Decentralized Control for Cooperative 2D Mobile Manipulation

A paradigmatic Example of Decentralization in Multi-Robot Control

Antonio Franchi

CNRS, LAAS, France, Europe

JNRR 2015, Baie de Somme

22nd October, 2015

From more info on the methods presented in the following:

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

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

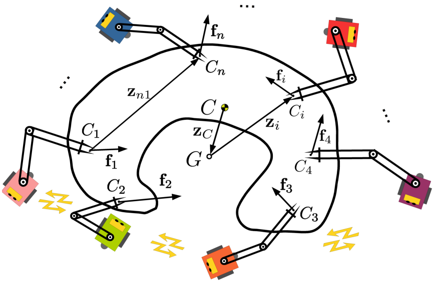
A. Petitti, A. Franchi, D. Di Paola, *et al.*, "Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators", in *Under Review*

Table of Contents

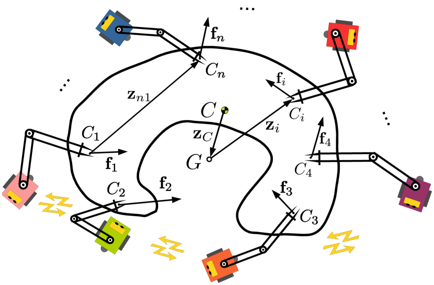
Section 1

Motivations and Modeling

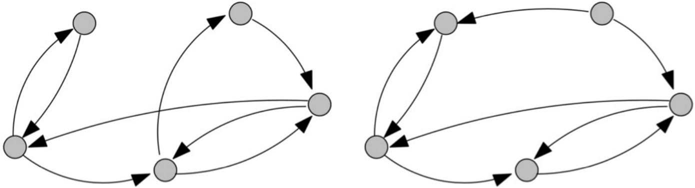
1. Present a novel **decentralized control method** for **cooperative manipulation** with a **team of mobile robots**



1. Present a novel **decentralized control method** for **cooperative manipulation** with a **team of mobile robots**



2. Show a **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**

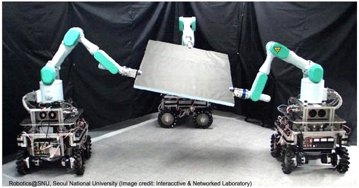
Some **tasks**

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

Some **problems**/**methods**

- **motion planning** [1]
- **formation control** [2]
- **optimal** control [3]

MuRoLa - Multi-robot laboratory (Image credit: S. Hirche @ Techn. Universität München)



Robotics@SNU, Seoul National University (Image credit: Interacctive & Networked Laboratory)

[1] A. Yamashita, T. Arai, J. Ota, *et al.*, "Motion planning of multiple mobile robots for cooperative manipulation and transportation", *IEEE Trans. on Robotics*, vol. 19, no. 2, pp. 223–237, 2003.
[2] A. Yufka, O. Parlaktuna, and M. Ozkan, "Formation-based cooperative transportation by a group of nonholonomic mobile robots", in *2010 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Istanbul, Turkey, 2010, pp. 3300–3307.
[3] 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.

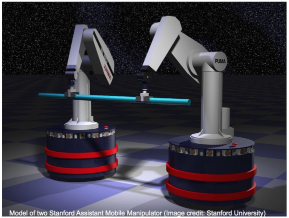
Why retrieving the manipulandum parameters...

...**online**: [4] [5]

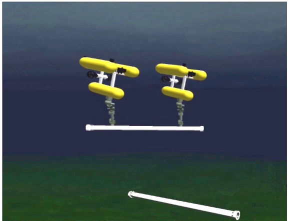
- less **control effort**
- use of **force control**
- **time-varying** manipulation tasks

...**distributively**:

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



Model of two Stanford Assistant Mobile Manipulator (Image credit: Stanford University)



MARIS Project (Image credit: GRAAL/University of Genova)

[4] Y. Yong, T. Arima, and S. Tsujio, "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.

