent** as a **free-floating mass** in R^3 with Energy $K_i = \frac{1}{2} p_i^T M_i^{-1} p_i$

$$\boxed{\begin{cases} \dot{p}_i = F_i^a + F_i^e - B_i M_i^{-1} p_i \\ v_i = \frac{\partial K_i}{\partial p_i} = M_i^{-1} p_i \end{cases}} \qquad i = 1, \ldots, N$$

- p_i = **agent momentum** and v_i the agent **velocity**.
- x_i, with $\dot{x}_i = v_i$ the agent position
- M_i = the agent **Inertia matrix**
- $B_i \geq 0$ is a **velocity damping term**
- Force (input) F_i^a the **interaction (coupling) with the other agents**
- Force (input) F_i^e the **interaction with the "external world"** (e.g., obstacles)

In PHS terms, an agent represents an **atomic element storing kinetic energy K_i** with two power ports (F_i^a, v_i) and (F_i^e, v_i).

• **Heterogeneity** in the group can be enforced by choosing different M_i and B_i

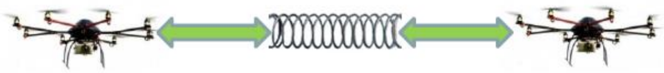
Neighbours: Let consider a max **communication range** D and let $d_{ij} = ||x_i - x_j||$ be the **interdistance** among two agents. → Two agents cannot be neighbors if $d_{ij} > D$ (if they are too far apart).
Moreover, to take into account a *time-varying neighbouring condition* consider $\sigma_{ij} \in \{0,1\}$ in which

$$\sigma_{ij}(t) = 0 \quad if \;\; d_{ij} > D$$

$\sigma_{ij}(t) = \sigma_{ji}(t)$→ we have a time-varying Undirected Graph $G = (V, \varepsilon(t))$.

When neighbors ($\sigma_{ij}(t) = 0$), the agents should keep a **cohesive formation:** assume we want to maintain a desired **interdistance** $0 < d_0 < D$ by means of local information (decentralization) and by exploiting the coupling force F_i^a in the agent dynamics.

These interactions are modelled as a (**nonlinear**) **elastic element**:



$$\begin{cases} \dot{p}_i = F_i^a + F_i^e - B_i M_i^{-1} p_i \\ v_i = \frac{\partial \mathcal{K}_i}{\partial p_i} = M_i^{-1} p_i \end{cases} \qquad \begin{cases} \dot{x}_{ij} = v_{ij} \\ F_{ij}^a = \frac{\partial V(x_{ij})}{\partial x_{ij}} \end{cases} \qquad \begin{cases} \dot{p}_j = F_j^a + F_j^e - B_j M_j^{-1} p_j \\ v_j = \frac{\partial \mathcal{K}_j}{\partial p_j} = M_j^{-1} p_j \end{cases}$$

PHS form for storing element

$V(x_{ij})$ the energy function (Hamiltonian) bounded and with a minimum in d_0 (flat when $d_{ij} > D$).

Example: 3 agents

$$\begin{bmatrix} F_1^a \\ F_2^a \\ F_3^a \\ v_{12} \\ v_{13} \\ v_{23} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ F_{12}^a \\ F_{13}^a \\ F_{23}^a \end{bmatrix} \Longrightarrow \boxed{\begin{bmatrix} F_1^a \\ F_2^a \\ F_3^a \\ v_{12} \\ v_{13} \\ v_{23} \end{bmatrix} = \begin{bmatrix} 0 & E_{\mathcal{G}} \\ -E_{\mathcal{G}}^T & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ F_{12}^a \\ F_{13}^a \\ F_{23}^a \end{bmatrix}}$$

Missing edge "23"

Note that the **coupling Force** F_i^a for agent i can be computed in a decentralized way (Need to know only x_i and x_j).

$$F_i^a = \sum_{j \in \mathcal{N}_i} e_{ij} F_{ij}^a := \sum_{j \in \mathcal{N}_i} e_{ij} \frac{\partial \bar{V}}{\partial d_{ij}} \frac{\partial d_{ij}}{\partial x_{ij}}$$

Matrix $E(t)$ is the incidence matrix of the graph that takes into account labelling and orientation (through v_{ij}) and for missing edges (all zeros).

