Fault-tolerant formation control using energy tanks

SAPIENZA
UNIVERSITÀ DI ROMA

Facci Matteo (1597454)
Galli Mattia (1753274)
Giunta Gaetano (1692966)
Sensolini Arrà Giuseppe (1661198)

Control of Multi-Robot Systems- A.Y. 2020-2021

Summary

- Introduction
- Brief overview on the theoretical aspects of formation control
  - Hamiltonian elements used to achieve such control
- Analysis of the passive reconfiguration strategy
  - Energy tanks
  - Split and join events
- Results and comments of our simulations
- Final considerations

Introduction

- Formation control: tries to achieve a geometrical shape for the network of agents
- Port-Hamiltonian systems theory: energy-based modeling framework
  - Systems as the interconnection of **energy storing** and **energy dissipating** components which exchange energy through **power-ports**
- Main goal of this work: implement a passivity-based reconfiguration strategy in case of faults of one agent
- Passivity will be preserved by exploiting the concept of energy **tanks**

Formation control of fully actuated systems

- Aim: achieve a prescribed geometrical shape for a network of agents using only local feedback controllers
- Controllers are all based on assigning virtual couplings between the agents
  - virtual spring + (optional) virtual damper
- **Fully actuated** agents: agents for which the number of inputs equals the degrees of freedom

The system

- Group of N agents, where each agent is modeled as a single point mass m_i moving in $\mathbb{R}^n$
- The position of agents i is denoted by $q_i \in \mathbb{R}^n$ and the corresponding momentum is defined as $p_i = m_i \dot{q}_i \in \mathbb{R}^n$
- Each agent has a control port (u_i, y_i), and a resistive port (u_i^r, y_i^r)

System dynamics

- The dynamics for a single agent are given by:

$$\begin{cases} \begin{pmatrix} \dot{q}_i \\ \dot{p}_i \end{pmatrix} = \begin{pmatrix} 0 & I_n \\ -I_n & -D_i{}^a(q_i, p_i) \end{pmatrix} \begin{pmatrix} \dfrac{\partial H_i{}^a}{\partial q_i}(q_i, p_i) \\ \dfrac{\partial H_i{}^a}{\partial p_i}(q_i, p_i) \end{pmatrix} + \begin{pmatrix} 0 \\ I_n \end{pmatrix} u_i + \begin{pmatrix} 0 \\ I_n \end{pmatrix} u_i{}^r \\ y_i = y_i{}^r = \dfrac{\partial H_i{}^a}{\partial p_i}(p_i) \end{cases}$$

- $D_i^a(q_i, p_i)$ is the dissipation matrix, $H_i^a(q_i, p_i)$ is the Hamiltonian
- The Hamiltonian equals the kinetic energy associated to the movement of the mass: $H_i^a(p_i) = \frac{1}{2} p_i^T M_i^{-1} p_i$

System dynamics (2)

- Then the dynamics for N agents are given by:

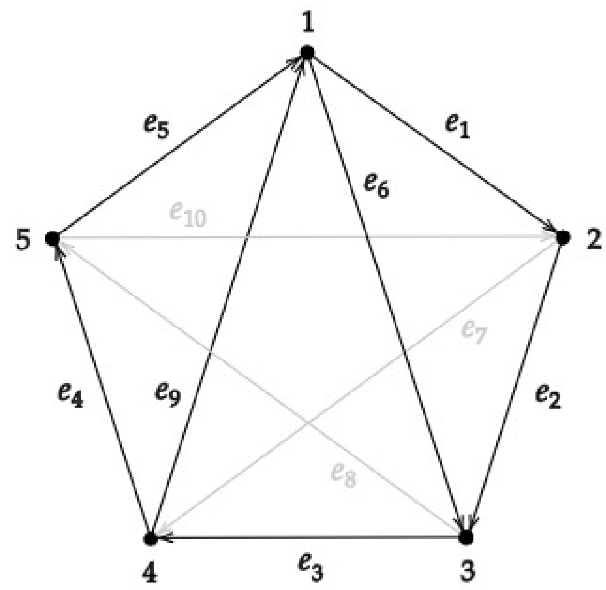
$$\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} = \begin{pmatrix} 0 & I_{Nn} \\ -I_{Nn} & -D^a(q,p) \end{pmatrix} \begin{pmatrix} \frac{\partial H^a}{\partial q}(q,p) \\ \frac{\partial H^a}{\partial p}(q,p) \end{pmatrix} + \begin{pmatrix} 0 \\ I_{Nn} \end{pmatrix} u + \begin{pmatrix} 0 \\ I_{Nn} \end{pmatrix} u^r$$

$$y = y^r = \frac{\partial H^a}{\partial p}(p)$$

with Hamiltonian $H^a(p) = \sum_{i=1}^{N} H_i^a(q_i, p_i) = \frac{1}{2} p^T M^{-1} p$

Tree graph

- Consider a network of N agents with the previous form (resistive port omitted for simplicity)
- Formation control is achieved by assigning virtual couplings in between the agents
- The interconnection topology amongst agents via virtual couplings is modeled by a tree graph
  - the N nodes of the graph correspond to the agents, while the E edges correspond to the virtual couplings