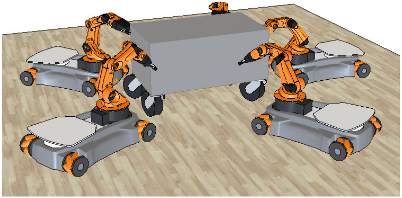
[5] D. Kubus, T. Kroger, and F. M. Wahl, "On-line estimation of inertial parameters using a recursive total least-squares approach", in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, France, 2008, pp. 3845–3852.

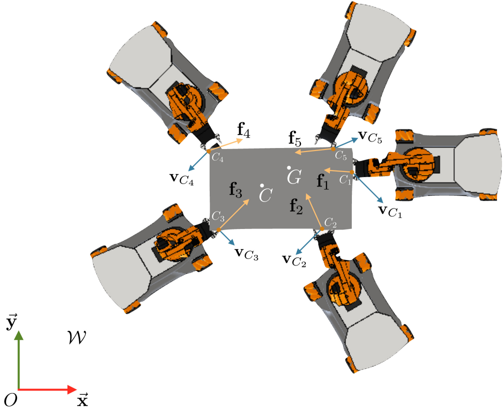
n **mobile manipulators on a plane**

- $\mathcal{G} = \{\mathcal{I}, \mathcal{E}\}$ 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** $C_i \in B$
  - negligible torque
- **measures** the **velocity** $\dot{\mathbf{p}}_{C_i}$ of C_i
- **anything else** is **unknown**





Model: Load

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

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 $\mathbf{f}_1, \ldots, \mathbf{f}_n$

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 $i = 1 \ldots n$

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

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

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

$J \in \mathbb{R}_{>0}$ **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 $i = 1 \dots n$

$(\cdot)^\perp$ 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

Control Objective

System model

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

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

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

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

using

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

Control Objective

System model

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

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

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

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

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 $\mathbf{v}_C(t)$, $\omega(t)$ (quantities to be controlled)
- **without knowledge** of the **parameters** of the dynamical system

Model: Decentralized Control Law

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**

Model: Decentralized Control Law

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

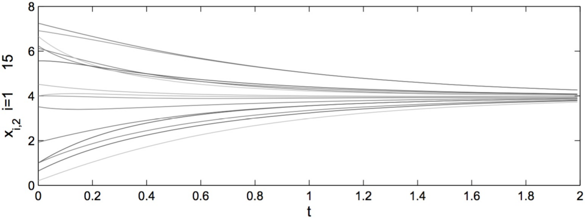
- **communication** bandwidth
- **computation** time (per step)
- **memory** used (inputs, outputs, local variables)

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**

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**[6], **dynamic consensus**[7], **PI average consensus**[8]



[6] 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.

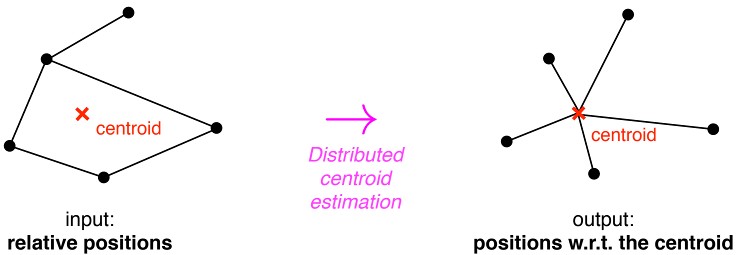
[7] M. Zhu and S. Martinez, "Discrete-time dynamic average consensus", *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[8] 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 $\mathbf{p}_{iG} = \mathbf{p}_i - \frac{1}{n}\sum_{i=k}^{n} \mathbf{p}_k$ (relative position w.r.t the centroid)
- Algorithm: Centroid estimation[9]



[9] R. Aragues, L. Carlone, C. Sagues, *et al.*, "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[10], given the linear system

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

where

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

Can be transformed in this system

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

where

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

$\Rightarrow$ θ can be estimated applying **online least squares** to $(\Diamond)$

[10] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991, ISBN: 9780130408907.