Let us now **generalize for N agents:**

For N agents → N(N-1)/2 elastic elements states(edges): $x = (x_{12}^T, \ldots, x_{1N}^T, x_{23}^T, \ldots, x_{2N}^T, \ldots, x_{N-1N}^T)^T$

- Let $p = (p_1^T, \ldots, p_N^T)^T \in \mathbb{R}^{3N}$ collect all the agent states (momenta)
- Let $B = diag(B_i) \in \mathbb{R}^{3N \times 3N}$ collect all the damping terms
- Let $H = \sum_{i=1}^{N} \mathcal{K}_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} V(x_{ij})$ be the Total Energy (Hamiltonian)

The overall group of **interconnected (power-preserving) agents** becomes the **PHS:**

$$\begin{cases} \begin{pmatrix} \dot{p} \\ \dot{x} \end{pmatrix} = \left[\begin{pmatrix} 0 & E(t) \\ -E^T(t) & 0 \end{pmatrix} - \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \right] \begin{pmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial x} \end{pmatrix} + G F^e \\ v = G^T \begin{pmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial x} \end{pmatrix} \end{cases}$$

Has an **external port** (v, F^e) interacting with the **external world** (obstacles, external commands).

Where $F^e = (F_1^{eT} \ldots. F_N^{eT})^T$ and $v = (v_1^T \ldots v_N^T)$

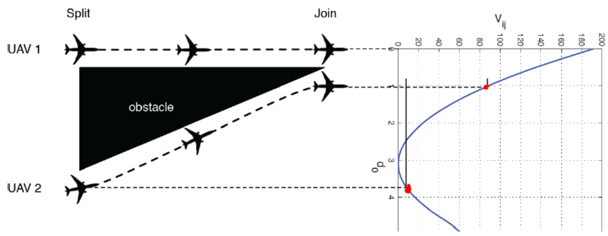
$$\begin{cases} \dot{p}_i = F_i^a + F_i^e - B_i M_i^{-1} p_i \\ v_i = \frac{\partial \mathcal{K}_i}{\partial p_i} = M_i^{-1} p_i \end{cases}$$

Let us then study the passivity of the group w.r.t. the port (v, F^e):

- If fixed topology $E(t) = E = const$ → the group of agents is **passive** w.r.t. its **external port** since bounded

$$\dot{H} = -\frac{\partial^T H}{\partial p} B \frac{\partial H}{\partial p} + v^T F^e \leq v^T F^e$$

- In general $E(t)$:
  - If **split** $\sigma_{ij} = 1 \rightarrow \sigma_{ij} = 0$ → The edge (i,j) is lost and the Incidence matrix is updated accordingly $E \rightarrow E'$ which still remain skew-symmetric matrix → overall <u>passivity is preserved.</u>
  - If **join** → $E \rightarrow E'$ BUT we need to also update the state $x_{ij} \leftarrow x_i - x_j$ which <u>costs extra energy</u>! (thus, <u>can violate passivity</u>)



$$V_{join} > V_{split}$$

If extra energy, $\Delta V = V_{join} - V_{split} > 0$, is needed, this must be taken from sources already present in the group to maintain passivity (no internal production of extra energy).
→ To cover it we can use **Energy Tanks** and **Energy Transfer control** (in decentralized way):

- we store back the agent inherent dissipation $\boldsymbol{D_i}$ and use this to passively implement the join