Formation control objective

- For each agent:
  - $q_i \in \mathbb{R}^n$ denotes its position
  - $p_i \in \mathbb{R}^n$ denotes the corresponding momentum.
  - $z_j \in \mathbb{R}^n$ denotes the relative displacement for two agents interconnected by virtual coupling j
  - $z_j^* \in \mathbb{R}^n$ denotes the desired relative displacement
- the formation control objective can be formally stated as

$$\begin{cases} p \to 0 \\ z \to z^* \end{cases} \quad \text{as} \quad t \to \infty$$

Virtual coupling

- Each virtual coupling consists of a virtual spring and damper in parallel
- The dynamics of such a spring-damper system are given by:

$$\dot{z}_j = v_j$$

$$F_j = \frac{\partial H_j{}^c}{\partial z_j} + D_j{}^c v_j$$

z_j is the spring elongation
v_j is the input velocity
F_j is the corresponding output force

- For each virtual coupling j, the Hamiltonian H_j^c equals the potential energy in the spring j

$$H_j^c(z_j) = \frac{1}{2}\left(z_j - z_j^*\right)^T K_j^c\left(z_j - z_j^*\right)$$

Virtual coupling (2)

- The dynamics of E virtual couplings are summarized as:

$$\dot{z} = v$$

$$F = \frac{\partial H^c}{\partial z} + D^c v$$

with Hamiltonian $H^c(z) = \sum_{i=1}^{E} H_j^c(z_j) = \frac{1}{2}(z - z^*)^T K^c (z - z^*)$

Closed-loop dynamics

- Let B denote the **incidence matrix** associated to the tree graph, then the coupling of agents on the nodes and virtual couplings at the edges is given by:

$$u = -(B \otimes I_n)F$$

$$v = -(B^T \otimes I_n)y$$

- the closed-loop dynamics are given by

$$\begin{pmatrix} \dot{p} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} -(D^a(p) + BD^cB^T) & -B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial z} \end{pmatrix}$$

with Hamiltonian

$$H(z,p) = \sum_{i=1}^{N} H_i^a(p_i) + \sum_{i=1}^{E} H_j^c(z_j) = \frac{1}{2} p^T M^{-1} p + \frac{1}{2}(z - z^*)^T K^c (z - z^*)$$

Control input

- The solutions of the closed-loop system converge to $p = 0, z = z^*$, achieving the formation control objectives
- The control input for the agents is given by:

$$u = -(B \otimes I_n)K^c(z - z^*) - (B \otimes I_n)D^c(B^T \otimes I_n)M^{-1}p$$

- First term: virtual spring force → ensures that the formation control objectives are achieved
- Second term: virtual damping force→ can be used to shape the transient response

Passivity of the system

- It is possible to prove that the system is **passive** with respect to the input/output pair (u, y)

$$\begin{aligned}\dot{H} &= \begin{pmatrix}\frac{\partial H}{\partial p} & \frac{\partial H}{\partial z}\end{pmatrix}^T \begin{pmatrix}\dot{p} \\ \dot{z}\end{pmatrix} \\ &= \frac{\partial H^T}{\partial p}\left(-D^a \frac{\partial H}{\partial p} - B\frac{\partial H}{\partial z}\right) + \frac{\partial H^T}{\partial z} B^T \frac{\partial H}{\partial p} \\ &= -\frac{\partial H^T}{\partial p} D^a \frac{\partial H}{\partial p} \leq 0\end{aligned}$$

Neighboring agents, split and join

- Let $d_{ij} = \|z_i - z_j\|$, be the interdistance among two agents, they cannot be neighbors if $d_{ij} > D$.

$$\begin{cases} \sigma_{ij} = 0 & if\, d_{ij} > D \\ \sigma_{ij} = 1 & if\, d_{ij} < D \end{cases}$$

- The formation is dynamic → the parameter σ_{ij} is time-varying
- **Split** event: it could happen that some agents disconnect due to lack of communication or simply because their interdistance grows
- **Join** event: it is possible that some agents get closer and their interdistance reduces below the threshold D, generating a new connection

Energy tanks and Energy Transfer control

- Necessary if $\Delta E = V(z_i - z_j) - V(z_{ij}^s) = V_{join} - V_{split} > 0$

- Energy tanks
  - Keep track of the energy dissipated by each agent
  - Defined by $t_i \in \mathbb{R}$
  - Energy function $T_i = \frac{t_i^2}{2}$

- Store back the energy dissipated $D_i = p_i^T M_i^{-T} D_i^a M_i^{-1} p_i$

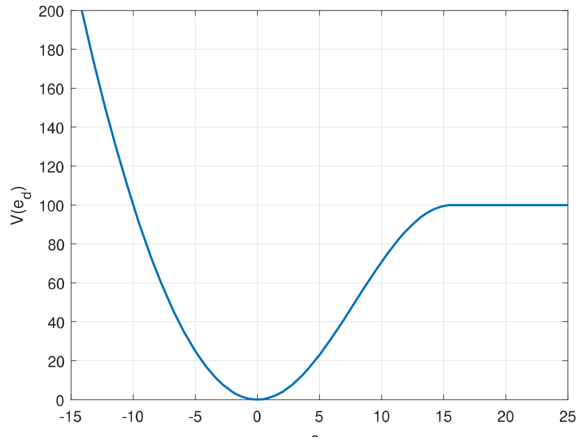
Augmented dynamics and interagent storing action

