

# Decentralized Estimation and Control for Cooperative Mobile Manipulation

A paradigmatic Example of Decentralization in Multi-Robot Control

**Antonio Franchi**

LAAS–CNRS, Toulouse, France, Europe

Sapienza University, Rome

April 26th, 2016



For **more info** on the methods shown in this presentation:

**Estimation part:**

A. Franchi, A. Petitti, and A. Rizzo, "Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements", in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, 2015, pp. 5517–5522

A. Franchi, A. Petitti, and A. Rizzo, "Distributed estimation for cooperative mobile manipulation", *Under Review*, 2016. [Online]. Available:  
<http://arxiv.org/abs/1602.01891>

**Control + Estimation part:**

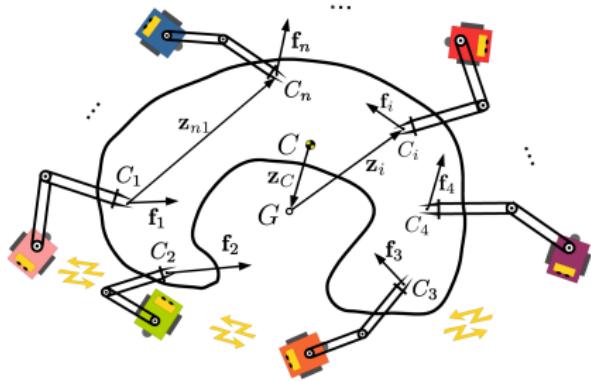
A. Petitti, A. Franchi, D. Di Paola, and A. Rizzo, "Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators", in *2016 IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, 2016, pp. 441–446

1. Motivations and Modeling
2. Decentralized Multi-Robot Control
3. Summary and Open Problems

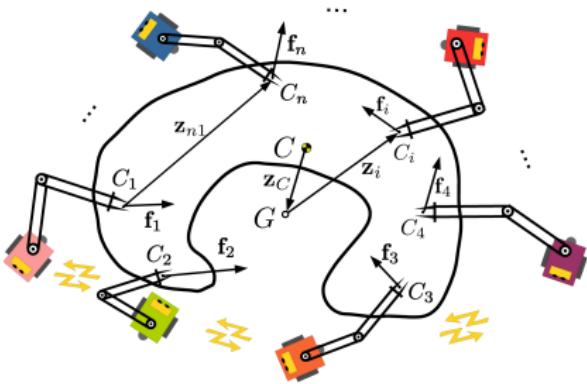
## Section 1

### Motivations and Modeling

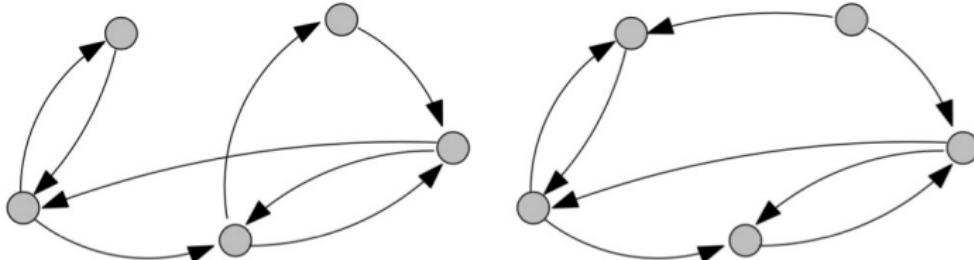
1. Decentralized control method for **cooperative manipulation** with a team of mobile robots



1. Decentralized control method for **cooperative manipulation** with a team of mobile robots



2. Useful paradigm for **decentralization** in multi-robot control



Credit: Graph Theoretic Methods in Multiagent Networks, Book by Magnus Egerstedt and Mehran Mesbah

# Motivation: Cooperative Mobile Manipulation

## Some areas

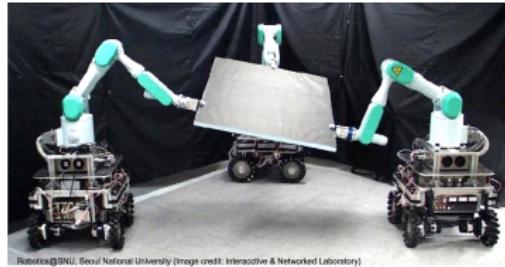
- **service and industrial robotics**
- **search&rescue**

## Some tasks

- cooperative **transportation**
- cooperative **assembly**

## Some problems/methods

- **motion planning**<sup>1</sup>
- **formation control**<sup>2</sup>
- **optimal control**<sup>3</sup>



<sup>1</sup> A. Yamashita, T. Arai, J. Ota, and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation", *IEEE Trans. on Robotics*, vol. 19, no. 2, pp. 223–237, 2003.

<sup>2</sup> A. Yufka, O. Parlaktuna, and M. Ozkan, "Formation-based cooperative transportation by a group of non-holonomic mobile robots", in *2010 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Istanbul, Turkey, 2010, pp. 3300–3307.

<sup>3</sup> D. Sieber, F. Deroo, and S. Hirche, "Formation-based approach for multi-robot cooperative manipulation based on optimal control design", in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 5227–5233.

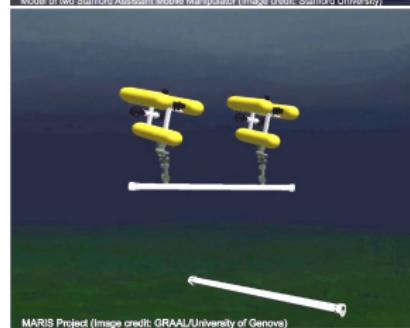
Why retrieving the manipulandum parameters . . .

. . . online:<sup>4</sup> 5

- less **control effort**
- use of **force control**
- **time-varying** manipulation tasks

. . . distributively:

- **flexibility**
- **robustness** to point failure
- less **computational** and **communication** overhead

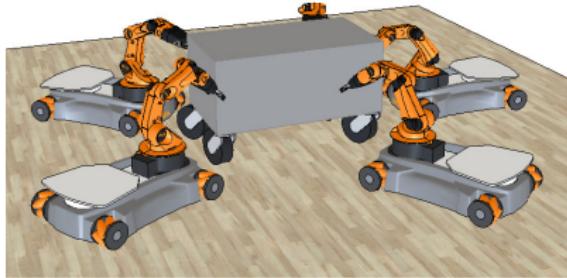


<sup>4</sup> Y. Yong, T. Arima, and S. Tsuji, "Inertia parameter estimation of planar object in pushing operation", in *2005 IEEE Int. Conf. on Information Acquisition*, Hong Kong and Macau, China, 2005, pp. 356–361.

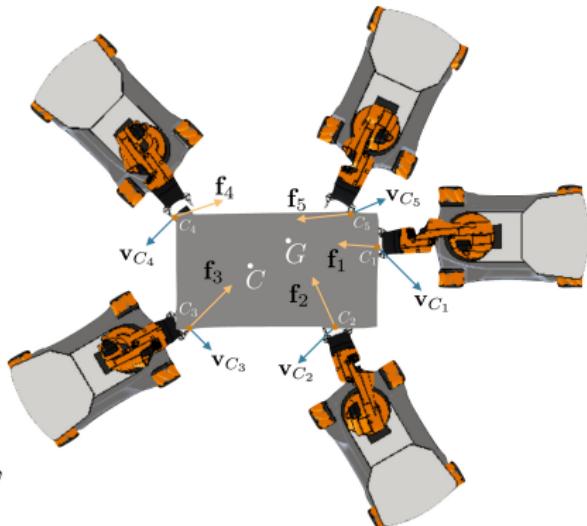
<sup>5</sup> D. Kubus, T. Kroger, and F. M. Wahl, "On-line estimation of inertial parameters using a recursive total least-squares approach", in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, France, 2008, pp. 3845–3852.

