Safety-guarantees with data-driven controls based on Gaussian processes for quadrotors UAVs and multi-agent systems

Leonardo J. Colombo

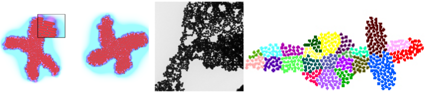
Centre for Automation and Robotics
Spanish National Research Council

Biological and Physical-Mechanical Swarms

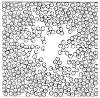
Complex behavior in the biological world

macro biological groups and swarms

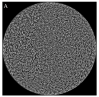
Micro (cellular) sets and formation of patterns in multiple particle swarms



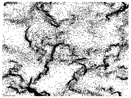
Spontaneous bubble aggregation



Vortex formation in unicycles

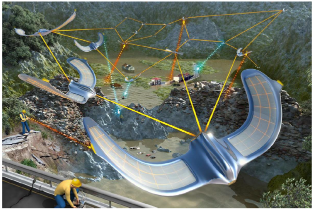


Granular rods



Dissipative gases

Robotic Swarms are coming!

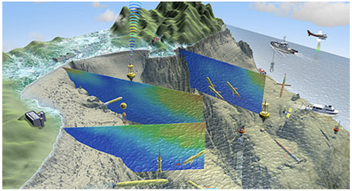


Laboratory of Intelligent Systems, EPFL

Image: NBC News

Persistent Surviellance

over geographically extended areas, situational awareness, suggested route for emergency services.

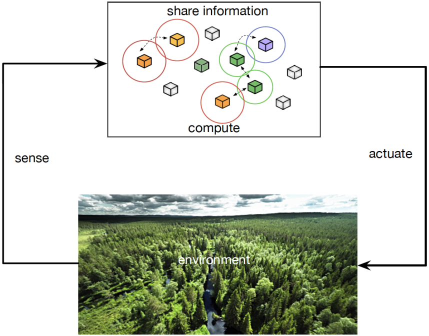


ASAP, Leonard et al

Environmental Monitoring

ocean dynamics sampling, evaluating the impact of pollutants on biological populations, validation of climate models, wildlife monitoring, storm modeling and prediction.

What is a robotic Swarm?



Much more than a collection of robots

- They actively coordinate their actions to achieve the overall task.
- Coherent, better intelligence than the sum of the individuals.
- Sinergy: resilient, spatially distributed, graceful degradation.

How do we control swarms?

Challenges to go beyond *the centralized paradigm of the coordination*.

- Individual capacities are limited.
- The information is local, partial and may be erroneous.
- Distributed interactions (communications) prone to failure.
- Heterogeneous dynamics, multiple scales.

Autonomy engineering

How to coordinate individual elements into an overall coherent?

- From top to bottom: design coordination to design the desired behavior.
- from bottom to top: global behavior of local interaction rules

Limitations for the coordination of robotic swarms

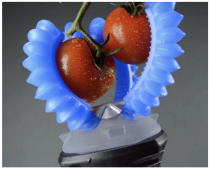
Robotic networks

Limitations

- Computational / memory: precision with respect to the physical system. Global tasks hard to compute/store.
- Physical modeling: Heterogeneous robots, multiple scales, unknown dynamics.
- Comunication: decentralized, asynchronous.
- Sensorial: Noise, partial (e.g., lack of information on the position)

Modeling control systems based on data

Lack of safety guarantees

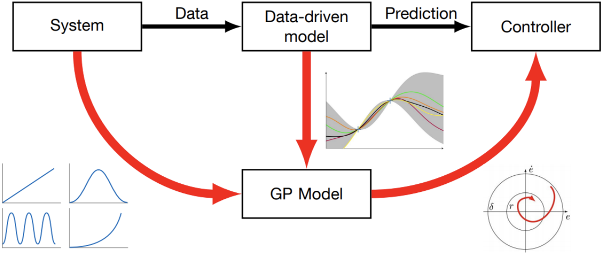

[Soft robotics]

[Kuka]

[Bitcraze]

- The parametric modeling is very time-consuming or even unfeasible.
- Limited to the number of parameters to estimate.
- The modern control methods based on data-driven approaches as neural networks, in general, does not present safety guarantees.
- There is a need to construct algorithms based on data guaranteeing the security of the system.

Data-driven control with GPR: Summary



System

Data

Data-driven model

Prediction

Controller

GP Model

There are no data-driven control laws for complex, multi-agent cooperative systems with safety guarantees.

Research line founded by *Leonardo Fellowship for researhers and cultural creators of the BBA Foundation*. Project: “Safety guarantees with data-driven control for cooperative systems.”

Fundación BBVA

Decentralized and almost-decentralized coordination strategies

- Decentralized formation control strategies

  T. Beckers, S. Hirche, L. Colombo. Safe Online Learning-based Formation Control of Multi-Agent Systems with Gaussian Processes. Proceedings of the 60th. Conference on Decision and Control, 2021.

- Flocking control (almost-decentralized)

  T. Beckers, G. Pappas, L. Colombo. Learning rigidity-based flocking control using Gaussian Processes with probabilistic stability guarantees. Proceedings of the 61st Conference on Decision and Control, 7254-7259, 2022.

Definitions from Graph Theory

Consider $n \geq 2$ autonomous agents whose positions are denoted by $\boldsymbol{p}_i \in \mathbb{R}^d$, $d = \{2, 3\}$ and $\boldsymbol{p} \in \mathbb{R}^{dn}$ denotes the stacked vector of agents' positions. The neighbor relationships between agents are described by an undirected graph $\mathbb{G} = (\mathcal{N}, \mathcal{E})$ with the ordered edge set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$.

The set of neighbors for $i \in \mathcal{N}$, denoted by $\mathcal{N}_i$, is defined by $\mathcal{N}_i := \{j \in \mathcal{N} : (i, j) \in \mathcal{E}\}$.

Agents can sense the relative positions of its nearest neighbors, in particular, agents can measure its Euclidean distance from other agents in the subset $\mathcal{N}_i \subseteq \mathcal{N}$. We define the elements of the incidence matrix $B \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{E}|}$ that establish the neighbors' relationships for $\mathbb{G}$ by

$$B_{i,k} = \begin{cases} +1 & \text{if} \quad i = \mathcal{E}_k^{\text{tail}} \\ -1 & \text{if} \quad i = \mathcal{E}_k^{\text{head}} \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $\mathcal{E}_k^{\text{tail}}$ and $\mathcal{E}_k^{\text{head}}$ denote the tail and head nodes, respectively, of the edge $\mathcal{E}_k$, i.e., $\mathcal{E}_k = (\mathcal{E}_k^{\text{tail}}, \mathcal{E}_k^{\text{head}})$.