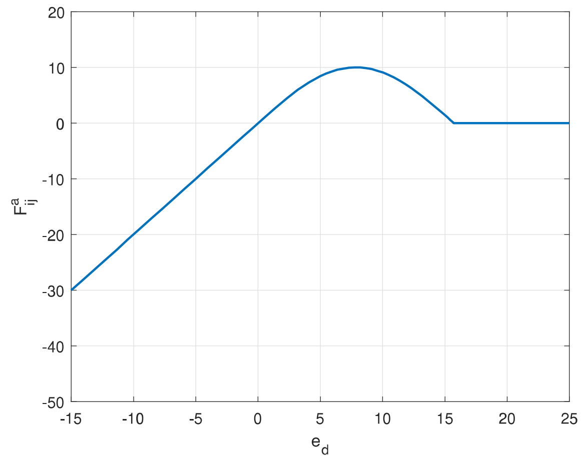
Agent state

$$
\begin{cases}
\dot{p}_i = F_i^a + F_i^e - D_i^a M^{-1} p_i \\
\dot{t}_i = (1-\beta_i)\left(\alpha_i \dfrac{1}{t_i} D_i + \sum_{j=1, j\neq i}^{N} w_{ij}^T F_{ij}^a\right) + \beta_i c_i \\
y_i = \begin{pmatrix} M_i^{-1} p_i \\ t_i \end{pmatrix}
\end{cases}
$$

Elastic element

$$
\begin{cases}
\dot{z}_{ij} = v_{ij} - w_{ij} t_i + w_{ji} t_j \\
F_{ij}^a = \dfrac{\partial V(e_{ij})}{\partial e_{ij}}
\end{cases}
$$





*always referring to the general *i*-th agent

Energy flow

- α_i enables/disables the energy storing of D_i : $\dot{T}_i = \alpha_i \frac{1}{t_i} D_i + \sum_{j=1}^{\mathcal{N}} w_{ij}^T F_{ij}^a$

$$\alpha_i = \begin{cases} 0, & \text{if } T_i \geq \overline{T}_i \\ 1, & \text{otherwise} \end{cases}$$

- $\beta_i \in \{0,1\}$ (optional) switch from *storage mode (0)* to *consensus mode (1)*
  - If $\beta_i = 1$, in order to obtain $\dot{T}_i = -\sum_{j \in \mathcal{N}_i}^{\mathcal{N}} (T_i - T_j)$: $c_i = -\frac{1}{t_i} \sum_{j \in \mathcal{N}_i}^{\mathcal{N}} (T_i(t_i) - T_j(t_j))$

- w_{ij} input to allow for drawing ΔE from the tanks of the respective agents

$$w_{ij} = \gamma_{ij}(1 - \beta_i) t_i F_{ij}^a$$

- γ_{ij} modulates the rate and direction of the energy flow

$$\begin{cases} \gamma_{ij} > 0 & \text{energy is extracted from elastic term and stored in tank} \\ \gamma_{ij} < 0 & \text{energy is extracted from tank and stored in the elastic term} \\ \gamma_{ij} = 0 & \text{agents } i \text{ and } j \text{ are not interconnected} \end{cases}$$

Passive join procedure

1. Agents i and j split → the one with the lower ID stores z_{ij} in a local variable z_{ij}^s (state of the virtual spring at the split time)
   - If the two agents never split before, z_{ij}^s is initialized such that
   $$V(z_{ij}^s) = \bar{V}(D) = \bar{V}_{ij}(\infty)$$
2. At the join moment, agent i computes the quantity
   $$\Delta E = V(z_i - z_j) - V(z_{ij}^s)$$
   - if $\Delta E \leq 0$, implement the join (and store ΔE back into the tanks)
   - if $\Delta E > 0$, extract ΔE from the tanks and then implement the join.
     If it is not sufficient:
     - Avoid join procedure
     - Exploit the tanks of the rest of the fleet (if they contain enough energy) → Consensus mode

Consensus mode

- If the energy stored in the tanks of the two agents is not sufficient:
  - agent i asks the fleet to activate the β_i in order to switch to consensus mode
  - the consensus is run until the redistribution of the energy among the tanks is completed (total tank energy will remain unchanged)
  - After this redistribution, agents i and j check again if there is enough energy in the tanks for joining
- If the energy in the tanks is not yet sufficient:
  - it is necessary to act directly on the robots to refill the tanks, augmenting the damping

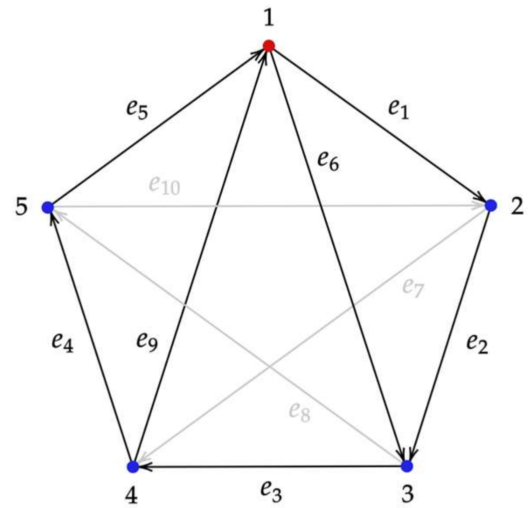
Procedure PassiveJoin

Data: $x_i, x_j, x_{ij}^s, t_i, t_j$

Compute $\Delta E = V(x_i - x_j) - V(x_{ij}^s)$;
if $\Delta E \leq 0$ **then**
  Store $(-\Delta E)/2$ in the tank through input w_{ij};
else
  if $T_i(t_i) + T_j(t_j) < \Delta E + 2\varepsilon$ **then**
    Run a *consensus* on the tank variables;
    if $2T_i(t_i) < \Delta E + 2\varepsilon$ **then**
      Dampen until $T(t_i) + T(t_j) \geq \Delta E + 2\varepsilon$;
  Extract $\dfrac{T(t_i)}{T(t_i) + T(t_j)} \Delta E$ from the tank through input w_{ij};
Join;