*n* mobile manipulators on a plane

- $\mathcal{G} = \{\mathcal{I}, \mathcal{E}\}$  undirected communication graph
- $\mathcal{G}$  is connected (not all-to-all)
- $\mathcal{N}_i = \{j \in \mathcal{I} : (i, j) \in \mathcal{E}\}$  neighbors of robot  $i$

Each robot  $i$ 

- exerts a force  $\mathbf{f}_i \in \mathbb{R}^2$ 
  - contact point  $C_i \in B$
  - null torque (extension is easy)
- measures the velocity  $\dot{\mathbf{p}}_{C_i}$  of  $C_i$
- anything else is unknown



Dynamics of the load  $B$ , subject to forces  $\mathbf{f}_1, \dots, \mathbf{f}_n$

Translational dynamics

$$\dot{\mathbf{v}}_C = \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i$$

$m \in \mathbb{R}_{>0}$  mass

$\mathbf{v}_C \in \mathbb{R}^2$  velocity of the CoM (center of mass)

# Model: Load

**Dynamics of the load**  $B$ , subject to forces  $\mathbf{f}_1, \dots, \mathbf{f}_n$

## Translational dynamics

$$\dot{\mathbf{v}}_C = \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i$$

$m \in \mathbb{R}_{>0}$  mass  
 $\mathbf{v}_C \in \mathbb{R}^2$  velocity of the CoM (center of mass)

## Rotational dynamics

$$\dot{\omega} = \frac{1}{J} \sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_C)^\perp {}^T \mathbf{f}_i$$

$J \in \mathbb{R}_{>0}$  moment of inertia

$\omega \in \mathbb{R}$  rotational rate

$\mathbf{p}_C \in \mathbb{R}^2$  position of the CoM

$\mathbf{p}_{C_i} \in \mathbb{R}^2$  position of the contact point  $C_i$ , for  $i = 1 \dots n$

$$(\cdot)^\perp \text{ rotation of } \pi/2: \quad \mathbf{v}^\perp = Q\mathbf{v} = \underbrace{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}}_{=Q} \begin{pmatrix} v^x \\ v^y \end{pmatrix} = \begin{pmatrix} -v^y \\ v^x \end{pmatrix}$$

# Model: Load

**Dynamics of the load**  $B$ , subject to forces  $\mathbf{f}_1, \dots, \mathbf{f}_n$

## Translational dynamics

$$\dot{\mathbf{v}}_C = \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i$$

$m \in \mathbb{R}_{>0}$  mass  
 $\mathbf{v}_C \in \mathbb{R}^2$  velocity of the CoM (center of mass)

## Rotational dynamics

$$\dot{\omega} = \frac{1}{J} \sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^{\perp T} \mathbf{f}_i = \frac{1}{J} \sum_{i=1}^n ((\mathbf{p}_{C_i} - \mathbf{p}_G)^{\perp T} \mathbf{f}_i) + \frac{1}{J} (\mathbf{p}_G - \mathbf{p}_C)^{\perp T} \sum_{i=1}^n \mathbf{f}_i$$

$J \in \mathbb{R}_{>0}$  moment of inertia

$\omega \in \mathbb{R}$  rotational rate

$\mathbf{p}_C \in \mathbb{R}^2$  position of the CoM

$\mathbf{p}_{C_i} \in \mathbb{R}^2$  position of the contact point  $C_i$ , for  $i = 1 \dots n$

$$(\cdot)^{\perp} \text{ rotation of } \pi/2: \quad \mathbf{v}^{\perp} = Q\mathbf{v} = \underbrace{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}}_{=Q} \begin{pmatrix} v^x \\ v^y \end{pmatrix} = \begin{pmatrix} -v^y \\ v^x \end{pmatrix}$$

$\mathbf{p}_G = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_{C_i}$  position of the centroid (geometric center) of the contact points

System model

$$\left\{ \begin{array}{lcl} \dot{\mathbf{v}}_C & = & \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i \\ \dot{\boldsymbol{\omega}} & = & \frac{1}{J} \sum_{i=1}^n (\mathbf{p}_{Ci} - \mathbf{p}_G)^{\perp T} \mathbf{f}_i + \frac{1}{J} (\mathbf{p}_G - \mathbf{p}_C)^{\perp T} \sum_{i=1}^n \mathbf{f}_i \end{array} \right.$$

Let the load velocity  $\mathbf{v}_C(t)$  and angular rate  $\boldsymbol{\omega}(t)$  follow a given

- **desired trajectory**  $\mathbf{v}_C^d(t)$  and  $\boldsymbol{\omega}^d(t)$ ,

using

- **only the available information** (local applied forces and local velocities)
- a **decentralized control law**

System model

$$\left\{ \begin{array}{lcl} \dot{\mathbf{v}}_C & = & \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i \\ \dot{\boldsymbol{\omega}} & = & \frac{1}{J} \sum_{i=1}^n (\mathbf{p}_{Ci} - \mathbf{p}_G)^{\perp T} \mathbf{f}_i + \frac{1}{J} (\mathbf{p}_G - \mathbf{p}_C)^{\perp T} \sum_{i=1}^n \mathbf{f}_i \end{array} \right.$$

Let the load velocity  $\mathbf{v}_C(t)$  and angular rate  $\boldsymbol{\omega}(t)$  follow a given

- desired trajectory  $\mathbf{v}_C^d(t)$  and  $\boldsymbol{\omega}^d(t)$ ,

using

- only the available information (local applied forces and local velocities)
- a decentralized control law

- without all-to-all communication
- without central processor
- without knowledge of  $\mathbf{v}_C(t)$ ,  $\boldsymbol{\omega}(t)$  (quantities to be controlled)
- without knowledge of  $(\mathbf{p}_G - \mathbf{p}_C)$  (internal state of the system)
- without knowledge of the parameters of the dynamical system

## Decentralized control law

Consider a network of robots performing a **control law**

The control law is **decentralized** if, for each robot  $i$ , the **size** of the

- **communication** bandwidth
- **computation** time (per step)
- **memory** used (inputs, outputs, local variables)

**depends only** on  $|\mathcal{N}_i|$  (number of comm. neighbors) and not on  $n$  (number of robots)

- a control law that is not decentralized is not **scalable**

## Decentralized control law

Consider a network of robots performing a **control law**

The control law is **decentralized** if, for each robot  $i$ , the **size** of the

- **communication** bandwidth
- **computation** time (per step)
- **memory** used (inputs, outputs, local variables)

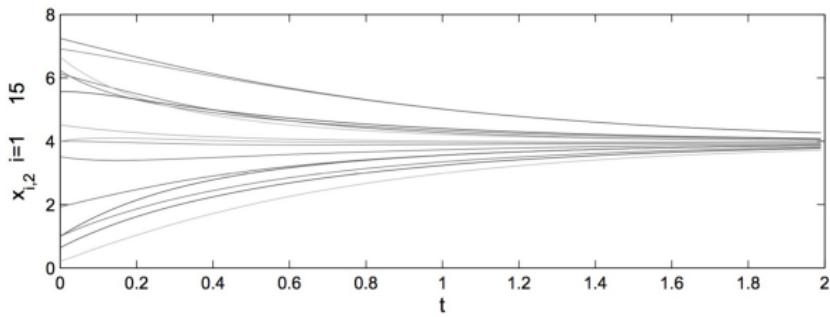
**depends only** on  $|\mathcal{N}_i|$  (number of comm. neighbors) and not on  $n$  (number of robots)

- a control law that is not decentralized is not **scalable**

# Examples of Decentralized Algorithms I

## Computing the Average

- Each robot  $i$  knows/measures a value  $x_i$  (either constant or time-varying)
- Goal: let each robot know/track  $\frac{1}{n} \sum_{i=1}^n x_i$
- Some algorithms: average consensus<sup>6</sup>, dynamic consensus<sup>7</sup>, PI average consensus<sup>8</sup>



<sup>6</sup> R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems", *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

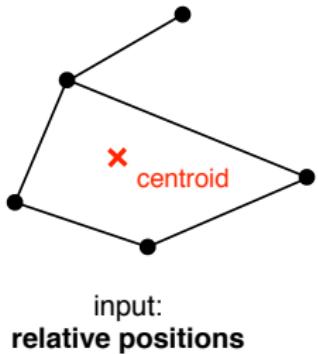
<sup>7</sup> M. Zhu and S. Martinez, "Discrete-time dynamic average consensus", *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

<sup>8</sup> R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators", in *45th IEEE Conf. on Decision and Control*, San Diego, CA, 2006, pp. 338–343.