1) augment each agent state with the Tank dynamics
2) provide the elastic elements with an additional input w_{ij}^x for exchanging energy with the Tanks (w_{ij}^x, w_{ij}^t, w_{ji}^t , will allow for drawing ΔV from the Tanks of agents i and j)

$$\begin{cases} \dot p_i &= F_i^a + F_i^e - B_i M_i^{-1} p_i \\ \dot t_i &= \dfrac{1}{x_{t_i}} D_i + w_{ij}^t \\ y &= \begin{bmatrix} v_i \\ x_{t_i} \end{bmatrix} \end{cases} \Longleftrightarrow \begin{cases} \dot x_{ij} &= v_{ij} + w_{ij}^x \\ F_{ij}^a &= \dfrac{\partial V(x_{ij})}{x_{ij}} \end{cases} \Longleftrightarrow \begin{cases} \dot p_j &= F_j^a + F_j^e - B_j M_j^{-1} p_j \\ \dot t_i &= \dfrac{1}{x_{t_j}} D_j + w_{ji}^t \\ y &= \begin{bmatrix} v_j \\ x_{t_j} \end{bmatrix} \end{cases}$$

- **Energy Transfer control** among two **PHS** implemented by the coupling

$$\begin{bmatrix} w_{ij}^x \\ w_{ij}^t \\ w_{ji}^t \end{bmatrix} = \begin{bmatrix} 0 & -\gamma_{ij} F_{ij}^a t_i & -\gamma_{ij} F_{ij}^a t_j \\ \gamma_{ij} F_{ij}^{a\,T} t_i & 0 & 0 \\ \gamma_{ij} F_{ij}^{a\,T} t_j & 0 & 0 \end{bmatrix} \begin{bmatrix} F_{ij}^a \\ t_i \\ t_j \end{bmatrix}$$

← γ_{ij} dictates the **rate** and **direction** of Energy transfer. If $\gamma_{ij} < 0$ refills the spring energy and draws from the two Tanks.

(passive) join decision strategy:

- compute $\Delta V = V(x_i - x_j) - V(x_{ij})$ (the one we need – the one we have)
  a. if $\Delta V \leq 0$, implement the join and store back ΔV into the tanks T_i and T_j)
  b. if $\Delta V > 0$, extract ΔV from T_i and T_j

- If $T_i + T_j < \Delta V$? → Do not join (and wait for better conditions) OR ask the rest of the group for "help", by implementing a consensus on all the **Tank Energies** (this redistributes the energies within the group, but it doesn't change the total amount of energy=passivity)

$$\begin{cases} \dot p_i &= F_i^a + F_i^e - B_i M_i^{-1} p_i \\ \dot x_{t_i} &= (1-\beta_i)\left(\dfrac{1}{x_{t_i}} D_i + \sum_{j=1}^N w_{ij}^t\right) + \beta_i c_i \\ y &= \begin{bmatrix} v_i \\ x_{t_i} \end{bmatrix} \end{cases}$$

$\beta_i \in (0,1)$ enable/disable consensus mode if after the consensus still not enough energy)→ The agents do not join, They can switch to a high damping mode for more quickly refilling the Tanks.

Proposition: the overall **group dynamics** (with Tanks, Energy Transfer, Consensus, and PassiveJoin Procedure) is still **passive** $\dot{H} \leq v^T F^e$

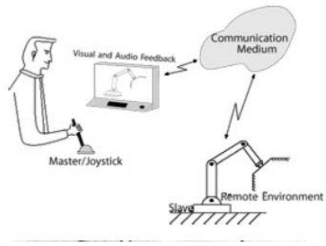
$$\begin{cases} \begin{bmatrix} \dot{p} \\ \dot{x} \\ \dot{x}_t \end{bmatrix} = \left(\begin{bmatrix} 0 & E(t) & 0 \\ -E^T(t) & 0 & \Gamma^T \\ 0 & -\Gamma & 0 \end{bmatrix} - \begin{bmatrix} B & 0 & 0 \\ 0 & 0 & 0 \\ -(I-\beta)PB & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \frac{\partial \mathcal{H}}{\partial p} \\ \frac{\partial \mathcal{H}}{\partial x} \\ \frac{\partial \mathcal{H}}{\partial x_t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \beta c \end{bmatrix} + GF^e \\ v = G^T \begin{bmatrix} \frac{\partial \mathcal{H}}{\partial p} \\ \frac{\partial \mathcal{H}}{\partial x} \\ \frac{\partial \mathcal{H}}{\partial x_t} \end{bmatrix} \end{cases}$$