Case of study: 5-Robots formation

- Let us consider a 5-robots formation in a leader-follower configuration
- The complete graph has 10 edges
- Robots are organized in a pentagon formation, with inner-edges e_6 and e_9 connected (in the initial configuration)
- The incidence matrix **B is updated at run-time**
- The complete incidence matrix is described by:

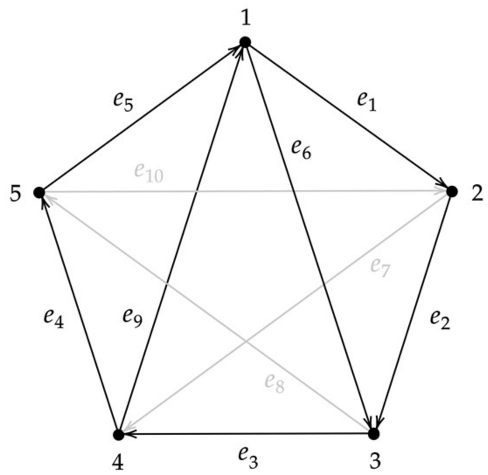


$$B^* = [e_1^* \quad \dots \quad e_{10}^*] = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

Case of study: 5-Robots formation

Fault tolerant control strategy → Reconfiguration in presence of a faulty agent

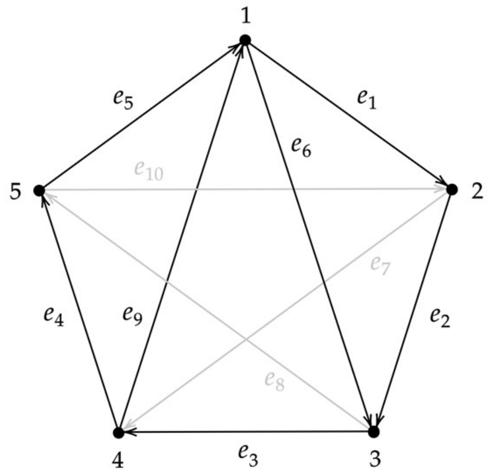
Initial graph



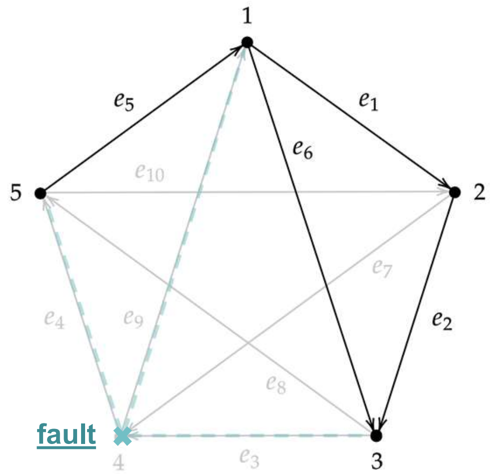
Case of study: 5-Robots formation

Fault tolerant control strategy → Reconfiguration in presence of a faulty agent

Initial graph

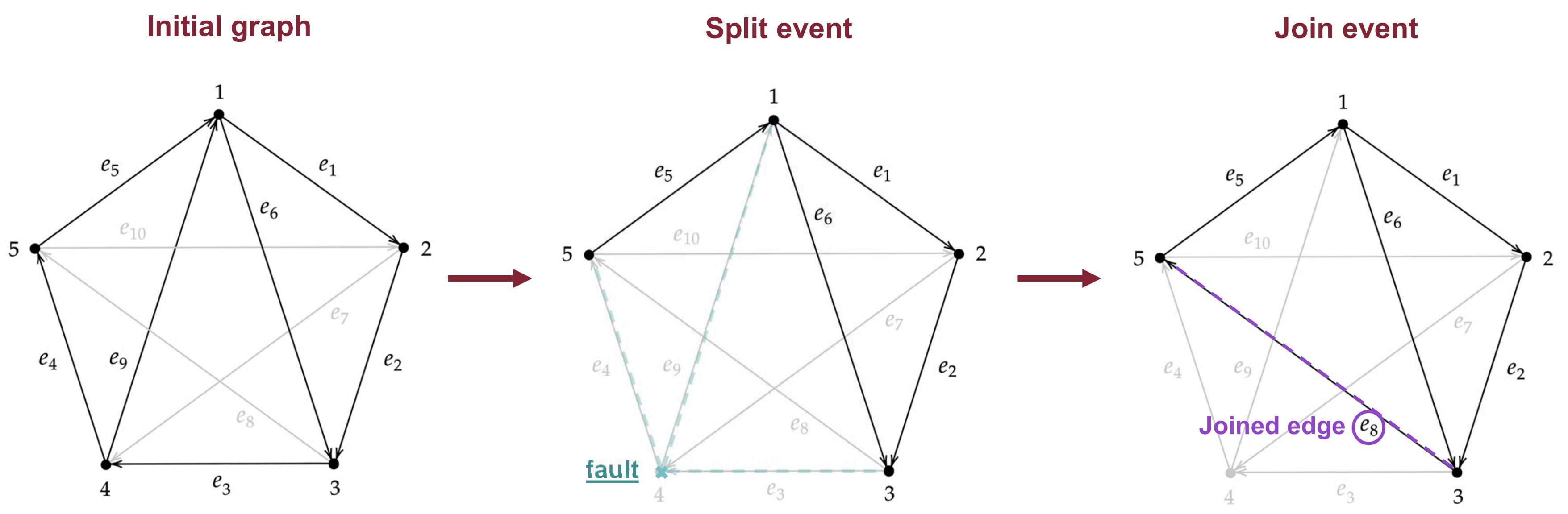


Split event

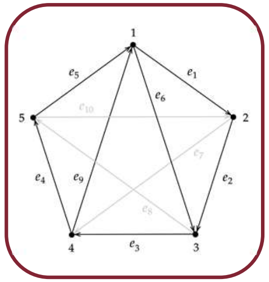