Section 2

Decentralized Multi-Robot Control

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

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

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

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

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

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

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

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

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

- Which **quantities have to be known** to implement it?
  - m, J, $\sum_i^n \|\mathbf{p}_{C_i} - \mathbf{p}_C\|^2$ (constant); and $\mathbf{v}_C$, ω, $\mathbf{p}_{C_i} - \mathbf{p}_G$, $\mathbf{p}_G - \mathbf{p}_C$ (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, $i = 1, \ldots, n \geq 2$, applies

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

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

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

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

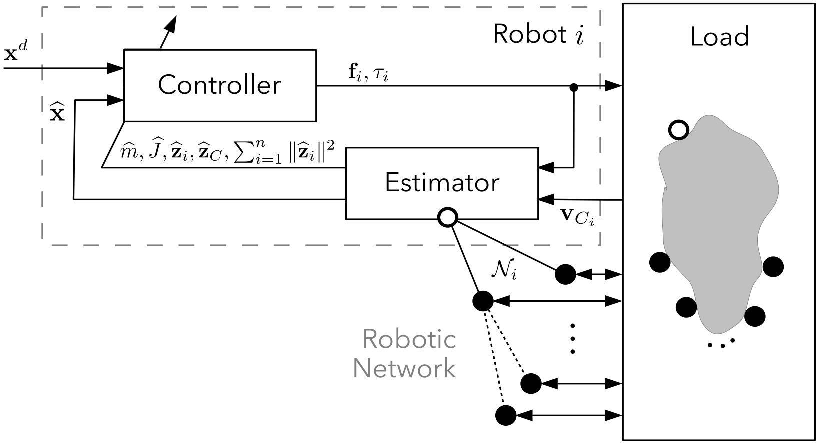
OK their size is constant $\Rightarrow$ it does not depend on the number of robots

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**?

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



Decentralized Estimation Problem

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. $\mathbf{v}_C$ *(time varying)*
3. J *(constant)*
4. ω *(time varying)*
5. $\mathbf{p}_{C_i} - \mathbf{p}_G$ *(time varying)*
6. $\mathbf{p}_G - \mathbf{p}_C$ *(time varying)*

Each robot i can only

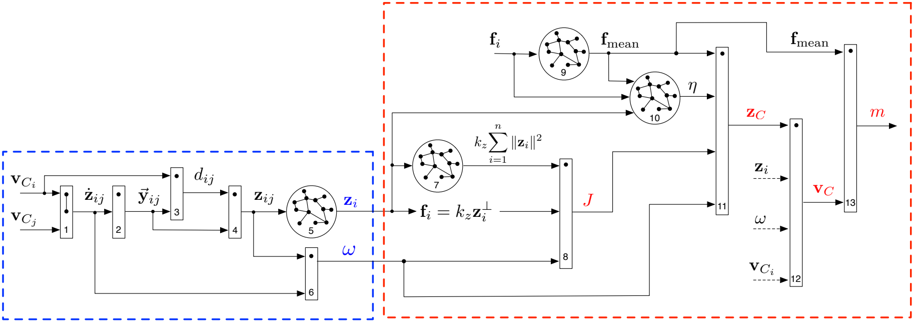
1. **locally measure** *the* **velocity** $\mathbf{v}_{C_i}$ *of the contact point C_i*
2. **locally control** *the applied* **force** $\mathbf{f}_i$
3. **communicate** *with its* **one-hop** *neighbors $\mathcal{N}_i$*

Kinematic part

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

Dynamic part

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



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 $\Rightarrow \vec{\mathbf{y}}_{ij} = \dfrac{\dot{\mathbf{z}}_{ij}^{\perp}}{\|\dot{\mathbf{z}}_{ij}^{\perp}\|} = \dfrac{(\dot{\mathbf{p}}_{C_i} - \dot{\mathbf{p}}_{C_j})^{\perp}}{\|\dot{\mathbf{p}}_{C_i} - \dot{\mathbf{p}}_{C_j}\|} \rightarrow$ '1-hop' computable
- d_{ij} (constant, a part from sign): **coordinate** of $\mathbf{z}_{ij}$ along $\vec{\mathbf{y}}_{ij}$
  differentiating: $\dot{\mathbf{z}}_{ij} = d_{ij} \dfrac{\mathrm{d}}{\mathrm{dt}}\vec{\mathbf{y}}_{ij} \rightarrow d_{ij}$ is the only constant unknown: online filtered **linear least squares**[11] (LLS)