Definitions from Graph Theory

The stacked vector of relative positions between neighboring agents, denoted by $\boldsymbol{z} \in \mathbb{R}^{d|\mathcal{N}|}$, is given by

$$\boldsymbol{z} = \overline{B}^{T} \boldsymbol{p},$$

where $\overline{B} := B \otimes I_d \in \mathbb{R}^{d|\mathcal{N}| \times d|\mathcal{E}|}$ with I_d being the $(d \times d)$ identity matrix, and $\otimes$ the Kronecker product.

Note that $\boldsymbol{z}_k \in \mathbb{R}^d$ and $\boldsymbol{z}_{k+|\mathcal{E}|} \in \mathbb{R}^d$ in $\boldsymbol{z}$ correspond to $\boldsymbol{p}_i - \boldsymbol{p}_j$ and $\boldsymbol{p}_j - \boldsymbol{p}_i$ for the edge $\mathcal{E}_k$. We can also define $\boldsymbol{p}_{ij} := \boldsymbol{p}_i - \boldsymbol{p}_j$ to reduce unnecessary verbosity.

In addition, we denote by $D(\boldsymbol{z}) := \operatorname{diag}(\boldsymbol{z}_1, \dots, \boldsymbol{z}_{|\mathcal{E}|}) \in \mathbb{R}^{d|\mathcal{E}| \times |\mathcal{E}|}$

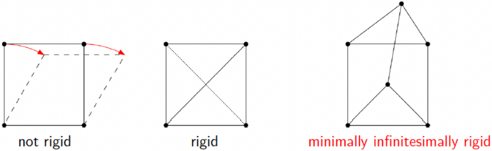
Mehran Mesbahi and Magnus Egerstedt. Graph theoretic methods in multiagent networks. Princeton University Press, 2010.

Formation control: Rigid frameworks

Framework: A *framework* is given by a pair $(\mathbb{G}, \boldsymbol{p})$ where $\mathbb{G} = (\mathcal{N}, \mathcal{E})$ is an undirected graph, with $\boldsymbol{p} \in \mathbb{R}^d$ the vector of positions.

Distance: $f : \mathbb{R}^{d|\mathcal{N}|} \to \mathbb{R}^{|\mathcal{E}|}$, $f(\boldsymbol{p}) = \left(\|p_i - p_j\|^2 \right)_{(i,j) \in \mathcal{E}} = D^\top(\boldsymbol{z})\boldsymbol{z}$.

$(\mathbb{G}, \boldsymbol{p})$ is *rigid* if it is **not** possible to smoothly vary one of the vertexes without moving the other nodes and keep the distance given by f.



L. Asimow and B. Roth. *The rigidity of graphs*. **Transactions of the American Mathematical Society**, Vol. 245, pp. 279-289, 1978.

L. Asimow and B. Roth. *The rigidity of graphs, II*. **Journal of Mathematical Analysis and Applications**. 68(1), 1979.

Rigidity matrix: $R(\boldsymbol{z}) = \frac{1}{2}(\partial f / \partial \boldsymbol{p})(\boldsymbol{p}) = D(\boldsymbol{z})^\top \overline{B} \in \mathbb{R}^{|\mathcal{E}| \times d|\mathcal{N}|}$.

Infinitesimally and minimally rigid frameworks

- error distance for the edge $\mathcal{E}_k$: $\boldsymbol{e}_k = \|p_i - p_j\|^2 - d_k^2 = \|\boldsymbol{z}_k\|^2 - d_k^2$, with d_k the desired distance between neighboring agents over $\mathcal{E}_k$.

- A framework $(\mathbb{G}, \boldsymbol{p})$ is infinitesimally rigid in $\mathbb{R}^d$ if $\boldsymbol{p}$ is a regular value of $f(\boldsymbol{p})$ and $(\mathbb{G}, \boldsymbol{p})$ is rigid in $\mathbb{R}^d$. In particular, $(\mathbb{G}, \boldsymbol{p})$ is *infinitesimally rigid* in $\mathbb{R}^2$ if rank $R(\boldsymbol{z}) = 2|\mathcal{N}| - 3$ (respectivelly, rank $R(\boldsymbol{z}) = 3|\mathcal{N}| - 6$ in $\mathbb{R}^3$).

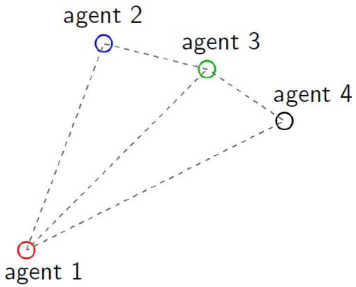
- $(\mathbb{G}, \boldsymbol{p})$ is minimally rigid if it has exactlly $2|\mathcal{N}| - 3$ nodes in $\mathbb{R}^2$ or $3|\mathcal{N}| - 6$ nodes in $\mathbb{R}^3$. This means that if we remove one of the nodes in a minimally rigid framework $(\mathbb{G}, \boldsymbol{p})$, then it is not rigid anymore.

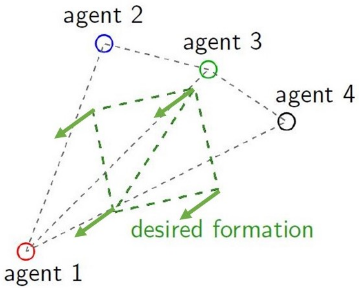
- Therefore, the only allowed movements of the agents in a minimally rigid framework, while they are in the desired formation, are those which are defined by rotations and translations of the desired formation

$$\mathcal{S} = \{\boldsymbol{z} \in \mathbb{R}^{d|\mathcal{N}|} \mid ||\boldsymbol{z}_k|| = d_k\}, \quad k \in \{1, \dots, |\mathcal{E}|\}.$$

- The rigidity matrix $R(z)$ has full-row rank if the framework $(\mathbb{G}, \boldsymbol{p})$ is infinitesimally and minimally rigid.

Formation control problem





Rigidity-based formation control

Consider the following control system in $\mathbb{R}^d$, $d = \{2,3\}$

$$\begin{cases}\dot{\boldsymbol{p}} = \boldsymbol{v} \\ \dot{\boldsymbol{v}} = \boldsymbol{u}.\end{cases}$$

where the control law is given by $\boldsymbol{u}(t) = -\mathcal{K}\boldsymbol{v} - R^\top(\boldsymbol{z})e(\boldsymbol{z})$, $\boldsymbol{z} = \overline{B}^T\boldsymbol{p}$, with $\mathcal{K} = K \otimes I_d$ and K the gain diagonal matrix with the i-th entry being $k_i > 0$. The closed loop system (double integrator formation stabilizer system) is given by

$$\begin{cases}\dot{\boldsymbol{p}} = \boldsymbol{v} \\ \dot{\boldsymbol{v}} = -\mathcal{K}\boldsymbol{v} - R^\top(\boldsymbol{z})e(\boldsymbol{z}).\end{cases}$$