where the new Hamiltonian is $\mathcal{H} = \sum_{i=1}^{N} \mathcal{K}_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} V(x_{ij}) + \sum_{i=1}^{N} T_i$ and $\beta = diag(\beta_i)$, $P = diag(\frac{1}{t_i} p_i^T M_i^{-T})$, and matrix $\Gamma \in \mathbb{R}^{N \times \frac{3N(N-1)}{2}}$ representing the interconnection between Tanks and springs

Bilateral Teleoperation of Multiple UAVs

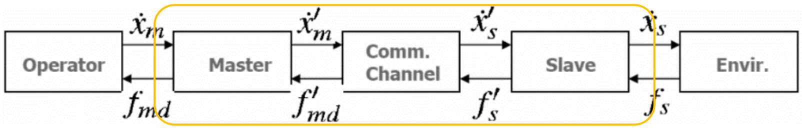
We can apply this application to the case of bilateral teleoperation system in which we attach agent group **(slave-side)** to the **master-side** thanks to passivity.

- Neighbouring condition: $d_{ij} \leq D$ + two agents can interact if and only if their ***line-of-sight*** is not occluded.



←"remote" coupling: $human\ operator \leftrightarrow master$ and $slave \leftrightarrow environment$

Then, can be modelled as an exchange of force/position information = exchange of energy → as a **N-port PHS**



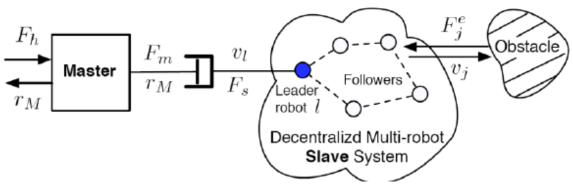
The **goals** are to endure a **stable** Teleoperation behavior (to operator and environment) while ensuring **transparency**. → we need to ensure that $master \leftrightarrow communication\ channel \leftrightarrow slave$ systems are **passive**.

Remark - Passivity of the Master:
We want to synchronize the position of the master with the velocity of the slave.
The master can be modelled as a *Euler Lagrange mechanical system* → which is passive w.r.t. (F_M, v_M) but not with respect to position-force.
Then we can make the system passive w.r.t. (F_M, r) with $r = v_M + \lambda x_M$,$\lambda > 0$ and then we rescale into r_M ← the contribution of the velocity can be made negligible, so the output is <u>almost</u> a position.

Model everything as a **PHS**: **Storing**, **Dissipation**, **Exchange** of Energy.

Consider 1 leader, and split its external force as $F_l^e = F_s + F_l^{env}$, then interconnect master ↔leader in this (passive) way:

$$\begin{cases} F_s = b_T(r_M - v_l) \\ F_m = -b_T(r_M - v_l) \end{cases} , \quad b_T > 0$$



• v_l is the **leader velocity** and r_M is (almost) the **master position**, corresponding to a velocity command

• Force F_m will inform about the mismatch $v_l - r_M$, it will weigh the total inertia (number of agents), the absolute speed of the group and Obstacles.
NB: **Obstacles** are considered as passive systems producing repulsive forces (spring-like elements).