[11] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991, ISBN: 9780130408907.
[12] R. Aragues, L. Carlone, C. Sagues, *et al.*, "Distributed centroid estimation from noisy relative measurements", *Systems & Control Letters*, vol. 61, no. 7, pp. 773–779, 2012.

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 $\Rightarrow \vec{\mathbf{y}}_{ij} = \dfrac{\dot{\mathbf{z}}_{ij}^{\perp}}{\|\dot{\mathbf{z}}_{ij}^{\perp}\|} = \dfrac{(\dot{\mathbf{p}}_{C_i} - \dot{\mathbf{p}}_{C_j})^{\perp}}{\|\dot{\mathbf{p}}_{C_i} - \dot{\mathbf{p}}_{C_j}\|} \rightarrow$ '1-hop' computable
- d_{ij} (constant, a part from sign): **coordinate** of $\mathbf{z}_{ij}$ along $\vec{\mathbf{y}}_{ij}$
  differentiating: $\dot{\mathbf{z}}_{ij} = d_{ij}\,\dfrac{\mathrm{d}}{\mathrm{dt}}\vec{\mathbf{y}}_{ij} \rightarrow d_{ij}$ is the only constant unknown: online filtered **linear least squares**[11] (LLS)

Then, estimate

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

[11] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991, ISBN: 9780130408907.
[12] R. Aragues, L. Carlone, C. Sagues, *et al.*, "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**[13] 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**[14] using
   - $\omega(t)$ (estimated)
   - $k_z \sum_{i=1}^{n} \|\mathbf{z}_i\|^2$ (previously computed at step 1)

[13] 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.
[14] 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: $\dot{\omega} = {\mathbf{z}_C^\perp}^T \bar{\mathbf{f}} + \eta \quad (\square)$, where

- $\bar{\mathbf{f}} = J^{-1} \sum_{i=1}^n \mathbf{f}_i = J^{-1} n\, \mathbf{f}_{\text{mean}}$
- $\eta = J^{-1} \sum_{i=1}^n {\mathbf{z}_i^\perp}^T (\mathbf{f}_i - \mathbf{f}_{\text{mean}})$

$\Rightarrow$ both $\bar{\mathbf{f}}$ and η are **locally known** using **dynamic consensus**

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

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

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

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 $\begin{cases} \dot{x}_1 &= -x_2 x_3 \\ \dot{x}_2 &= x_1 x_3 \\ \dot{x}_3 &= x_1 u_2 - x_2 u_1 + u_3 \\ y &= x_3 \end{cases}$ $(\star)$, using y, u_1, u_2, u_3

[14] 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 $\begin{cases} \dot{x}_1 &= -x_2 x_3 \\ \dot{x}_2 &= x_1 x_3 \\ \dot{x}_3 &= x_1 u_2 - x_2 u_1 + u_3 \\ y &= x_3 \end{cases}$ $(\star)$, using y, u_1, u_2, u_3

Observability

The system $(\star)$ is locally observable in the sense of *Hermann and Krener* [14] if

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

[14] 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 $\begin{cases} \dot{x}_1 = -x_2 x_3 \\ \dot{x}_2 = x_1 x_3 \\ \dot{x}_3 = x_1 u_2 - x_2 u_1 + u_3 \\ y = x_3 \end{cases}$ $(\star)$, using y, u_1, u_2, u_3

Observability

The system $(\star)$ is locally observable in the sense of *Hermann and Krener* [14] if

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

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 'global' **decentralized** asymptotic **observer** iff

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

[14] 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: Estimation of $\mathbf{v}_C(t)$ and m