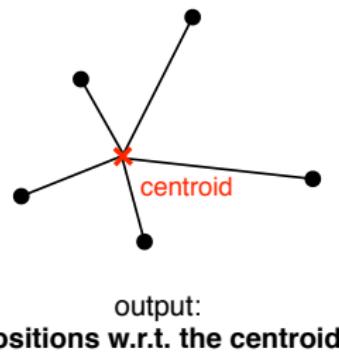
## Examples of Decentralized Algorithms II

### Computing the Relative Position w.r.t. Centroid

- Each **communicating pair** of robots  $i, j$  knows their **relative position**  $\mathbf{p}_i - \mathbf{p}_j$
- **absolute position**  $\mathbf{p}_i$  is **unknown**
- Goal: let each robot know/track  $\mathbf{p}_{iG} = \mathbf{p}_i - \frac{1}{n} \sum_{k=1}^n \mathbf{p}_k$  (relative position w.r.t the centroid)
- Algorithm: Centroid estimation<sup>9</sup>



→  
*Distributed  
centroid  
estimation*



<sup>9</sup> R. Aragues, L. Carbone, C. Sagües, and G. Calafiore, "Distributed centroid estimation from noisy relative measurements", *Systems & Control Letters*, vol. 61, no. 7, pp. 773–779, 2012.

# A Simple Useful Algorithm

Simple useful algorithm<sup>10</sup>, given the linear system

$$\dot{\mathbf{y}} = \theta \mathbf{u}$$

where

- $\mathbf{y}$  and  $\mathbf{u}$  are time-varying and **measured** (not  $\dot{\mathbf{y}}$ )
- $\theta$  is a **unknown constant parameter** to be estimated

Can be transformed in this system

$$k_f(\underbrace{\mathbf{y} - \mathbf{y}^f}_{\text{known}}) = \underbrace{\theta}_{\text{unknown}} \underbrace{\mathbf{u}^f}_{\text{known}} \quad (\diamond)$$

where

- $\mathbf{y}^f$  and  $\mathbf{u}^f$  are the **low-pass filtered**  $\mathbf{y}$  and  $\mathbf{u}$  with filter gain  $k_f$

⇒  $\theta$  can be estimated applying **online least squares** to  $(\diamond)$

<sup>10</sup> J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991, ISBN: 9780130408907.

## Section 2

### Decentralized Multi-Robot Control

System model

$$\left\{ \begin{array}{lcl} \dot{\mathbf{v}}_C & = & \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i \\ \dot{\boldsymbol{\omega}} & = & \frac{1}{J} \sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^{\perp T} \mathbf{f}_i + \frac{1}{J} (\mathbf{p}_G - \mathbf{p}_C)^{\perp T} \sum_{i=1}^n \mathbf{f}_i \end{array} \right.$$

System model

$$\left\{ \begin{array}{lcl} \dot{\mathbf{v}}_C & = & \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i \\ \dot{\omega} & = & \frac{1}{J} \sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp T \mathbf{f}_i + \frac{1}{J} (\mathbf{p}_G - \mathbf{p}_C)^\perp T \sum_{i=1}^n \mathbf{f}_i \end{array} \right.$$

## Partially Decentralized Control Law

Given a **load desired trajectory**  $\mathbf{v}_C^d(t)$  and  $\omega^d(t)$ , each robot  $i$ ,  $i = 1, \dots, n \geq 2$ , applies

$$\mathbf{f}_i = \frac{m}{n} \mathbf{u}_C + \frac{Ju_\omega - m(\mathbf{p}_G - \mathbf{p}_C)^\perp T \mathbf{u}_C}{\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2} (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp \quad \text{where}$$

$$\begin{aligned} \mathbf{u}_C &= \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C) \\ u_\omega &= \dot{\omega}^d + k_\omega(\omega^d - \omega) \end{aligned}$$

System model

$$\left\{ \begin{array}{lcl} \dot{\mathbf{v}}_C & = & \frac{1}{m} \sum_{i=1}^n \mathbf{f}_i \\ \dot{\omega} & = & \frac{1}{J} \sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp T \mathbf{f}_i + \frac{1}{J} (\mathbf{p}_G - \mathbf{p}_C)^\perp T \sum_{i=1}^n \mathbf{f}_i \end{array} \right.$$

## Partially Decentralized Control Law

Given a **load desired trajectory**  $\mathbf{v}_C^d(t)$  and  $\omega^d(t)$ , each robot  $i$ ,  $i = 1, \dots, n \geq 2$ , applies

$$\mathbf{f}_i = \frac{m}{n} \mathbf{u}_C + \frac{Ju_\omega - m(\mathbf{p}_G - \mathbf{p}_C)^\perp T \mathbf{u}_C}{\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2} (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp \quad \text{where}$$

$$\begin{aligned} \mathbf{u}_C &= \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C) \\ u_\omega &= \dot{\omega}^d + k_\omega(\omega^d - \omega) \end{aligned}$$

Plugging  $\mathbf{f}_i$  into the load dynamics, and exploiting  $\sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp = \mathbf{0}$ , we obtain

$$\dot{\mathbf{v}}_C = \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C)$$

$$\dot{\omega} = \dot{\omega}^d + k_\omega(\omega^d - \omega)$$

which implies

$$\mathbf{v}_C \rightarrow \mathbf{v}_C^d(t)$$

$$\omega \rightarrow \omega^d(t)$$

# Partially Decentralized Controller (proof)

Plugging the control equations into the system dynamics we obtain

$$\begin{aligned}\dot{\mathbf{v}}_C &= \sum_{i=1}^n \frac{\mathbf{u}_C}{n} + \left( \frac{m^{-1} J u_\omega}{\sum_{i=1}^n \|\mathbf{z}_i\|^2} - \frac{\mathbf{z}_C^\perp{}^T \mathbf{u}_C}{\sum_{i=1}^n \|\mathbf{z}_i\|^2} \right) \underbrace{\sum_{i=1}^n \mathbf{z}_i^\perp}_{=0} = \mathbf{u}_C, \\ \dot{\omega} &= \underbrace{\frac{m}{Jn} \mathbf{u}_C \sum_{i=1}^n \mathbf{z}_i^\perp{}^T}_{=0} - \frac{m \mathbf{z}_C^\perp{}^T \mathbf{u}_C}{J \sum_{i=1}^n \|\mathbf{z}_i\|^2} \sum_{i=1}^n \mathbf{z}_i^\perp{}^T \mathbf{z}_i^\perp + \frac{u_\omega}{\sum_{i=1}^n \|\mathbf{z}_i\|^2} \sum_{i=1}^n \mathbf{z}_i^\perp \mathbf{z}_i^\perp{}^T + \\ &\quad + \underbrace{\frac{m}{Jn} \mathbf{z}_C^\perp{}^T \sum_{i=1}^n \mathbf{u}_C}_{=0} - \frac{m \mathbf{z}_C^\perp{}^T \mathbf{u}_C}{J \sum_{i=1}^n \|\mathbf{z}_i\|^2} \underbrace{\mathbf{z}_C^\perp{}^T \sum_{i=1}^n \mathbf{z}_i^\perp}_{=0} + \frac{u_\omega}{\sum_{i=1}^n \|\mathbf{z}_i\|^2} \underbrace{\mathbf{z}_C^\perp{}^T \sum_{i=1}^n \mathbf{z}_i^\perp}_{=0} = \\ &= -\frac{m \mathbf{z}_C^\perp{}^T \mathbf{u}_C}{J} + u_\omega + \frac{m \mathbf{z}_C^\perp{}^T \mathbf{u}_C}{J} = u_\omega.\end{aligned}$$