Remark: as an agent move, **dissipate energy** (damping terms). This is stored back into the **Tank** but then still used to implement **joins**. → new needed energy can be only supplied by the **master**.
However, also the master passive, it cannot create energy over time so at some point, its internal energy <u>storage will also be depleted</u> → The energy to keep everything in motion comes from the Human operator , which acts on the master performing mechanical work (= energy)

Velocity Synchronization at SS

Assume a constant velocity for the leader $r_M = const$. We look if the agents at SS synchronize with the velocity command $v_i \to r_M,\ \forall i$.

The existence of a **steady state** is guaranteed if:

1. No environmental forces/ obstacles $F_i^{env} = 0$
   → Assume $F_{leader}^e = F_s = b_T(r_M - v_{leader})$ and for all others $F_i^e = F_i^{env} = 0$

2. Tanks are full and there is no joins and no energy exchanges with elastic elements (T_i and $\Gamma = 0$)

3. $\mathcal{G}$ is connected (→, $ker E^T = \mathbf{1}_{N_3}$)
   → Then $\frac{\partial H}{\partial p} = \mathbf{1}_{N_3} v_{ss}$ = All the agents have the **same velocity.**

At steady state, we have that $(\dot{p}, \dot{x}, \dot{t}) = (0,0,0)$. In fact, velocity stays constant if we assume constant mass, relative positions stay constant and energy tanks stay constant too.

Assume $F_{leader}^e = F_s = b_T(r_M - v_{leader})$ and for all others $F_i^e = F_i^{env} = 0$ then

The velocities converge to: $\boxed{v_i \to v_{ss} = (\mathbf{1}_{N_3}^T B' \mathbf{1}_{N_3})^{-1} b_T r_M}$

And $||v_{ss}|| < ||r_M||$ → agents always **travel "slower"** than the commanded r_M because of dumping (Perfect synchronization only if $b_i = 0$).

→ In fact, we have that damping B is good for **stabilization** and **Tank refill**, but it **slows** down agent.

$$\dot{p}_i = F_i^a + F_i^e - B_i M_i^{-1} p_i \qquad \qquad \left\{ \ \dot{p}_i = F_i^a + F_i^e + F_i^s + F_i^d \right.$$

We add a dumping force $\boldsymbol{F_i^d}$ into the agent dynamics in which we consider a "**switching" damping** that is present only if the tanks are not full; in that case, we need the damping to refill them.

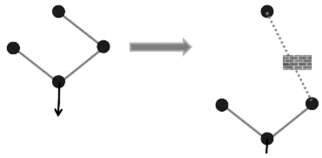
$$B_i(t_i) = \begin{cases} 0 & \text{if} \quad T(t_i) = \bar{T}_i \\ \bar{B}_i & \text{if} \quad T(t_i) < \bar{T}_i \end{cases}$$

The additional (**synchronization**) force $\boldsymbol{F_i^s}$ is designed as (consensus among velocities) $F_i^s = -b \sum_{j \in \mathcal{N}_i} (v_i - v_j)$

The overall system Is still passive.

→Therefore, the system converges towards a **steady-state condition** $(\dot{p}, \dot{x}, \dot{t}) = (0,0,0)$ with

- **perfect synchronization** with leader velocity commands $v_i \to r_M = const$
- all relative positions stay constant $\dot{x} = 0$



Connectivity Maintenance

We can understand if a graph can stay connected while maintaining split and join.

Connected graph → $\lambda_2 > 0$ is a measure of the degree of connectivity in a graph, the larger its value, the "more connected" the graph.