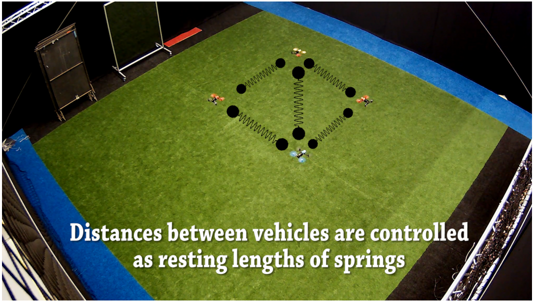
To reach the desired formation, for each node $\mathcal{E}_k$ in an *infinitesimally and minimally rigid framework* consider the artificial potential for each node $\mathcal{E}_k$, $V_k : \mathbb{R}^d \to \mathbb{R}$, given by $V_k(\boldsymbol{z}_k) = \frac{1}{4}(||\boldsymbol{z}_k||^2 - d_k^2)^2$. Denoting by $V_0 = \dfrac{1}{4}\displaystyle\sum_{k=1}^{|\mathcal{E}|}(||\boldsymbol{z}_k||^2 - d_k^2)^2$, with $V_0 : \mathbb{R}^{d|\mathcal{N}|} \to \mathbb{R}$, then

$$\dot{\boldsymbol{p}} = \boldsymbol{v},\ \dot{\boldsymbol{v}} = -\mathcal{K}\boldsymbol{v} - \nabla_{\boldsymbol{p}}V_0.$$

Distance-based formation control as a physical system

The double integrator formation stabilizer system is given by the forced Euler-Lagrange equations for the Lagrangian L subject to the external force F,

$$L(\boldsymbol{p}, \dot{\boldsymbol{p}}) = \frac{1}{2}\sum_{i=1}^{|\mathcal{N}|}\left(||\dot{p}_i||^2 - \frac{1}{4}\sum_{k=1}^{|\mathcal{E}|}(||\boldsymbol{z}_k||^2 - d_k^2)^2\right), \quad F(\boldsymbol{p}, \dot{\boldsymbol{p}}) = -\kappa\sum_{i=1}^{|\mathcal{N}|}\dot{p}_i$$


Distances between vehicles are controlled
as resting lengths of springs

Agent i can be influenced by a non-conservative external force F_i. For instance, F_i can describe a virtual linear damping among two agents (i.e., $F_i = -\kappa \dot{q}_i$).

Stability for rigidity-based formation control

In order to control the velocity of the agents, we consider the potential function $V_1 : \mathbb{R}^{d|\mathcal{N}|} \to \mathbb{R}$ defined as

$$V_1(\boldsymbol{v}) = \frac{1}{2} \sum_{i=1}^{|\mathcal{N}|} ||\boldsymbol{v}_i||^2. \tag{2}$$

By considering $V_0 + V_1$ as energy function of the networked control system with double integrator dynamics, one can show the local asymptotic convergence of the formation to the shape given by

$$\mathcal{S} = \{\boldsymbol{z} \in \mathbb{R}^{d|\mathcal{N}|} |\ ||\boldsymbol{z}_k|| = d_k\}, \quad k \in \{1, \dots, |\mathcal{E}|\}.$$

with velocity zero for all the agents if the framework $(\mathbb{G}, \boldsymbol{p})$ is rigid and local exponential stability for infinitesimally and minimally rigid frameworks.

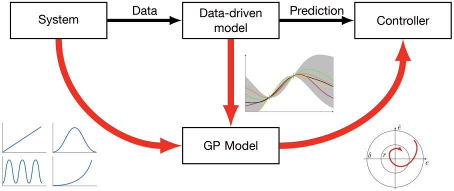
Data-driven control for multiagent systems

Consider the double-integrator formation stabilization system with partially unknown dynamics

$$\begin{cases} \dot{\boldsymbol{p}}_i = \boldsymbol{v}_i, \\ \dot{\boldsymbol{v}}_i = \boldsymbol{u}_i + \boldsymbol{f}_i(\boldsymbol{p}_i, \boldsymbol{v}_i), \end{cases}$$

Problem: Design a decentralized control law $\boldsymbol{u}(t) \in \mathbb{R}^{d|\mathcal{N}|}$ for each agent to achieve the desired rigid formation (infinitesimally and minimally).

Approach: Use GPR to learn the unknown dynamics and to provide an online learning to the agents to reach a desired rigid formation as well as an upper bound for the estimations between the real estimation and the learned (i.e., the prediction of the mean in the GP) and the real dynamics.



System

Data

Data-driven model

Prediction

Controller

GP Model

Online learning

- Consider the estimations $\hat{\boldsymbol{f}}_i : \mathbb{R}^{2d} \to \mathbb{R}^d$ which can include existing prior knowledge about the unknown dynamics $\boldsymbol{f}_i$, e.g., using off-the-shelf modeling or classical system identification.

  K.L. Aström and P. Eykhoff. System identification—a survey. **Automatica** 7(2), 123-162, 1971.

- We employ a GP model for the learning of the unknown dynamics $\boldsymbol{\rho}: \mathbb{R}^{2d|\mathcal{N}|} \to \mathbb{R}^{d|\mathcal{N}|}$, $\boldsymbol{\rho}_i(\boldsymbol{p}_i, \boldsymbol{v}_i) = \boldsymbol{f}_i(\boldsymbol{p}_i, \boldsymbol{v}_i) - \hat{\boldsymbol{f}}_i(\boldsymbol{p}_i, \boldsymbol{v}_i)$.
  - Each agent collects $m(t) \in \mathbb{N}$ training data based on their own dynamics, and giving rise to the training data set $\mathcal{D}_{i,m(t)} = \{\boldsymbol{q}_i^{\{j\}}, \boldsymbol{y}_i^{\{j\}}\}_{j=1}^{m(t)}$, $\boldsymbol{q}_i = [\boldsymbol{p}_i^\top, \boldsymbol{v}_i^\top]^\top$.
  - The output data is given by $\boldsymbol{y}_i = \dot{\boldsymbol{v}}_i - \hat{\boldsymbol{f}}_i(\boldsymbol{p}_i, \boldsymbol{v}_i) - \boldsymbol{u}_i$.
  - The number of training data $m(t)$ in the training data set can change over time t: Permits online learning.
  - Let $\mathcal{D}_{m(t)} = \{\mathcal{D}_{i,m(t)}, \ldots, \mathcal{D}_{|\mathcal{N}|,m(t)}\}$ the set containing all the training data. Assume there are only finitely many switches of $m(t)$ over time and there exists a time $T \in \mathbb{R}_{\geq 0}$ where $\mathcal{D}_{m(T)} = \mathcal{D}_{m(t)}$, $\forall t \geq T, \forall i \in \{1, \ldots, |\mathcal{N}|\}$.