Hence, we obtain

$$\dot{\mathbf{v}}_C = \dot{\mathbf{v}}_C^d + k_v (\mathbf{v}_C^d - \mathbf{v}_C), \quad \dot{\omega} = \dot{\omega}^d + k_\omega (\omega^d - \omega),$$

which, in turn, implies  $\mathbf{v}_C \rightarrow \mathbf{v}_C^d(t)$  and  $\omega \rightarrow \omega^d(t)$  globally and exponentially.

Given a **load desired trajectory**  $\mathbf{v}_C^d(t)$  and  $\omega^d(t)$ , each robot  $i$ ,  $i = 1, \dots, n \geq 2$ , applies

$$\mathbf{f}_i = \frac{m}{n} \mathbf{u}_C + \frac{Ju_\omega - m(\mathbf{p}_G - \mathbf{p}_C)^\perp^T \mathbf{u}_C}{\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2} (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp \quad \text{where} \quad \begin{aligned} \mathbf{u}_C &= \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C) \\ u_\omega &= \dot{\omega}^d + k_\omega(\omega^d - \omega) \end{aligned}$$

Plugging  $\mathbf{f}_i$  into the load dynamics, and exploiting  $\sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp = \mathbf{0}$ , we obtain

$$\begin{aligned} \dot{\mathbf{v}}_C &= \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C) && \text{which implies} && \mathbf{v}_C \rightarrow \mathbf{v}_C^d(t) \\ \dot{\omega} &= \dot{\omega}^d + k_\omega(\omega^d - \omega) && && \omega \rightarrow \omega^d(t) \end{aligned}$$

- Which **quantities have to be known** to implement it?
  - $m$ ,  $J$ ,  $\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2$  (constant); and  $\mathbf{v}_C$ ,  $\omega$ ,  $\mathbf{p}_{C_i} - \mathbf{p}_G$ ,  $\mathbf{p}_G - \mathbf{p}_C$  (time varying)

Given a **load desired trajectory**  $\mathbf{v}_C^d(t)$  and  $\omega^d(t)$ , each robot  $i$ ,  $i = 1, \dots, n \geq 2$ , applies

$$\mathbf{f}_i = \frac{m}{n} \mathbf{u}_C + \frac{Ju_\omega - m(\mathbf{p}_G - \mathbf{p}_C)^\perp^T \mathbf{u}_C}{\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2} (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp \quad \text{where} \quad \begin{aligned} \mathbf{u}_C &= \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C) \\ u_\omega &= \dot{\omega}^d + k_\omega(\omega^d - \omega) \end{aligned}$$

Plugging  $\mathbf{f}_i$  into the load dynamics, and exploiting  $\sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp = \mathbf{0}$ , we obtain

$$\begin{aligned} \dot{\mathbf{v}}_C &= \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C) && \text{which implies} && \mathbf{v}_C \rightarrow \mathbf{v}_C^d(t) \\ \dot{\omega} &= \dot{\omega}^d + k_\omega(\omega^d - \omega) && && \omega \rightarrow \omega^d(t) \end{aligned}$$

- Which **quantities have to be known** to implement it?
  - $m$ ,  $J$ ,  $\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2$  (constant); and  $\mathbf{v}_C$ ,  $\omega$ ,  $\mathbf{p}_{C_i} - \mathbf{p}_G$ ,  $\mathbf{p}_G - \mathbf{p}_C$  (time varying)

**OK** their size is constant  $\Rightarrow$  it does not depend on the number of robots

Given a **load desired trajectory**  $\mathbf{v}_C^d(t)$  and  $\omega^d(t)$ , each robot  $i$ ,  $i = 1, \dots, n \geq 2$ , applies

$$\mathbf{f}_i = \frac{m}{n} \mathbf{u}_C + \frac{Ju_\omega - m(\mathbf{p}_G - \mathbf{p}_C)^\perp^T \mathbf{u}_C}{\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2} (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp \quad \text{where} \quad \begin{aligned} \mathbf{u}_C &= \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C) \\ u_\omega &= \dot{\omega}^d + k_\omega(\omega^d - \omega) \end{aligned}$$

Plugging  $\mathbf{f}_i$  into the load dynamics, and exploiting  $\sum_{i=1}^n (\mathbf{p}_{C_i} - \mathbf{p}_G)^\perp = \mathbf{0}$ , we obtain

$$\begin{aligned} \dot{\mathbf{v}}_C &= \dot{\mathbf{v}}_C^d + k_v(\mathbf{v}_C^d - \mathbf{v}_C) && \text{which implies} && \mathbf{v}_C \rightarrow \mathbf{v}_C^d(t) \\ \dot{\omega} &= \dot{\omega}^d + k_\omega(\omega^d - \omega) && && \omega \rightarrow \omega^d(t) \end{aligned}$$

- Which **quantities have to be known** to implement it?
  - $m$ ,  $J$ ,  $\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2$  (constant); and  $\mathbf{v}_C$ ,  $\omega$ ,  $\mathbf{p}_{C_i} - \mathbf{p}_G$ ,  $\mathbf{p}_G - \mathbf{p}_C$  (time varying)