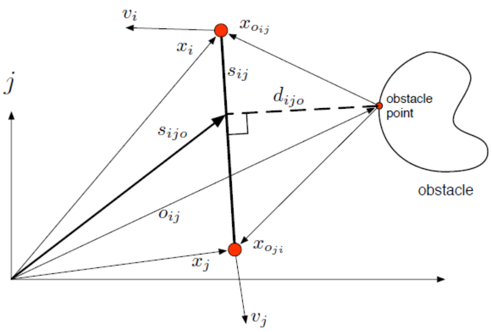
We would lie to have $\lambda_2 = \lambda_2(x)$ and then just implement some **gradient-like controller** $u = \frac{\partial \lambda_2}{\partial x}$.
This situation is possible if we assume the weights of the Adjacency matrix are **smooth functions** of the state $A_{ij} = A_{ij}(x) \geq 0$ (rather than $A_{ij} = \{0,1\}$)
→ Then, the Laplacian itself becomes a smooth function $L(x) = \Delta(x) - A(x)$.
By choosing the control in this way, we also keep the decentralized structure (because we obtain from $\lambda_2 = v_2^T L v_2 \rightarrow \; d\lambda_2 = v_2^T dL\, v_2 \rightarrow \frac{\partial \lambda_2}{\partial x_i} = \sum_{(i,j)\in\mathcal{E}} \frac{\partial A_{ij}}{\partial x_i}(v_{2_i} - v_{2_j})^2$)

How to design of the **weights $A_{ij}(x)$**:
They should take into account physical limitations for interacting (occlusion and max range), requirements that agents should preferably met and the one that they should necessarily met → This can be achieved by maximizing λ_2 ("physical" connectivity + any additional group requirement)*

→ A possible choice: $A_{ij} = \alpha_{ij}\, \beta_{ij}\, \gamma_{ij}$

define the set $S_i = \{j | \gamma_{ij} > 0\}$ as the sensing neighbours and $N_i = \{j | A_{ij} > 0\}$ as the usual neighbours. As for sensing/communication constraints we consider again maximum range and line-of-sight occlusion.

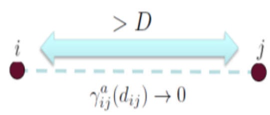
- We consider the following definitions:
  - s_{ij} is the segment joining agents i and j
  - o_{ij} is the closest obstacle point to s_{ij}
  - s_{ijo} is the closest point on s_{ij} to o_{ij}
  - d_{ijo} is the distance from s_{ij} to o_{ij}

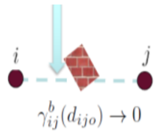


- The term $\gamma_{ij} \geq 0$ accounts for **physical limitation** so it represents the **sensing/communication model**.

Take $\gamma_{ij} = \gamma_{ij}^a(d_{ij})\, \gamma_{ij}^b(d_{ijo})$, d_{ijo} is the distance between the edge (i,j) and the closest obstacle.

- $\gamma_{ij}^a(d_{ij}) \rightarrow 0$ when exceeding the maximum range ($d_{ij} \rightarrow D$)
- $\gamma_{ijo}^b(d_{ijo}) \rightarrow 0$ when occlusion occurs.





- The term $\beta_{ij} \geq 0$ accounts for **soft requirements** (as keep a desired distance)

  $\beta_{ij}(d_{ij}) \rightarrow 0$ as $\left\| d_{ij} - d_0 \right\| \rightarrow \infty$ And it has a unique maximum at $d_{ij} = d_0$.

- The term $\alpha_{ij} \geq 0$ accounts for **hard/mandatory requirements** (as collision avoidance)

  - $a_{ij}(d_{ij}) \rightarrow 0$ as $d_{ij} \rightarrow 0$ (so, the two agents disconnect if they are too close).
  - $a_{ik} \rightarrow 0$, $\forall k \in \mathcal{N}_i$ (whenever two agents become too close, all the neighbouring agents will disconnect).

This will lead to the disconnected graph ($\lambda 2 \rightarrow 0$), that is the situation that we want to avoid.