Bounded model error

We assume that the continuous kernel k be chosen in such a way that the functions ρ_i, $i \in \{1, \ldots, d|\mathcal{N}|\}$ have a bounded reproducing kernel Hilbert Space norm on a compact set $\Omega \subset \mathbb{R}^{2d|\mathcal{N}|}$, i.e. $\|\rho_i\|_k < \infty$

The model error is probabilistically bounded by

$$\mathsf{P}\left\{\|\boldsymbol{\mu}(\boldsymbol{\rho} \mid \boldsymbol{q}, \mathcal{D}_m) - \boldsymbol{\rho}(\boldsymbol{q})\| \le \left\|\boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}}(\boldsymbol{\rho} \mid \boldsymbol{q}, \mathcal{D}_m)\right\|\right\} \ge \delta$$

with $\boldsymbol{q} \in \Omega \subset \mathbb{R}^{2d|\mathcal{N}|}$ compact, with $\delta \in (0,1), \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^d$ and denoting by m the number of entries in the data set $\mathcal{D}_m$,

$$\beta_j = \sqrt{2\,\|\rho_j\|_k^2 + 300\gamma_j \ln^3\left(\frac{m+1}{1 - \delta^{1/(d|\mathcal{N}|)}}\right)}$$

The variable $\gamma_j \in \mathbb{R}$ is the maximum information gain

$$\gamma_j = \max_{\boldsymbol{q}^{\{1\}}, \ldots, \boldsymbol{q}^{\{m+1\}} \in \Omega} \frac{1}{2} \log\left|I + \sigma_j^{-2} K(\mathbf{x}, \mathbf{x}')\right|, \ \mathbf{x}, \mathbf{x}' \quad \in \left\{\boldsymbol{q}^{\{1\}}, \ldots, \boldsymbol{q}^{\{m+1\}}\right\}.$$

N. Srinivas, A. Krause, S.M. Kakade, and M.W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. **IEEE Transactions on Information Theory**, 58(5), 2012.

Stability of Hamiltonian systems

Consider the following one-parameter family of systems with double integrator formation stabilization dynamics $\mathcal{H}_\lambda$ given by

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} -\lambda I_{d|\mathcal{N}|} & (1-\lambda) I_{d|\mathcal{N}|} \\ (\lambda - 1) I_{d|\mathcal{N}|} & -\mathcal{K} I_{d|\mathcal{N}|} \end{bmatrix} \begin{bmatrix} \nabla_{\boldsymbol{p}} V \\ \nabla_{\boldsymbol{v}} V \end{bmatrix}, \quad V = \frac{1}{4} \sum_{k=1}^{|\mathcal{E}|} (||z_k||^2 - d_k^2)^2$$

where $\lambda \in [0, 1]$. The system continuously interpolates all convex combinations between a dissipative system for $\lambda = 0$ and a gradient system for $\lambda = 1$.

F. Dörfler and F. Bullo. On the critical coupling for kuramoto oscillators. **SIAM Journal on Applied Dynamical Systems**. 10(3),1070–1099, 2011.

The family $\mathcal{H}_\lambda$ has two important properties summarized in the following

- For all $\lambda \in [0, 1]$, the equilibrium set of $\mathcal{H}_\lambda$ is given by the set of the critical points of the potential function V, and is independent of λ.
- For any equilibrium of $\mathcal{H}_\lambda$ for all $\lambda \in [0, 1]$, the numbers of the stable, neutral, and unstable eigenvalues of the Jacobian of $\mathcal{H}_\lambda$ are the same and independent of λ.

Safe decentralized and online data-driven control law

Theorem

The control law

$$\boldsymbol{u}(t) = -\mathcal{K}\boldsymbol{v} - R^{\top}(\boldsymbol{z})\boldsymbol{e}(\boldsymbol{z}) - \hat{\boldsymbol{f}}(\boldsymbol{p}, \boldsymbol{v}) - \boldsymbol{\mu}(\boldsymbol{\rho}|\boldsymbol{q}, \mathcal{D}_m)$$

guarantees that the error in the convergence to the desired shape for all the agents, is uniformly ultimately bounded in probability by

$$\mathbb{P}\{\|E_{\boldsymbol{e},\boldsymbol{v}}(t)\| \leq \sqrt{2}\max_{q\in\Omega}\bar{\Delta}_{m(T)}(\boldsymbol{q}), \forall t \geq T_\delta\} \geq \delta$$

on Ω with $T_\delta \in \mathbb{R}_{\geq 0}$ and where $E_{\boldsymbol{e},\boldsymbol{v}} := (\boldsymbol{e}, \boldsymbol{v})$ denotes the stacked vector of relative positions errors and velocities for the formation stabilization, where $\bar{\Delta}_m(\boldsymbol{q}) : \Omega \to \mathbb{R}_{\geq 0}$ defines an upper bound of the model error.

- Our result introduces the learning-based control law with guaranteed boundedness of the error for the formation stabilization with certain probability and specify explicitly the upper bound.

- The individual control law $\boldsymbol{u}_i(t)$ of each agent depends on the distance to its neighbors and the data set based on its own dynamics only.

Idea of the proof (part 1)

- Consider the squared distance error for the edge $\mathcal{E}_k$, that is, $\boldsymbol{e}_k = \|\boldsymbol{z}_k\|^2 - d_k^2$ and the stacked vector of squared distance errors denoted by $\boldsymbol{e} = \left[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{|\mathcal{E}|}\right]^\top$. Note that $\dot{\boldsymbol{e}}_k = 2\boldsymbol{z}_k^T \dot{\boldsymbol{z}}_k$.

- Denoting by $E_{\boldsymbol{e},\boldsymbol{v}}^{\lambda}$ the stacked vector of errors $E_{\boldsymbol{e},\boldsymbol{v}}$ for any $\lambda \in [0,1]$, which includes our system for $\lambda = 1$, we know that $E_{\boldsymbol{e},\boldsymbol{v}}^{\lambda}$ and $E_{\boldsymbol{e},\boldsymbol{v}}$ share the same stability properties. So, we will study the system for $\lambda = 0.5$, without loss of generality, that is,

$$\dot{\boldsymbol{p}} = -\frac{1}{2}\overline{B}D(\boldsymbol{z})\boldsymbol{e} + \frac{1}{2}\boldsymbol{v}; \quad \dot{\boldsymbol{z}} = -\frac{1}{2}\overline{B}^T\overline{B}D(\boldsymbol{z})\boldsymbol{e} + \frac{1}{2}\overline{B}^T\boldsymbol{v}$$

$$\dot{\boldsymbol{e}} = -D(\boldsymbol{z})^T\overline{B}^T\overline{B}D(\boldsymbol{z})\boldsymbol{e} + D(\boldsymbol{z})^T\overline{B}^T\boldsymbol{v}; \quad \dot{\boldsymbol{v}} = -\frac{1}{2}\overline{B}D(\boldsymbol{z})\boldsymbol{e} - \mathcal{K}\boldsymbol{v}.$$

- Consider the Lyapunov candidate function with control law $\boldsymbol{u}(t)$ given before,

$$V(\boldsymbol{e},\boldsymbol{v}) = \frac{1}{2}\|\boldsymbol{e}\|^2 + \|\boldsymbol{v}\|^2.$$