**OK** their size is constant  $\Rightarrow$  it does not depend on the number of robots

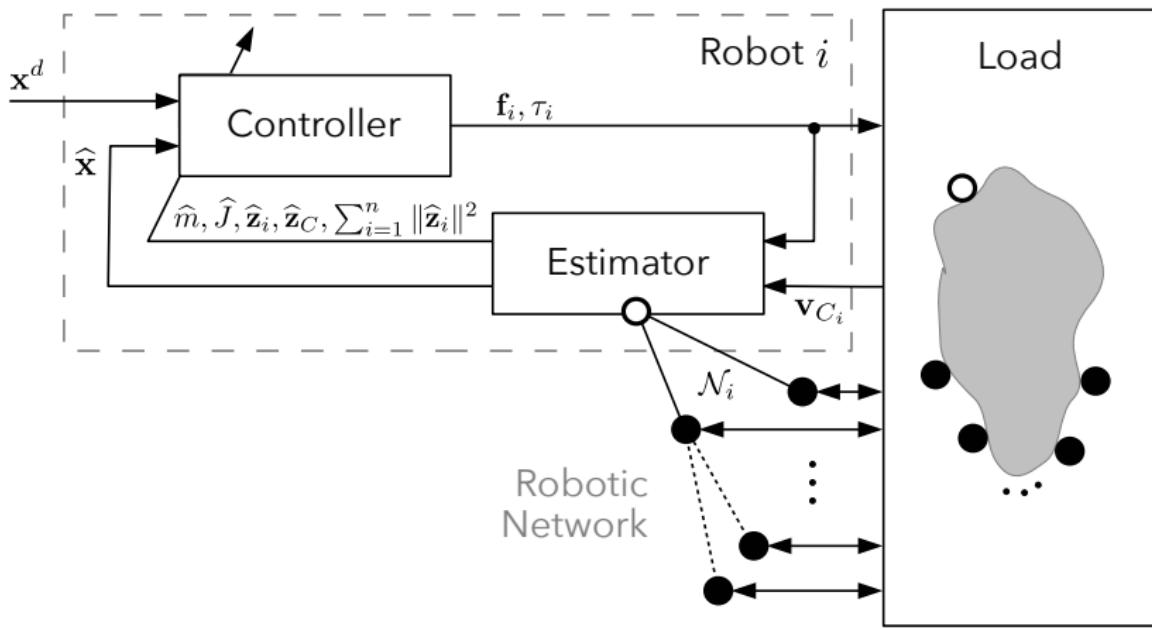
**KO** each robot  $i$  can only

1. **locally measure** the velocity  $\mathbf{v}_{C_i}$  of the contact point  $C_i$
2. **locally control** the applied **force**  $\mathbf{f}_i$
3. **communicate** with its **one-hop** neighbors  $\mathcal{N}_i$

How to make the control law **fully decentralized**?

# Decentralized Controller/Estimator Scheme

How to make the control law **fully decentralized**?



How to make the previous control law **fully decentralized**?

Problem (Decentralized Estimation for Cooperative Manipulation)

Design a **decentralized algorithm** by which each robot  $i$  estimates

1.  $m$  (constant)
2.  $v_C$  (time varying)
3.  $J$  (constant)
4.  $\omega$  (time varying)
5.  $p_{C_i} - p_G$  (time varying)
6.  $p_G - p_C$  (time varying)

Each robot  $i$  can only

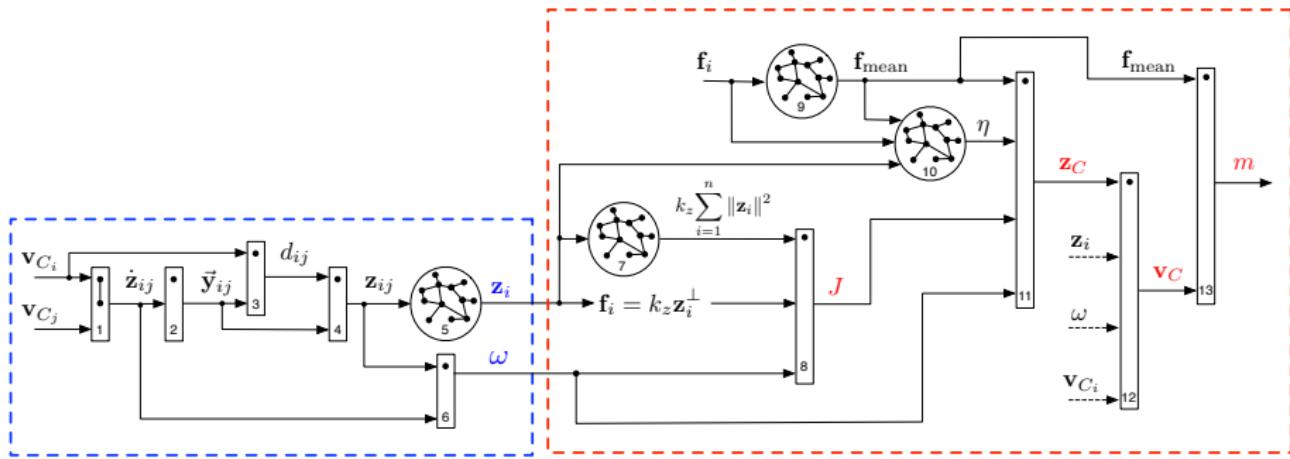
1. **locally measure** the **velocity**  $v_{C_i}$  of the contact point  $C_i$
2. **locally control** the applied **force**  $f_i$
3. **communicate** with its **one-hop** neighbors  $\mathcal{N}_i$

**Kinematic part**

- only **velocity measurements  $v_{C_i}$**  used
- only rigid body **kinematics** used
- leads to the estimation of
  - $\mathbf{z}_i = \mathbf{p}_{C_i} - \mathbf{p}_G$
  - $\omega$

**Dynamic part**

- both  $v_{C_i}$  and **force inputs  $f_i$**  used
- both rigid body **kinematics** and **dynamics** used
- leads to the estimation of
  - $J$
  - $\mathbf{z}_C = \mathbf{p}_G - \mathbf{p}_C$
  - $v_C$
  - $m$



## Kinematic Part: Estimation of $\mathbf{z}_i(t)$ and $\omega(t)$

Remember: each robot  $i$  measures only  $\dot{\mathbf{p}}_{C_i}$  and communicates only with  $j \in \mathcal{N}_i$ .

First, estimate  $\mathbf{z}_{ij} = \mathbf{p}_{C_i} - \mathbf{p}_{C_j}$ , decomposable as  $\mathbf{z}_{ij} = d_{ij}\vec{\mathbf{y}}_{ij}$

- $\vec{\mathbf{y}}_{ij}$  (time varying): axis along which  $\mathbf{z}_{ij}$  lies  
using rigid body constraint  $\Rightarrow \vec{\mathbf{y}}_{ij} = \frac{\dot{\mathbf{z}}_{ij}^\perp}{\|\dot{\mathbf{z}}_{ij}^\perp\|} = \frac{(\dot{\mathbf{p}}_{C_i} - \dot{\mathbf{p}}_{C_j})^\perp}{\|\dot{\mathbf{p}}_{C_i} - \dot{\mathbf{p}}_{C_j}\|} \rightarrow$  '1-hop' computable
- $d_{ij}$  (constant, a part from sign): coordinate of  $\mathbf{z}_{ij}$  along  $\vec{\mathbf{y}}_{ij}$   
differentiating:  $\dot{\mathbf{z}}_{ij} = d_{ij} \frac{d}{dt} \vec{\mathbf{y}}_{ij} \rightarrow d_{ij}$  is the only constant unknown: online filtered linear least squares<sup>11</sup> (LLS)

---

<sup>11</sup> J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991, ISBN: 9780130408907.

<sup>12</sup> R. Aragues, L. Carbone, C. Sagüés, and G. Calafiori, "Distributed centroid estimation from noisy relative measurements", *Systems & Control Letters*, vol. 61, no. 7, pp. 773–779, 2012.

## Kinematic Part: Estimation of $\mathbf{z}_i(t)$ and $\omega(t)$

Remember: each robot  $i$  measures only  $\dot{\mathbf{p}}_{C_i}$  and communicates only with  $j \in \mathcal{N}_i$ .