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

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

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

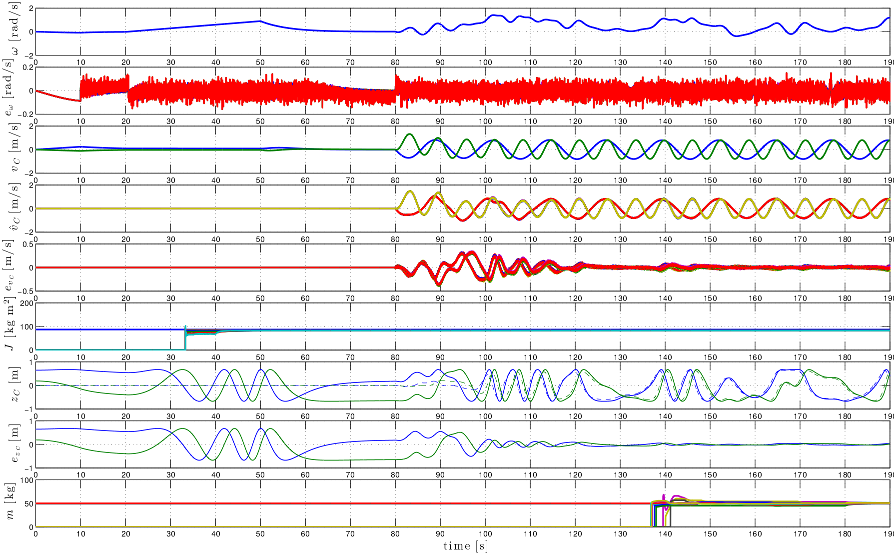
Estimation of m

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

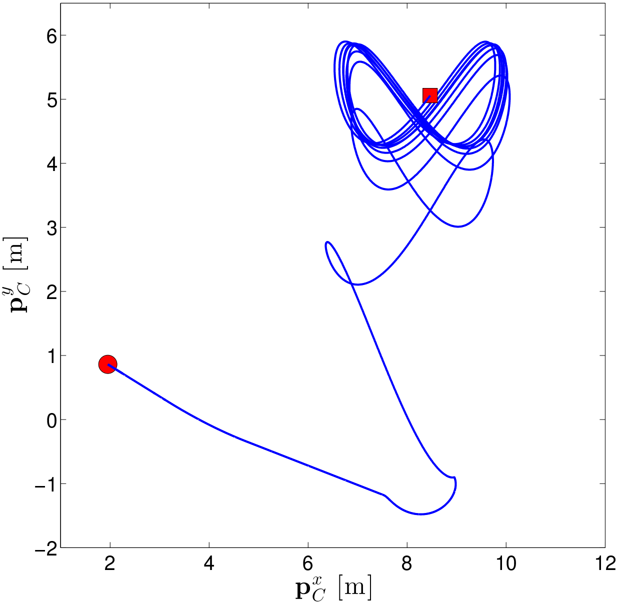
- $\mathbf{v}_C$ locally known
- $\mathbf{f}_{\mathrm{mean}}$ locally known

$\rightarrow$ 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

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 algorthms

[15] 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.

[16] 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.

[17] A. Petitti, A. Franchi, D. Di Paola, *et al.*, "Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators", in *Under Review*.

- **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 retrive 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 retrive 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...

Multi-target Visiting with Connectivity Maintenance

[17] P. Robuffo Giordano, A. Franchi, C. Secchi, *et al.*, "A passivity-based decentralized strategy for generalized connectivity maintenance", *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.
[17] T. Nestmeyer, P. Robuffo Giordano, and A. Franchi, "Simultaneous multi-target exploration and connectivity maintenance", *Under Review*,

[17] D. Zelazo, A. Franchi, H. H. Bülthoff, *et al.*, "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.

Fuure/Open Problems

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**

Questions?

Decentralized Control for Cooperative 2D Mobile Manipulation

A paradigmatic Example of Decentralization in Multi-Robot Control

Antonio Franchi

CNRS, LAAS, France, Europe

JNRR 2015, Baie de Somme

22nd October, 2015

IEEE RAS Technical Committee on Multi-Robot Systems

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`