- Note that V is positive definite and radially unbounded.
- The time derivative of V along the closed-loop system is given by

$$\dot{V} = -\boldsymbol{e}^\top R(\boldsymbol{z})R(\boldsymbol{z})^\top\boldsymbol{e} - \boldsymbol{v}^\top\mathcal{K}\boldsymbol{v} + \boldsymbol{v}^\top(\boldsymbol{\rho}(\boldsymbol{q}) - \boldsymbol{\mu}(\boldsymbol{\rho}|\boldsymbol{q},\mathcal{D}_m)).$$

Idea of the proof (part 2)

- Denote by λ_{min} and κ_{min} the minimum eigenvalues of $R(\boldsymbol{z})R^\top(\boldsymbol{z})$ and $\mathcal{K}$, respectively.
- Since $\mathcal{S}$ is infinitesimally and minimally rigid then the rigidity matrix is full rank. Therefore $\lambda_{min} > 0$. Note also that, by definition $\kappa_{min} > 0$.
- By the model error bound, it follows that

$$P\{\dot{V} \leq -\lambda_{\min}\|\boldsymbol{e}\|^2 - \kappa_{\min}\|\boldsymbol{v}\|^2 + \|\boldsymbol{v}\|\bar{\Delta}_m(\boldsymbol{q})\} \geq \delta,$$

- here $\bar{\Delta}_m(\boldsymbol{q}) : \Omega \to \mathbb{R}_{\geq 0}$ is a bounded function such that $\|\boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}}(\boldsymbol{\rho} \mid \boldsymbol{q}, \mathcal{D}_m)\| \leq \bar{\Delta}_m(\boldsymbol{q})$, which exists because the kernel function is continuous and therefore it is bounded on a compact set $\Omega \subset \mathbb{R}^{2d|\mathcal{N}|}$, and then the variance $\Sigma(\boldsymbol{\rho} \mid \boldsymbol{q}, \mathcal{D}_m)$ is bounded.
- Then, $\dot{V}$ is negative with probability δ for all $E_{\boldsymbol{e},\boldsymbol{v}}$ with $\|E_{\boldsymbol{e},\boldsymbol{v}}\| > \max\limits_{\boldsymbol{q}\in\Omega} \sqrt{2}\bar{\Delta}_m(\boldsymbol{q})$, where the maximum exists since $\bar{\Delta}_m(\boldsymbol{q})$ is bounded in Ω.
- Finally, using Assumption 1 we define $T \in \mathbb{R}_{\geq 0}$ such that $\mathcal{D}_{m(T)} = \mathcal{D}_{m(t)}$ for all $t \geq T$. Then, V is uniformly ultimately bounded in probability by $P\{\|E_{v,e}\| \leq b,\ \forall t \geq T_\delta \in \mathbb{R}_{\geq 0}\} \geq \delta$ with bound $b = \max\limits_{\boldsymbol{q}\in\Omega} \sqrt{2}\bar{\Delta}_{m(T)}(\boldsymbol{q})$.

Numerical Example

We consider $n = 4$ agents in a 2-dimensional space such that the position of each agent $i \in \mathcal{N}$ is denoted by $\boldsymbol{p}_i = [x_i, y_i]^\top$. The neighbor's relations and desired shape are depicted in the Figure. The graph defines a framework which is infinitesimally and minimally rigid.

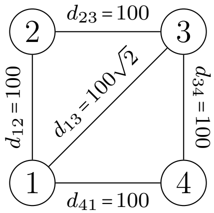


Figure: Neighbor's relations and desired shape

The double integrator dynamics of agent 1 and agent 3 are affected by an arbitrarily chosen unknown dynamics

$$\begin{aligned} \boldsymbol{f}_1(\boldsymbol{p}_1, \boldsymbol{v}_1) &= [200\sin(0.05p_{1,y}), 200\cos(0.05p_{1,x})]^\top \\ \boldsymbol{f}_3(\boldsymbol{p}_3, \boldsymbol{v}_3) &= [50\exp(-0.1(p_{3,y} - 600)^2), 100]^\top. \end{aligned} \tag{3}$$

Numerical Example

The gain matrix $\mathcal{K}$ of the proposed control law is set to $\mathcal{K} = 2I_{2d|\mathcal{N}|}$.

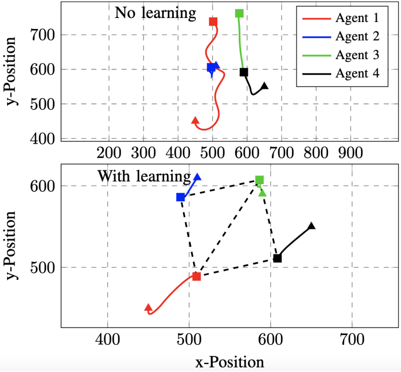
The GP models to predict $\boldsymbol{f}_1, \boldsymbol{f}_3$ are equipped with a squared exponential kernel. No prior model knowledge is assumed, i.e., $\hat{\boldsymbol{f}}_1 = 0, \hat{\boldsymbol{f}}_3 = 0$. At starting time $t = 0$, the data set $\mathcal{D}_m$ is empty.

A training point is added to $\mathcal{D}_m$ every 0.2 seconds and the GP models are updated every 0.4 seconds until a simulation time of 2.6 seconds.

During each update of the GP model, the hyperparameters are optimized by means of the likelihood function. We arbitrarily choose the following initial position $\boldsymbol{p}(0) = [450, 450, 510, 610, 590, 590, 650, 550]^\top$.

Numerical Example

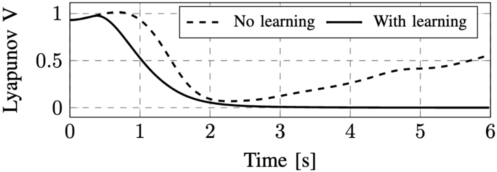
The figure visualizes the trajectories of the agents for i) the standard control law without GP model (top) and ii) the proposed control law with GP model (bottom).



The initial position of each agent is denoted by a triangle whereas the position after the simulation time of $6sec.$ is denoted by a square. The standard control approach fails to reach the desired formation.

Numerical Example

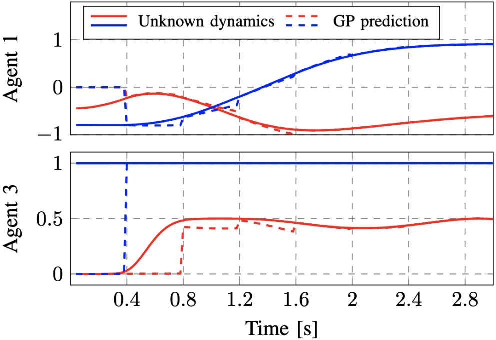
The evolution of the Lyapunov function in the Figure highlights the superior of the proposed control law as it allows the Lyapunov function to converge to a tight set around zero.



Note that the evolution of the Lyapunov function is not always decreasing but bounded in a neighborhood around zero. The size of the set shrinks with improved accuracy of the GP model.