First, estimate  $\mathbf{z}_{ij} = \mathbf{p}_{C_i} - \mathbf{p}_{C_j}$ , decomposable as  $\mathbf{z}_{ij} = d_{ij}\vec{\mathbf{y}}_{ij}$

- $\vec{\mathbf{y}}_{ij}$  (time varying): axis along which  $\mathbf{z}_{ij}$  lies

using rigid body constraint  $\Rightarrow \vec{\mathbf{y}}_{ij} = \frac{\dot{\mathbf{z}}_{ij}^\perp}{\|\dot{\mathbf{z}}_{ij}^\perp\|} = \frac{(\dot{\mathbf{p}}_{C_i} - \dot{\mathbf{p}}_{C_j})^\perp}{\|\dot{\mathbf{p}}_{C_i} - \dot{\mathbf{p}}_{C_j}\|}$  → '1-hop' computable

- $d_{ij}$  (constant, a part from sign): coordinate of  $\mathbf{z}_{ij}$  along  $\vec{\mathbf{y}}_{ij}$

differentiating:  $\dot{\mathbf{z}}_{ij} = d_{ij} \frac{d}{dt} \vec{\mathbf{y}}_{ij} \rightarrow d_{ij}$  is the only constant unknown: online filtered linear least squares<sup>11</sup> (LLS)

Then, estimate

1.  $\mathbf{z}_i(t)$  using distributed centroid estimation<sup>12</sup> from relative positions  $\mathbf{z}_{ij}$
2.  $\omega(t)$  using  $\omega = -\frac{\mathbf{z}_{ij}^T \dot{\mathbf{z}}_{ij}^\perp}{\mathbf{z}_{ij}^T \mathbf{z}_{ij}}$ . Distributed averaging possible to decrease the noise

<sup>11</sup> J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991, ISBN: 9780130408907.

<sup>12</sup> R. Aragues, L. Carbone, C. Sagüés, and G. Calafiori, "Distributed centroid estimation from noisy relative measurements", *Systems & Control Letters*, vol. 61, no. 7, pp. 773–779, 2012.

Remember:  $J > 0$  is the constant rotational inertia

1. Each robot **computes the constant**  $\sum_{i=1}^n \|\mathbf{z}_i\|^2$  using any **consensus**<sup>13</sup> algorithm
2. Each robot  $i$  **applies a force**  $\mathbf{f}_i = k_z \mathbf{z}_i^\perp$ , being  $k_z$  any constant
3. The **rotational dynamics simplifies** in

$$\dot{\omega} = \frac{1}{J} \sum_{i=1}^n \mathbf{z}_i^\perp{}^T \mathbf{f}_i + \frac{1}{J} (\mathbf{p}_G - \mathbf{p}_C)^\perp{}^T \underbrace{\sum_{i=1}^n \mathbf{f}_i}_{=0} = J^{-1} k_z \sum_{i=1}^n \|\mathbf{z}_i\|^2$$

4. **Compute locally the constant**  $J^{-1}$  with **online filtered LLS**<sup>14</sup> using
  - o  $\omega(t)$  (estimated)
  - o  $k_z \sum_{i=1}^n \|\mathbf{z}_i\|^2$  (previously computed at step 1)

<sup>13</sup> R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems", *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

<sup>14</sup> J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991, ISBN: 9780130408907.

## Dynamic Part: Estimation of $\mathbf{z}_C(t)$

Remember:  $\mathbf{z}_C(t) = \mathbf{p}_G - \mathbf{p}_C$  position of the center of the contact points w.r.t. CoM

Rewrite the **rotational dynamics** as:  $\dot{\omega} = \mathbf{z}_C^\perp{}^T \bar{\mathbf{f}} + \eta$  ( $\square$ ), where

- $\bar{\mathbf{f}} = J^{-1} \sum_{i=1}^n \mathbf{f}_i = J^{-1} n \mathbf{f}_{\text{mean}}$   $\Rightarrow$  both  $\bar{\mathbf{f}}$  and  $\eta$  are **locally known** using **dynamic consensus**
- $\eta = J^{-1} \sum_{i=1}^n \mathbf{z}_i^\perp{}^T (\mathbf{f}_i - \mathbf{f}_{\text{mean}})$

**Rigid body constraint** implies:  $\dot{\mathbf{z}}_C^\perp = -\mathbf{z}_C \omega$  ( $\triangle$ )

Stacking ( $\square$ ) and ( $\triangle$ )

$$\begin{cases} \dot{x}_1 &= -x_2 x_3 \\ \dot{x}_2 &= x_1 x_3 \\ \dot{x}_3 &= x_1 u_2 - x_2 u_1 + u_3 \\ y &= x_3 \end{cases} \quad \text{where}$$

- $\mathbf{z}_C = [z_C^x \ z_C^y]^T = [x_1 \ x_2]^T$  (non-measured state)
- $\omega = x_3$  (measured state)
- $\bar{\mathbf{f}} = [\bar{f}_x \ \bar{f}_y]^T = [u_1 \ u_2]^T$  (known input)
- $\eta = u_3$  (known input)

Estimate  $\mathbf{z}_C \Leftrightarrow$  **observe**  $x_1, x_2$  in using  $y, u_1, u_2, u_3$

## Dynamic Part: Estimation of $\mathbf{z}_C(t)$ (cont.)

Estimate  $\mathbf{z}_C \Leftrightarrow$  **observe**  $x_1, x_2$  in  $\begin{cases} \dot{x}_1 = -x_2 x_3 \\ \dot{x}_2 = x_1 x_3 \\ \dot{x}_3 = x_1 u_2 - x_2 u_1 + u_3 \\ y = x_3 \end{cases}$  (\*), using  $y, u_1, u_2, u_3$

---

<sup>14</sup> R. Hermann and A. J. Krener, "Nonlinear controllability and observability", *IEEE Trans. on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

Estimate  $\mathbf{z}_C \Leftrightarrow$  observe  $x_1, x_2$  in  $\begin{cases} \dot{x}_1 = -x_2 x_3 \\ \dot{x}_2 = x_1 x_3 \\ \dot{x}_3 = x_1 u_2 - x_2 u_1 + u_3 \\ y = x_3 \end{cases}$  (\*), using  $y, u_1, u_2, u_3$

## Observability

The system (\*) is locally observable in the sense of Hermann and Krener<sup>14</sup> if

- $x_3(t) = \omega(t) \not\equiv 0$  and
- $[\bar{f}_x(t) \quad \bar{f}_y(t)]^T = [u_1(t) \quad u_2(t)]^T \not\equiv \mathbf{0}^T$

---

<sup>14</sup> R. Hermann and A. J. Krener, "Nonlinear controllability and observability", *IEEE Trans. on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

Estimate  $\mathbf{z}_C \Leftrightarrow$  observe  $x_1, x_2$  in  $\begin{cases} \dot{x}_1 = -x_2 x_3 \\ \dot{x}_2 = x_1 x_3 \\ \dot{x}_3 = x_1 u_2 - x_2 u_1 + u_3 \\ y = x_3 \end{cases}$  (\*), using  $y, u_1, u_2, u_3$

## Observability

The system (\*) is locally observable in the sense of Hermann and Krener<sup>14</sup> if

- $x_3(t) = \omega(t) \not\equiv 0$  and
- $[\bar{f}_x(t) \quad \bar{f}_y(t)]^T = [u_1(t) \quad u_2(t)]^T \not\equiv \mathbf{0}^T$

## Decentralized Nonlinear Observer

$$\begin{cases} \dot{\hat{x}}_1 = -\hat{x}_2 x_3 + u_2(x_3 - \hat{x}_3) \\ \dot{\hat{x}}_2 = \hat{x}_1 x_3 - u_1(x_3 - \hat{x}_3) \\ \dot{\hat{x}}_3 = \hat{x}_1 u_2 - \hat{x}_2 u_1 + k_e(x_3 - \hat{x}_3) + u_3 \end{cases}$$

is a  
**decentralized**  
asymptotic  
**observer** iff

- $k_e > 0$
- $x_3 \not\equiv 0$
- $[u_1 \quad u_2]^T \not\equiv \mathbf{0}$

---

<sup>14</sup> R. Hermann and A. J. Krener, "Nonlinear controllability and observability", *IEEE Trans. on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

# Dynamic Part: Observability of $\mathbf{z}_C(t)$

## Observability

The system  $(\star)$  is locally observable in the sense of Hermann and Krener<sup>14</sup> if