The term α_{ij} is made of a product of several terms $\alpha_{ij}^*(d_{ij}) \geq 0$

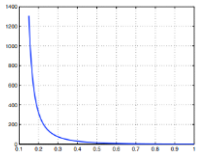
$$\alpha_{ij} = \left(\prod_{k \in \mathcal{S}_i} \alpha_{ik}^*\right) \cdot \left(\prod_{k \in \mathcal{S}_j / \{i\}} \alpha_{jk}^*\right) = \alpha_i \alpha_j$$

Each $\alpha_{ij}^* \rightarrow 0$ as $d_{ij} \rightarrow 0$ (i,j) gets too close. → we want to enforce that if $d_{ij} \rightarrow 0$ for a particular pair (i,j), **<u>the whole graph approaches disconnection</u>** → in this way the entire i-th row of A will vanish.

The term α_j is introduced just for symmetry so to ensure that $a_{ij} = a_{ji}$. Moreover $b_{ij} = b_{ji}$ and $\gamma_{ij} = \gamma_{ji}$ → symmetric Adjacency matrix $A = A^T$

*A possible control design: Since we have a PHS system, we define the **Hamiltonian** as a **connectivity Potential function** $V^\lambda(\lambda_2) \geq 0$ which

- Vanishes for $\lambda_2 \rightarrow \lambda_2^{max}$
- Grows unbounded for $\lambda_2 \rightarrow \lambda_2^{\min}$



This will be the storage function for our passivity arguments.
Its gradient (**connectivity force**) is:

$$\boxed{F_i^\lambda = \frac{\partial V^\lambda(\lambda_2)}{\partial \lambda_2} \frac{\partial \lambda_2(x_R, x_O)}{\partial x_i}}$$

This is function of the state (x_R, x_o) (if we consider edges and agent-obstacles position) which is equals to

$$F_i^\lambda = \frac{\partial V^\lambda(\lambda_2)}{\partial \lambda_2} \sum_{j \in \mathcal{N}_i} \left(\frac{\partial A_{ij}}{\partial x_{ij}} + \frac{\partial A_{ij}}{\partial x_{ijo}}\right) (v_{2_i} - v_{2_j})^2$$

This function still can be implemented with a local and 1-hop information controller, but we need $\lambda_2, v_{2_i}, v_{2j}$ for implementing it → thus we need the full Laplacian (centralized)
Alternatively, we can use a **decentralized estimation** of these value and get a fully decentralized implementation of $\widehat{F}_i^\lambda$.

If we look at the group dynamics → behave as a PHS and is passive with respect to power ports (F^e, v) and (F^0, v_0) associated to the **obstacle motion**. With $H(p, x_R, x_O) = \sum_{i=1}^{N} \mathcal{K}_i(p_i) + V^\lambda(x_R, x_O) \geq 0$

And it should be **passive** w.r.t. its power ports! $\dot{H} \leq v^T F^e + v_0 F^0$!

However, there can be two source of non-passive behaviour:

1) First: possible **positive jumps** in $V^\lambda(\lambda_2)$ because of **join decisions** (as before)
2) Second: **estimation errors** in evaluating $\widehat{F}_i^\lambda$ (in place of the real F_i^λ)

Anyway, in this framework, the first issue cannot happen. In fact, A varies smoothly, so there are no discontinuities. To solve estimation error, we can use the **tank energy**: store dissipated energy, and use this energy for implementing $\widehat{F}_i^\lambda$. This works well because <u>in our framework</u>, the tank will never deplete over time.

New agent dynamics augmented with the Tank element:

$$\begin{cases} \dot{p}_i = F_i^e - w_i x_{t_i} - B_i M_i^{-1} p_i \\ \dot{x}_{t_i} = s_i \frac{1}{x_{t_i}} D_i + w_i^T v_i \\ y_i = \left(v_i^T x_{t_i}\right)^T \end{cases} \qquad s_i = \begin{cases} 0, & \text{if } T_i \geq T_{\max} \\ 1, & \text{if } T_i < T_{\max} \end{cases} \qquad w_i = -\varsigma_i \frac{\hat{F}_i^\lambda}{x_{t_i}}, \quad \varsigma_i \in \{0, 1\}$$