Numerical Example

The online learning process for agent 1 and 3 is depicted inThe figure. The solid line shows the unknown dynamics over time whereas the dashed line is given by the GP prediction.



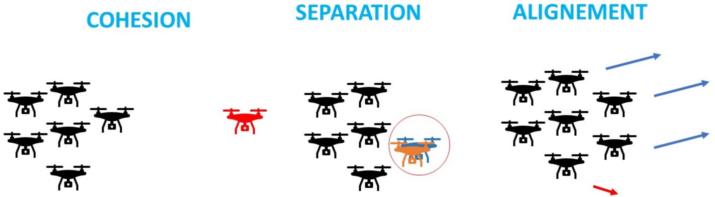
The jumps of the prediction occur due to the model update every 0.4 seconds. After 2 seconds, the GP model can accurately mimic the unknown dynamics.

Flocking control: Problem setting

Flocking control: Problem setting

Goal

Design a distributed control law to converge to a desired formation and to a consensus velocity according to Reynolds's flocking model in the presence of external disturbances.



C. W. Reynolds. Flocks, herds and schools: A distributed behavioral mode. Proceedings of the 14th annual conference on Computer graphics and interactive techniques, 1987, pp. 25-34.

Elements from Graph Theory

Flocking control can be achieved by means of the Laplacian matrix associated with an undirected graph $\mathbb{G} = (\mathcal{N}, \mathcal{V})$.

The Laplacian matrix L is the matrix whose entries are given by $l_{ij} = -1$ for $i \neq j$, if there is an edge between agents j and i, else $l_{ij} = 0$. Moreover, $l_{ii} = -\sum_{j \in \mathcal{N}_i} l_{ij}$. In the case of $\mathbb{G}$ being an undirected graph, it follows that $L = BB^T$.

Modeling flocking control of double integrator agents

Consider the set $\mathcal{N}$ consisting of $n \geq 2$ free autonomous agents evolving on $\mathbb{R}^d$, $d = \{2, 3\}$, under a double integrator dynamics, that is

$$\begin{cases} \dot{\boldsymbol{q}} & = \boldsymbol{v} \\ \dot{\boldsymbol{v}} & = \boldsymbol{u}. \end{cases}$$

According to the Reynolds flocking model, the motion of every agent in the flock is defined by the three rules of alignment, cohesion and separation.

• Cohesion and separation might be achieved by using artificial potential fields.

• We are interested on shape control with flocking motion, that is, agents reach a desired formation shape and they also move along the 2D plane or 3D space by achieving a consensus on their velocities.

• To exponentially achieve this collective behaviour, and provide their convergence rates, we make the following assumption

Assumption: $(\mathbb{G}, \boldsymbol{q})$ is an infinitesimally and minimally rigid framework with $\mathbb{G}$ undirected, static and connected.

Nominal control law

To reach a desired shape, for each edge $\mathcal{E}_k = (i, j)$ in the infinitesimally and minimally rigid framework we introduce the artificial potential functions $V_k : \mathbb{R}^d \to \mathbb{R}$, given by $V_k(\boldsymbol{z}_k) = \frac{1}{4}(||\boldsymbol{z}_k||^2 - d_k^2)^2$, to provide a measure for the interaction between agents and their nearest neighbors.

In these potentials, $\boldsymbol{z}_k$ denotes the relative position between agents for the edge $\mathcal{E}_k$, and d_k denotes the desired length for the edge $\mathcal{E}_k$.

The acceleration of agent i is determined by

$$\boldsymbol{u}_i(t) = -l_{ij} \underbrace{\sum_{j \in \mathcal{N}_i} (\boldsymbol{v}_i - \boldsymbol{v}_j)}_{\text{alignment}} - \underbrace{\sum_{j \in \mathcal{N}_i} \nabla_{\boldsymbol{q}_i} V_k}_{\text{cohesion + separation}},$$

that is, for $\mathcal{L} = L \otimes I_d$, then

$$\boldsymbol{u}(t) = -\mathcal{L}\boldsymbol{v} - R^T(\boldsymbol{z})e(\boldsymbol{z}).$$

Exponential convergence of nominal control

The closed loop system, called double integrator flocking stabilization system is given by

$$\begin{cases} \dot{\boldsymbol{q}}_i = \boldsymbol{v}_i \\ \dot{\boldsymbol{v}}_i = -l_{ij} \sum\limits_{j \in \mathcal{N}_i} (\boldsymbol{v}_i - \boldsymbol{v}_j) - \sum\limits_{j \in \mathcal{N}_i} \nabla_{\boldsymbol{q}_i} V_k. \end{cases} \tag{4}$$

Therefore, we can define the artificial potential function $V_0 : \mathbb{R}^{d|\mathcal{N}|} \to \mathbb{R}$ for the overall networked control system as $V_0(\boldsymbol{z}) = \sum_{k=1}^{|\mathcal{E}|} V_k(||\boldsymbol{z}_k||)$.

In order to control the velocity of the agents, we introduce the disagreement vector $\boldsymbol{\delta} \in \mathbb{R}^{d|\mathcal{N}|}$. Consider the potential function $V_1 : \mathbb{R}^{d|\mathcal{N}|} \to \mathbb{R}$ defined as $V_1(\boldsymbol{\delta}) = \frac{1}{2} \sum_{i=1}^{|\mathcal{N}|} ||\boldsymbol{\delta}_k||^2$, $\boldsymbol{\delta}_k = \boldsymbol{\delta}_i - \boldsymbol{\delta}_j$, where $\boldsymbol{\delta} = [\boldsymbol{\delta}_1^T, \dots, \boldsymbol{\delta}_{|\mathcal{N}|}^T]^T$ is the velocity disagreement vector where each component $\boldsymbol{\delta}_i$ is given by $\boldsymbol{\delta}_i = \boldsymbol{v}_i - \bar{\boldsymbol{v}}$ with $\bar{\boldsymbol{v}}(t) := \dfrac{1}{|\mathcal{N}|} \sum\limits_{i=1}^{|\mathcal{N}|} \boldsymbol{v}_i(t) \in \mathbb{R}^d$ denoting the average velocity of the agents.

Note that $\dot{\bar{\boldsymbol{v}}}(t) = 0$ and hence $\dot{\boldsymbol{\delta}}_i = \dot{\boldsymbol{v}}_i$, since $\bar{\boldsymbol{v}}$ is constant.

Exponential convergence of nominal control

By considering the semi-definite function $V := V_0 + V_1$ as energy function for the double integrator flocking stabilization system, one can show local exponential convergence of the agents to desired shapes with flocking motion behaviour with our graph assumptions.

Deghat, M and Anderson, B.D.O and Lin, Z. Combined flocking and distance-based shape control of multi-agent formations. IEEE Transactions on Automatic Control, 61, 1824–1837, 2016.