Case of study: 5-Robots formation

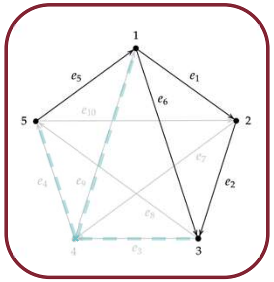
Fault tolerant control strategy → Reconfiguration in presence of a faulty agent



Case of study: 5-Robots formation

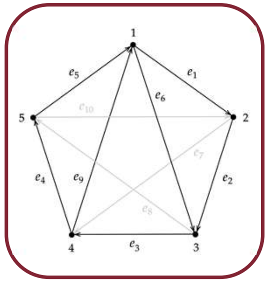


Split procedure

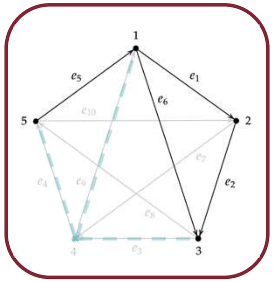



```
SplitProcedure(faulted_robot,B):
  faulted_edges = getEdges(faulted_robot,B);
  For edge in faulted_edges:
    B(:,edge) = zeros(N,1);
  End
  Return B;
```

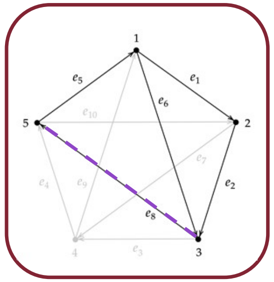
Case of study: 5-Robots formation



Split procedure



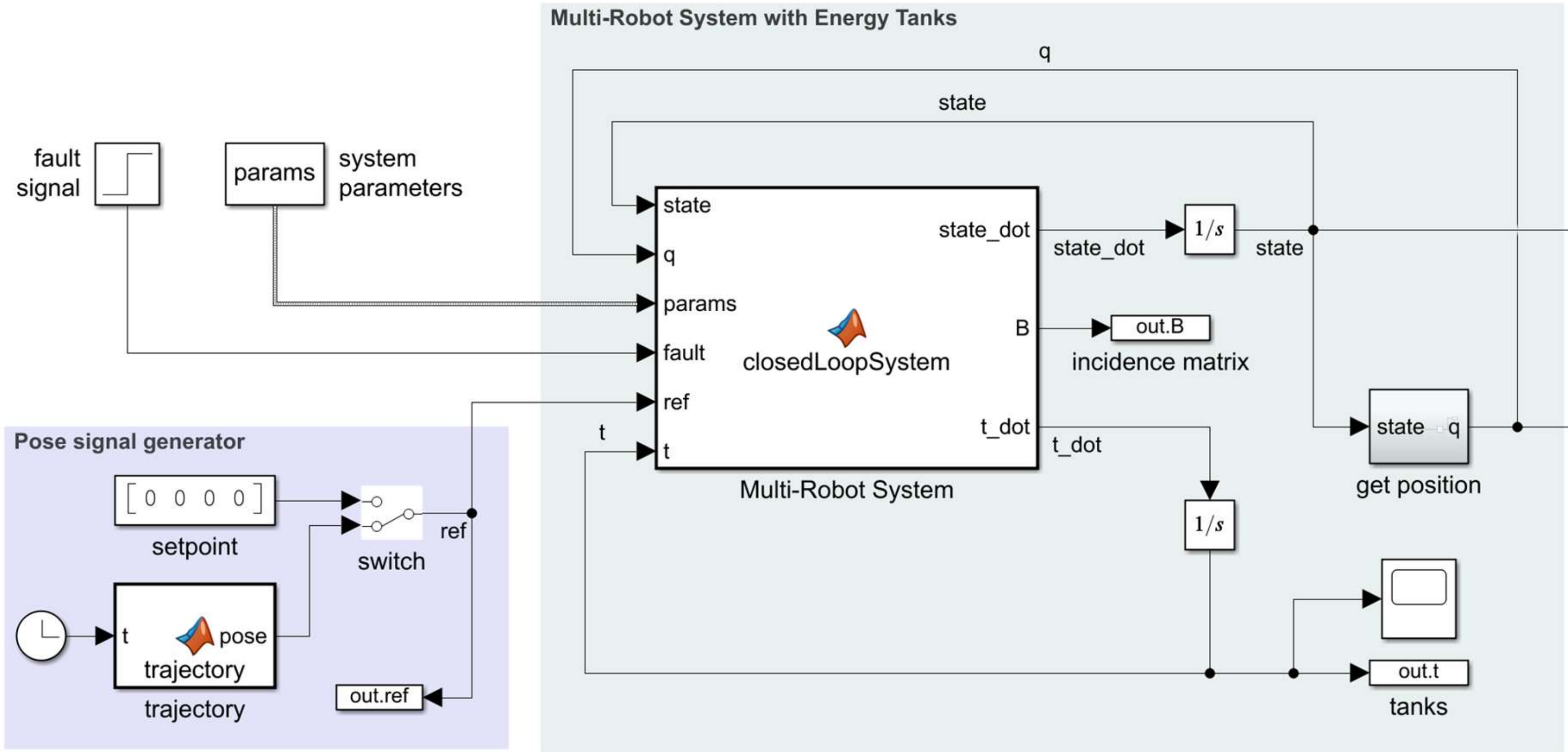
Join procedure




```
SplitProcedure(faulted_robot,B):
  faulted_edges = getEdges(faulted_robot,B);
  For edge in faulted_edges:
    B(:,edge) = zeros(N,1);
  End
  Return B;
```

```
JoinProcedure(faulted_robot,B,B*):
  i = faulted_robot - 1;
  j = faulted_robot + 1;
  joined_edge = getEdge(i,j,B*);
  B(:,joined_edge) = B*(:,joined_edge)
  Return B;
```