The parameter s_i prevents excessive storage in the Tank. → Force $\widehat{F}_i^\lambda$ is then implemented by setting ω_i.
While ς enables/disables the implementation of $\widehat{F}_i^\lambda$ when **close to Tank depletion.**

→ ensures **passivity of the group** but it does not automatically guarantee **Connectivity Maintenance.**

$$\mathcal{H}(p, x_R, x_O, x_t) = \sum_{i=1}^{N} (\mathcal{K}_i(p_i) + T_i(x_{t_i})) + V^\lambda(x_R, x_O) \qquad \dot{\mathcal{H}} \leq v^T F^e + v_o^T F^o$$

In general $\widehat{F}_i^\lambda$ could <u>not be implemented</u> because the Tank is depleted (bad estimation), but doesn't happen in our case.

Fact 1: <u>Tank will never deplete</u> (provided a correct initialization of $T(x_{t_i}(t_0))$)

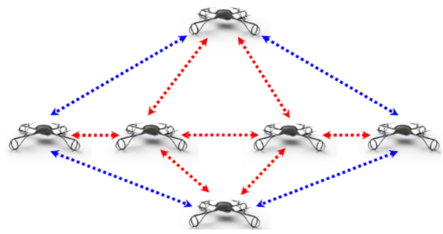
Fact 2: the estimation strategy used for $\hat{F}_i^\lambda$ is guaranteed to have a <u>bounded error</u> (with tunable accuracy)

The **estimation algorithm** is a continuous-time version of the **Power Iteration Procedure** for computing eigenvectors and eigenvalues of a matrix. The idea is to estimate (in a decentralized way) the eigenvector v_2 . This, in turn, allows to also estimate λ_2.
It consists of three steps

- 1) Deflation $\dot{\hat{v}}_2 = -\frac{k_1}{N}\mathbf{1}\mathbf{1}^T\hat{v}_2$ for removing the components spanned by $v_1 = \mathbf{1}$
- 2) Direction update $\dot{\hat{v}}_2 = -k_2 L \hat{v}_2$ for moving towards v_2
- 3) Renormalization $\dot{\hat{v}}_2 = -k_3\left(\frac{\hat{v}_2^T\hat{v}_2}{N} - 1\right)\hat{v}_2$ from staying away from the null-vector
- Altogether: $\dot{\hat{v}}_2 = -\frac{k_1}{N}\mathbf{1}\mathbf{1}^T\hat{v}_2 - k_2 L\hat{v}_2 - k_3\left(\frac{\hat{v}_2^T\hat{v}_2}{N} - 1\right)\hat{v}_2$
- And it can be shown that $\hat{\lambda}_2 = \frac{k_3}{k_2}\left(1 - \|\hat{v}_2\|^2\right)$

Every decentralized apart from the average and the average norm which can be estimated <u>in a decentralized way</u> by making use of the PI-ACE estimator (**proportional/integral-Average Consensus Estimator**)

Rigidity Maintenance



The desired formation cannot be maintained using only the available distance measurements → A minimum number of distance measurements are required to uniquely determine the desired formation!

→ Graph rigidity!

By embedding in R^3 one obtains $\ker\big(R_G(p)\big) = 6$ for a rigid graph. The constraint-preserving motions are the 3 translations and 3 rotations around an arbitrary p^* (the motions of a rigid body in 3D space)

- The **symmetric rigidity matrix** is defined as $\mathcal{R} = R_{\mathcal{G}}^T(p)R_{\mathcal{G}}(p) \in \mathbb{R}^{3N\times 3N}$
- The eigenvalues satisfy $\lambda_1 = \lambda_2 = \cdots = \lambda_6 = 0, \quad \lambda_7 > 0$

One can define a "**Rigidity Eigenvalue**"λ_7 that can be used as smooth measure of rigidity.
→by maintaining a formation rigidity we can run a decentralized estimator able to obtain relative positions out of measured relative distances. Relative positions are then needed by the rigidity controller