Agents with partially unkown dynamics

Next, consider each agent $i \in \{1, \dots, |\mathcal{N}|\}$ disturbed by an additive unknown dynamics given by

$$\begin{cases} \dot{\boldsymbol{q}}_i = \boldsymbol{v}_i, \\ \dot{\boldsymbol{v}}_i = \boldsymbol{u}_i + \boldsymbol{f}_i(\boldsymbol{q}_i, \boldsymbol{v}_i), \end{cases} \tag{5}$$

where $\boldsymbol{f}_i : \mathbb{R}^{2d} \to \mathbb{R}^d$ is a state-dependent unknown function.

Then, (5) can be written as

$$\begin{cases} \dot{\boldsymbol{q}} = \boldsymbol{v}, \\ \dot{\boldsymbol{v}} = \boldsymbol{u} + \boldsymbol{\rho}(\boldsymbol{q}) + \hat{\boldsymbol{f}}(\boldsymbol{q}, \boldsymbol{v}), \end{cases} \tag{6}$$

with the stacked vector of estimating functions $\hat{\boldsymbol{f}}(\boldsymbol{q}, \boldsymbol{v}) = [\hat{\boldsymbol{f}}_1(\boldsymbol{q}_1, \boldsymbol{v}_1)^\top, \dots, \hat{\boldsymbol{f}}_{|\mathcal{N}|}(\boldsymbol{q}_{|\mathcal{N}|}, \boldsymbol{v}_{|\mathcal{N}|})^\top]^\top$ and the unknown dynamics $\boldsymbol{\rho}: \mathbb{R}^{2d|\mathcal{N}|} \to \mathbb{R}^{d|\mathcal{N}|}$ with elements defined by $\boldsymbol{\rho}_i(\boldsymbol{p}_i) = \boldsymbol{f}_i(\boldsymbol{q}_i, \boldsymbol{v}_i) - \hat{\boldsymbol{f}}_i(\boldsymbol{q}_i, \boldsymbol{v}_i)$, where $\boldsymbol{p}_i = [\boldsymbol{q}_i^\top, \boldsymbol{v}_i^\top]^\top$.

In the next step, we employ a GP model for the learning of the unknown dynamics $\boldsymbol{\rho}$.

General assumptions for learning-based control

Each agent collects $m(t) \in \mathbb{N}$ training points based on its own dynamics such that data sets

$$\mathcal{D}_{i,m(t)} = \{\boldsymbol{p}_i^{\{j\}}, \boldsymbol{y}_i^{\{j\}}\}_{j=1}^{m(t)} \tag{7}$$

are created. The output data $\boldsymbol{y}_i \in \mathbb{R}^d$ are given by $\boldsymbol{y}_i = \dot{\boldsymbol{v}}_i - \hat{\boldsymbol{f}}_i(\boldsymbol{q}_i, \boldsymbol{v}_i) - \boldsymbol{u}_i$. The number of training points $m(t)$ of the data sets $\mathcal{D}_{i,m(t)}, i \in \{1, \dots, |\mathcal{N}|\}$ with $m: \mathbb{R}_{\geq 0} \to \mathbb{N}$ can change over time t, i.e., it allows online learning. Let $\mathcal{D}_{m(t)} = \{\mathcal{D}_{i,m(t)}, \dots, \mathcal{D}_{|\mathcal{N}|,m(t)}\}$ be a set that contains all training set. We introduce the following assumption on the data collection.

Assumption: There are only finitely many switches of $m(t)$ over time and there exists a time $T \in \mathbb{R}_{\geq 0}$ where $\mathcal{D}_{m(T)} = \mathcal{D}_{m(t)}, \forall t \geq T, \forall i \in \{1, \dots, |\mathcal{N}|\}$.

Assumption: The continuous kernel k is chosen in such a way the functions ρ_i, $i \in \{1, \dots, d|\mathcal{N}|\}$ have a bounded reproducing kernel Hilbert Space (RKHS) norm on a compact set $\Omega \subset \mathbb{R}^{2d|\mathcal{N}|}$, i.e. $\|\rho_i\|_k < \infty$ for all $i \in \{1, \dots, d|\mathcal{N}|\}$.

Model error

Consider the system (6) and a GP model satisfying the previous assumptions. Then the model error is probabilistically bounded by

$$P\left\{\left\|\boldsymbol{\mu}(\boldsymbol{\rho}\mid\boldsymbol{p},\mathcal{D}_m)-\boldsymbol{\rho}(\boldsymbol{p})\right\|\leq\left\|\boldsymbol{\beta}^\top\Sigma^{\frac{1}{2}}(\boldsymbol{\rho}\mid\boldsymbol{p},\mathcal{D}_m)\right\|\right\}\geq\epsilon$$

for $\boldsymbol{p}\in\Omega\subset\mathbb{R}^{2d|\mathcal{N}|}$ compact, with $\epsilon\in(0,1),\boldsymbol{\beta},\boldsymbol{\gamma}\in\mathbb{R}^d$, and denoting by m the number of entries in the data set $\mathcal{D}_m$,

$$\beta_j=\sqrt{2\left\|\rho_j\right\|_k^2+300\gamma_j\ln^3\left(\frac{m+1}{1-\epsilon^{1/(d|\mathcal{N}|)}}\right)}. \tag{8}$$

The variable $\gamma_j\in\mathbb{R}$ is the maximum information gain

$$\gamma_j=\max_{\boldsymbol{p}^{\{1\}},\dots,\boldsymbol{p}^{\{m+1\}}\in\Omega}\frac{1}{2}\log\left|I+\sigma_j^{-2}K\left(\boldsymbol{x},\boldsymbol{x}'\right)\right| \tag{9}$$

$$\boldsymbol{x},\boldsymbol{x}'\in\left\{\boldsymbol{p}^{\{1\}},\dots,\boldsymbol{p}^{\{m+1\}}\right\}. \tag{10}$$

Online learning for stable flocking control

Consider the potential function $V : \mathbb{R}^{2d|\mathcal{N}|} \to \mathbb{R}$, $V = V_0 + V_1$ as described before.

In the absence of unknown disturbances, V allows to write the closed-loop system formation as

$$\begin{cases} \dot{\boldsymbol{q}} = \nabla_{\boldsymbol{v}} V \\ \dot{\boldsymbol{v}} = -\nabla_{\boldsymbol{v}} V - \nabla_{\boldsymbol{q}} V. \end{cases} \tag{11}$$

Local exponential convergence to the set

$$\mathcal{S} = \{(\boldsymbol{q}^*, \boldsymbol{\delta}^*) \in \mathbb{R}^{2d|\mathcal{N}|} | \nabla_{\boldsymbol{q}} V(\boldsymbol{q}^*) = \mathbf{0}, \boldsymbol{\delta}^* = \mathbf{0}\} \tag{12}$$

for the system (11), has been shown in

Deghat, M and Anderson, B.D.O and Lin, Z. Combined flocking and distance-based shape control of multi-agent formations. IEEE Transactions on Automatic Control, 61, 1824–1837, 2016.