Matlab/Simulink implementation



Simulation settings

- Robots are assumed to be equal, their dynamic parameters are:

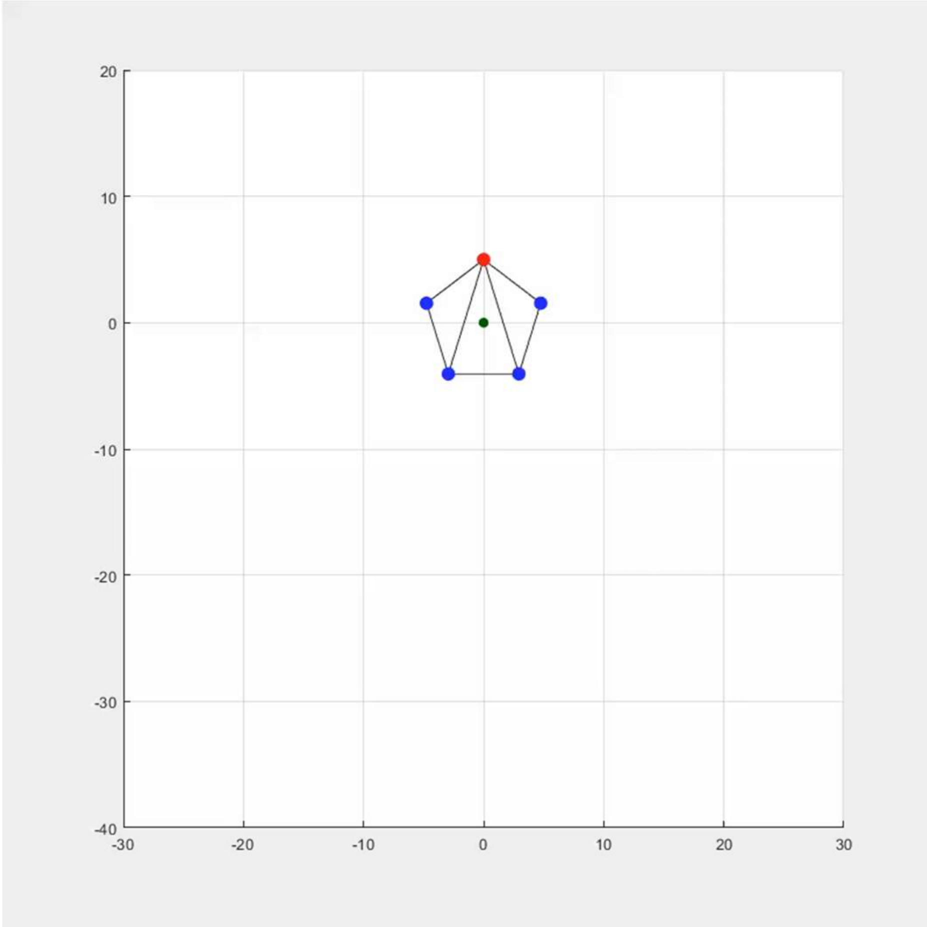
parameter	value	unit
m (robot mass)	1	Kg
μ (friction coeff.)	1	/
d_c (damping coeff.)	3	/
k_c (elastic coeff.)	5	/

- Initial conditions:
  - `p(0) = zeros(15,1)`
  - `q(0)` and `z(0)` constitute a pentagon inscribed in a circumference of radius `r=5`
  - `t(0) = 5*ones(5,1)`