- $x_3(t) = \omega(t) \not\equiv 0$  and
- $[\bar{f}_x(t) \quad \bar{f}_y(t)]^T = [u_1(t) \quad u_2(t)]^T \not\equiv \mathbf{0}^T$

## Proof.

The observability matrix<sup>14</sup> is  $O(\mathbf{x}) = \begin{pmatrix} dL_f^0(y) \\ dL_f^1(y) \\ dL_f^2(y) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ \bar{u}_2 & -u_1 & 0 \\ -\bar{u}_1 x_3 & -u_2 x_3 & -u_1 x_1 - u_2 x_2 \end{pmatrix}$  and

$\det(O) = -x_3(u_1^2 + u_2^2)$ . Locally observability iff  $\det(O) \neq 0$

□

## Recap

$\mathbf{z}_C$  is observable from local velocity measurements if and only if

- the rotational rate of the object and
- the average vector of the applied forces

are not identically zero

<sup>14</sup> R. Hermann and A. J. Krener, "Nonlinear controllability and observability", *IEEE Trans. on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

## Theorem

*The following dynamical system*

$$\begin{aligned}\dot{\hat{x}}_1 &= -\hat{x}_2 x_3 + u_2(x_3 - \hat{x}_3) \\ \dot{\hat{x}}_2 &= \hat{x}_1 x_3 - u_1(x_3 - \hat{x}_3) \\ \dot{\hat{x}}_3 &= \hat{x}_1 u_2 - \hat{x}_2 u_1 + k_e(x_3 - \hat{x}_3) + u_3\end{aligned}\tag{1}$$

*is an asymptotic observer for the system  $(*)$  if*

- $k_e > 0$
- $x_3 \not\equiv 0$
- $[u_1 \ u_2]^T \not\equiv \mathbf{0}$

# Observer of $\mathbf{z}_C(t)$

Proof.

Error vector  $\mathbf{e} = (e_1 \ e_2 \ e_3)^T = ((x_1 - \hat{x}_1) \ (x_2 - \hat{x}_2) \ (x_3 - \hat{x}_3))^T$ ; the error dynamics is

$$\dot{\mathbf{e}} = \begin{pmatrix} 0 & -x_3 & -u_2 \\ x_3 & 0 & u_1 \\ u_2 & -u_1 & -k_e \end{pmatrix} \mathbf{e} \quad (2)$$

Candidate Lyapunov function:

$$V(\mathbf{e}) = \frac{1}{2} (e_1^2 + e_2^2 + e_3^2),$$

whose time derivative along the system trajectories is

$$\begin{aligned} \dot{V}(\mathbf{e}) &= e_1(-e_2 x_3 - u_2 e_3) + e_2(x_3 e_1 + u_1 e_3) + e_3(u_2 e_1 - u_1 e_2 - k_e e_3) \\ &= -e_1 e_2 x_3 - e_1 u_2 e_3 + e_2 x_3 e_1 + e_2 u_1 e_3 + e_3 u_2 e_1 - e_3 u_1 e_2 - k_e e_3^2 = -k_e e_3^2, \end{aligned}$$

which is negative semidefinite.

Considering the set  $\mathcal{V} = \left\{ \mathbf{e} \text{ s.t. } \dot{V}(\mathbf{e}) = 0 \right\} = \{ \mathbf{e} \text{ s.t. } e_3 = 0 \}$ , and generic vector  $\bar{\mathbf{e}} = (\bar{e}_1 \ \bar{e}_2 \ 0)^T \in \mathcal{V}$ , it is easy to verify by means of (2) that the first and second time derivatives of  $e_3$  along a trajectory containing  $\bar{\mathbf{e}}$  are given by

$$\frac{de_3}{dt} \Big|_{\bar{\mathbf{e}}} = u_2 \bar{e}_1 - u_1 \bar{e}_2 \quad \frac{d^2 e_3}{dt^2} \Big|_{\bar{\mathbf{e}}} = \frac{de_1}{dt} \Big|_{\bar{\mathbf{e}}} = -x_3(u_2 \bar{e}_2 + u_1 \bar{e}_1).$$

Therefore, if  $x_3$  is not vanishing, and  $u_1, u_2$  are not simultaneously zero then the largest invariant set  $M \subset \mathcal{V}$  consists of the only equilibrium point  $(0 \ 0 \ 0)^T$ . Thus, the thesis holds due to the invariance Krasovskii–LaSalle principle. □

## Estimation of $\mathbf{v}_C(t)$

$\mathbf{v}_C(t)$  is **computed locally** by robot  $i$  exploiting the **rigid body constraint**

$$\mathbf{v}_C(t) = \mathbf{v}_{C_i}(t) - \omega(t)(\mathbf{z}_C(t) + \mathbf{z}_i(t))$$

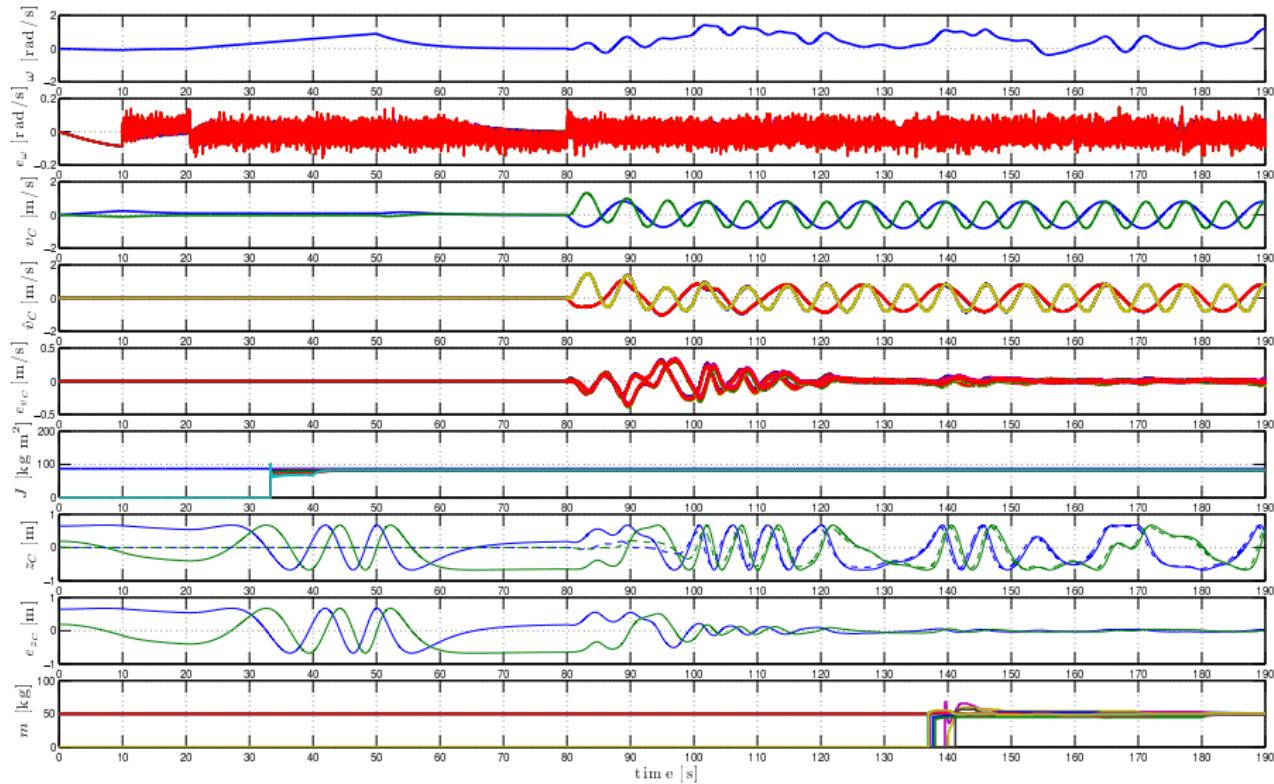
## Estimation of $m$

Rewriting the **translational dynamics** as  $\frac{d}{dt} \mathbf{v}_C = m^{-1} \cdot n \mathbf{f}_{\text{mean}}$ , where