- Desired edges: $|z\_des_i|=20$ for $i=1,\dots,5$, $|z\_des_i|=32.36$ for $i=6,\dots,10$

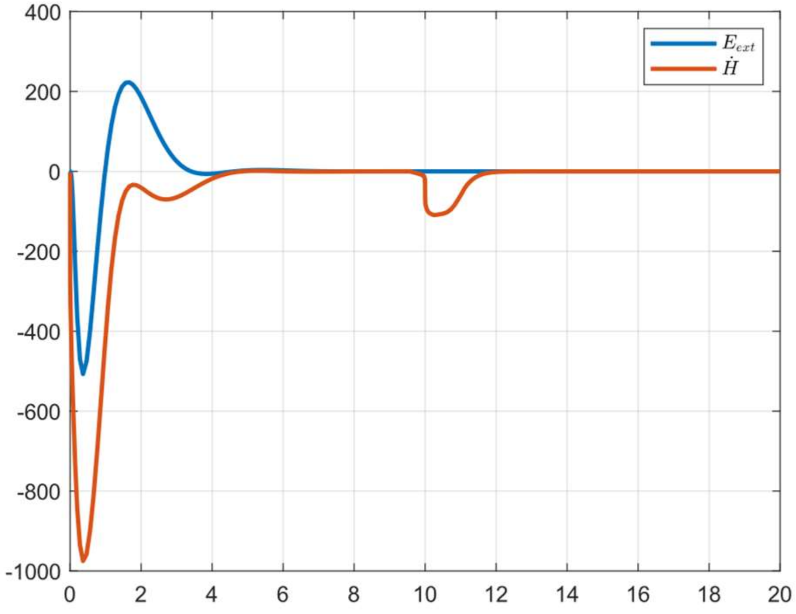
Simulation 1: setpoint regulation

- Setpoint regulation tasks:
  - build desired formation
  - proportional controller (leader)
  - PD controller (ensuring safe land fault agent)
  - setpoint $[x\ y\ z\ \theta] = \left[0\ 0\ 0\ \frac{\pi}{2}\right]$
- Split at time:
  - 15 [sec]
- Settling time
  - 5 [sec]
- Expected results:
  - Energy drain from tanks
  - Passivity

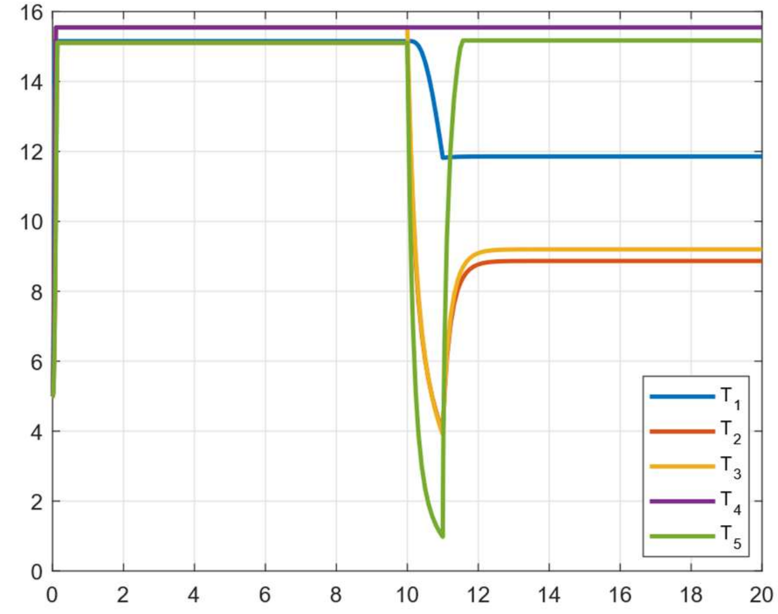


Simulation 1: setpoint regulation

- passivity

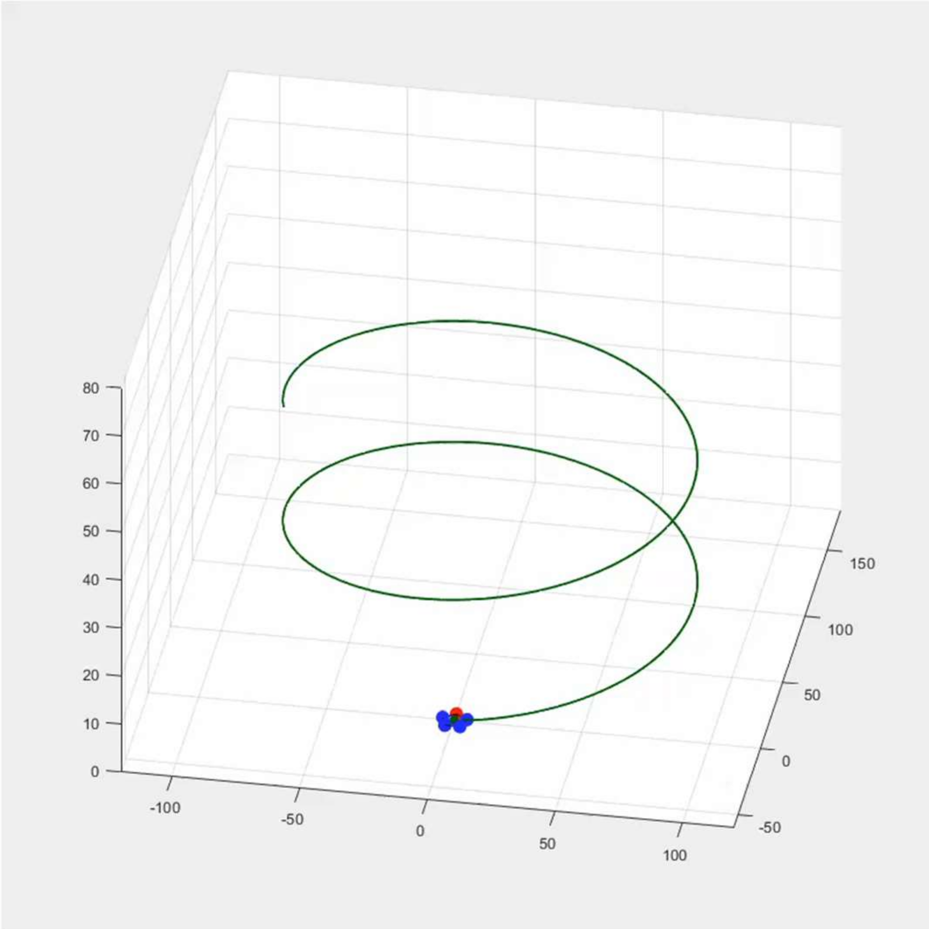


- tanks



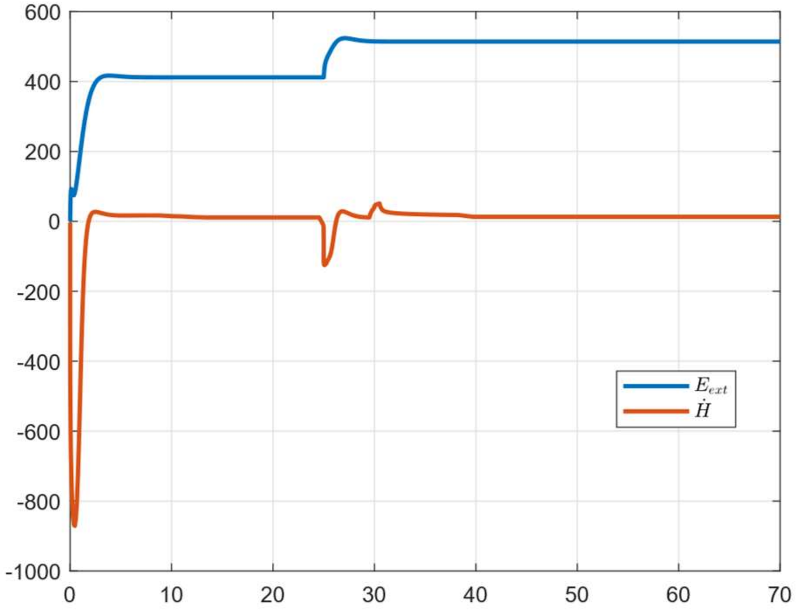
Simulation 2: trajectory tracking

- Trajectory tracking tasks:
  - helicoidal trajectory (green line)
  - proportional controller (leader)
  - PD controller (ensuring safe land fault agent)
- Split at time:
  - 25 [sec]
- Settling time
  - 5 [sec]
- Expected results:
  - Energy drain from tanks
  - Passivity

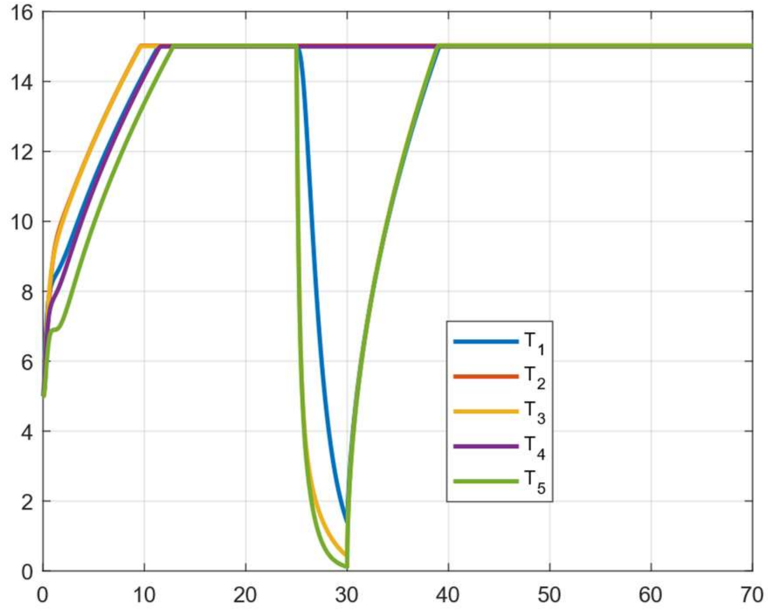


Simulation 2: trajectory tracking

- passivity



- tanks



Conclusion

The proposed fault tolerant control strategy over the 5-robots system works as expected:

- Reconfiguration through Split and Join procedures guarantee the expected results in the presented case (and also with different formations/faulty agent)
- Leader correctly track the reference signal
- Tanks dynamics behave as expected
- The overall multi-robot system remains **passive**

Thank you for your attention