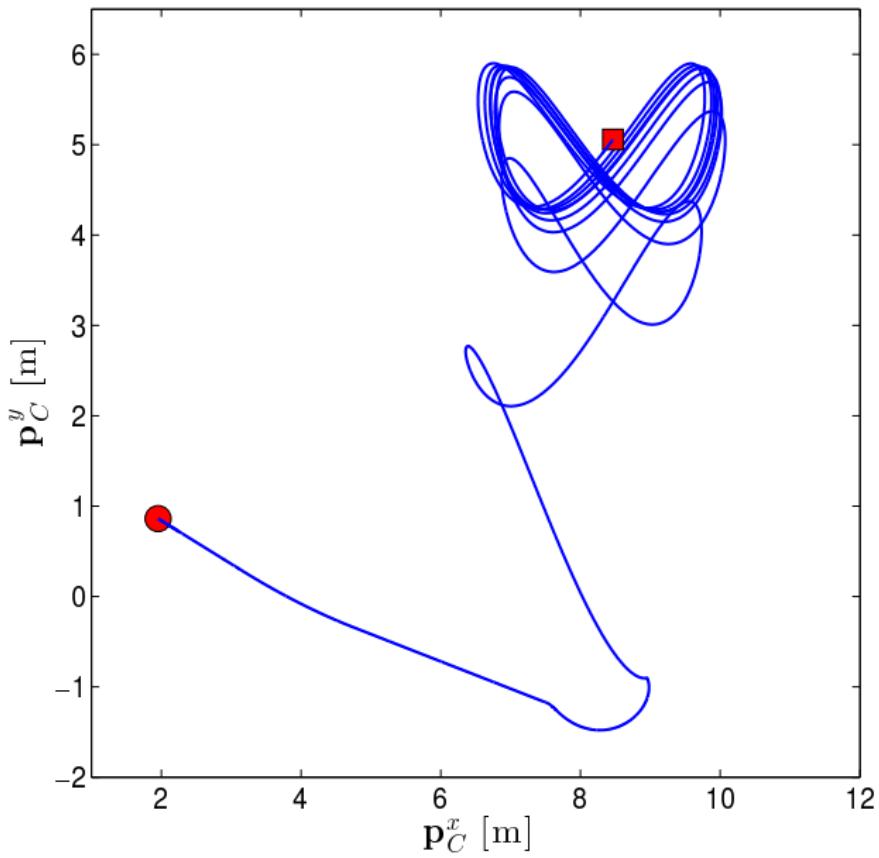
- $\mathbf{v}_C$  locally known
- $\mathbf{f}_{\text{mean}}$  locally known

→ Compute  $m^{-1}$  with **online filtered LLS**

# Simulation (with Noise)



## Simulation (with Noise)



## Simulation (with Noise)

## Simulation (with Noise)

## Section 3

### Summary and Open Problems

# Summary: Contribution and Methodology

Control and online estimation for cooperative mobile manipulation:

Main **limitations** in the state of the art

- **centralization**
- need for **acceleration** measurements
- **absolute positioning** measurements

**Contributions** of the method

- totally **decentralized**
- **all the quantities** (constant and time-varying) estimated in one framework
- use of **velocity-only measurements**
  - no accelerations
  - no relative/absolute positions

## Methodology

- geometrical and dynamical analysis
- adaptive control
- nonlinear observer
- consensus/centroid algorithms

<sup>15</sup> A. Franchi, A. Petitti, and A. Rizzo, "Distributed estimation of the inertial parameters of an unknown load via multi-robot manipulation", in *53rd IEEE Conf. on Decision and Control*, Los Angeles, CA, 2014, pp. 6111–6116.

<sup>16</sup> A. Franchi, A. Petitti, and A. Rizzo, "Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements", in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, 2015, pp. 5517–5522.

<sup>17</sup> A. Petitti, A. Franchi, D. Di Paola, and A. Rizzo, "Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators", in *2016 IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, 2016, pp. 441–446.

- **Adaptiveness** and **decentralization** are **fundamental properties** for a multi-robot system
- **Methodology used** to design a decentralized controller:
  1. design a **partially decentralized control law** that is implementable with
    - local or 1-hop communicable **measurements/known** quantities
    - a fixed number of **global quantities**
  2. design a **decentralized estimator** to retrieve online the **global quantities**, possibly exploiting the available algorithms like, e.g.,
    - consensus algorithms
    - centroid estimation

- **Adaptiveness** and **decentralization** are **fundamental properties** for a multi-robot system
- **Methodology used** to design a decentralized controller:
  1. design a **partially decentralized control law** that is implementable with
    - local or 1-hop communicable **measurements/known** quantities
    - a fixed number of **global quantities**
  2. design a **decentralized estimator** to retrieve online the **global quantities**, possibly exploiting the available algorithms like, e.g.,
    - consensus algorithms
    - centroid estimation

The same methodology is usable for other problems, such as...

---

<sup>17</sup> P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülfhoff, "A passivity-based decentralized strategy for generalized connectivity maintenance", *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

<sup>17</sup> T. Nestmeyer, P. Robuffo Giordano, H. H. Bülfhoff, and A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots", *Autonomous Robots*, 2016.

---

<sup>17</sup> D. Zelazo, A. Franchi, H. H. Bülfhoff, and P. Robuffo Giordano, "Decentralized rigidity maintenance control with range measurements for multi-robot systems", *The International Journal of Robotics Research*, vol. 34, no. 1, pp. 105–128, 2014.

## Future works on Cooperative Manipulation

- extension to **underactuated robots, contact constraints** and **aerial robots**
- extension to **3D case**

Open issue in the used methodology (Partially Decentralized Controller + Decentralized Estimator)

- decentralized tracking of **highly dynamic quantities**

# Acknowledgements

For related work, visit <http://homepages.laas.fr/afranchi/robotics/>

joint work with:

**Antonio Petitti**  
(PostDoc at ISSIA-CNR)



**Donato Di Paola**  
(Researcher at ISSIA-CNR)



**Alessandro Rizzo**  
(Prof. at Politecnico di Torino)

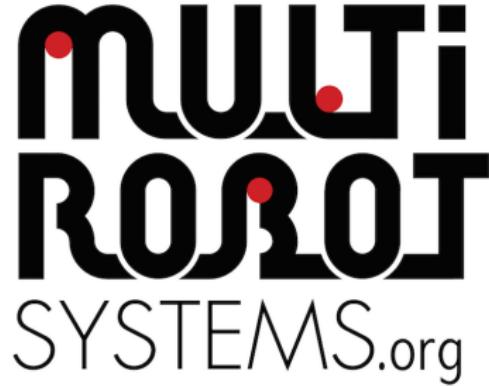


**Questions?**

## IEEE RAS Technical Committee on **Multi-Robot Systems**:

<http://multirobotsystems.org/>

- recently founded (Fall 2014)
- 260 members
- identifying and constantly tracking the **common characteristics, problems, and achievements** of multi-robot systems research in its several and diverse domains
  - robotics
  - automatic control
  - telecommunications
  - computer science / AI
  - optimization
  - ...



If you work/are interested on multi-robot/agent systems then **become a member!**  
<http://multirobotsystems.org/?q=user/register>