Next, we design a distributed data-driven control law by using GP's, such that, by learning and update the learning of the unknown disturbances, exponentially stabilizes the partially unknown motion of the agents to a desired formation shape with flocking motion.

Online learning for stable flocking control

Consider the one-parameter family of systems with double integrator flocking stabilization dynamics $\mathcal{H}_\lambda$ given by

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} -\lambda I_{d|\mathcal{N}|} & (1-\lambda) I_{d|\mathcal{N}|} \\ (\lambda - 1) I_{d|\mathcal{N}|} & -\mathcal{L} I_{d|\mathcal{N}|} \end{bmatrix} \begin{bmatrix} \nabla_{\boldsymbol{p}} V \\ \nabla_{\boldsymbol{v}} V \end{bmatrix}, \qquad (13)$$

where $\lambda \in [0, 1]$. Equation (13) continuously interpolates all convex combinations between the dissipative system (11) for $\lambda = 0$ and a gradient system for $\lambda = 1$. The family $\mathcal{H}_\lambda$ has two important properties summarized in the following Lemma.

Lemma

(I) *For all $\lambda \in [0, 1]$, the equilibrium set of $\mathcal{H}_\lambda$ is given by the set of the critical points of the potential function V, and is independent of λ.*
(II) *For any equilibrium of $\mathcal{H}_\lambda$ for all $\lambda \in [0, 1]$, the numbers of the stable, neutral, and unstable eigenvalues of the Jacobian of $\mathcal{H}_\lambda$ are the same and independent of λ.*

Online learning for stable flocking control

Denote by $\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}} \coloneqq (\boldsymbol{e}, \boldsymbol{\delta})$ the stacked vector of relative positions errors and velocities disagreement vector for stabilization to desired formation shapes with flocking motion.

The control law

$$\boldsymbol{u}(t) = -\boldsymbol{\mathcal{L}}\boldsymbol{v} - R^{\top}(\boldsymbol{e})\boldsymbol{e}(\boldsymbol{z}) - \hat{\boldsymbol{f}}(\boldsymbol{q}, \boldsymbol{v}) - \boldsymbol{\mu}(\boldsymbol{\rho}|\boldsymbol{p}, \mathcal{D}_m) \tag{14}$$

guarantees that the solution trajectories converge locally exponentially fast to the equilibrium set $\mathcal{S}$ and are ultimately uniformly bounded in probability on Ω with $T_\epsilon \in \mathbb{R}_{\geq 0}$ by

$$\mathbb{P}\{\|\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}}(t)\| \leq \sqrt{2} \max_{p \in \Omega} \bar{\Delta}_{m(T)}(\boldsymbol{p}), \forall t \geq T_\epsilon\} \geq \epsilon, \tag{15}$$

where $\bar{\Delta}_m(\boldsymbol{p}) : \Omega \to \mathbb{R}_{\geq 0}$ defines an upper bound of the model error.

T. Beckers, G. Pappas, L. Colombo. Learning rigidity-based flocking control using Gaussian Processes with probabilistic stability guarantees. Proceedings of the 61st Conference on Decision and Control, 7254-7259, 2022.

Simulation results

We consider 4 agents forming a shape as depicted in the Figure

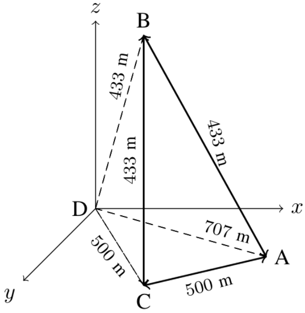


Figure: Neighbor's relations and desired shape

Consider the perturbations over agents 1 and 3 given by
$\boldsymbol{f}_1(\boldsymbol{v}_1) = [300\sin(0.2v_{1,y}), 300\cos(0.2v_{1,x}), 10]^\top$,
$\boldsymbol{f}_3(\boldsymbol{v}_3) = [300\sin(0.2v_{3,y}), -200, 300\sin(0.2v_{3,y})]^\top$. Initial positions are
$\boldsymbol{q}_1(0) = (100, 0, 0)$, $\boldsymbol{q}_2(0) = (0, 0, 200)$, $\boldsymbol{q}_3(0) = (0, -300, 0)$,
$\boldsymbol{q}_4(0) = (100, 0, -300)$.

Simulation results

Note the significant deterioration of the motion when the classical control law has to deal with the unknown forces, as observed in the Figure

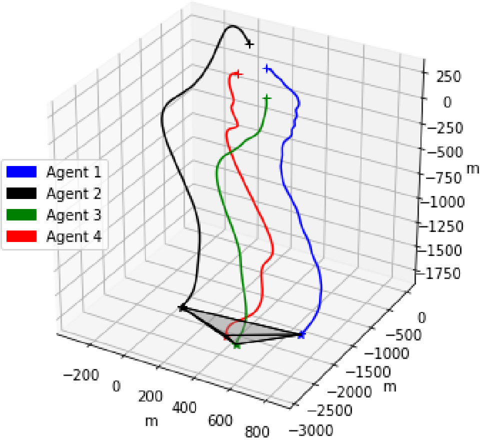


Figure: Nominal control with unknown perturbations

Simulation results

The learning control law has been implemented using a square exponential kernel. At $t = 0$, the training set $\mathcal{D}_{m(0)}$ is empty and a new data point is added every 0.02 seconds, until the final amount of 500 data points is reached at $t = 10$ seconds.

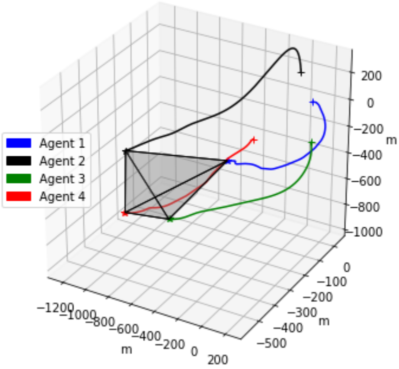


Figure: The proposed learning-based control law can compensate the unknown perturbations.

Simulation results

The evolution of the Lyapunov function highlights the superiority of the proposed control law as it allows the Lyapunov function to converge to a tight set around zero. Note that the evolution of the Lyapunov function is not always decreasing but bounded in a neighborhood around zero. The size of the set shrinks with improved accuracy of the GP model.

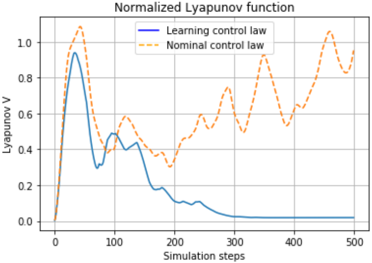


Figure: Normalized Lyapunov function with the standard control law (dashed) and the proposed, learning-based control law (solid) which converges to a tight set around zero.

Control Systems on Lie groups

Control Systems on Lie groups

A Lie group is a differentiable manifold G with group stucture, where the multiplication $(g,h) \mapsto gh$ for $g,h \in G$ and the inversion, $g \mapsto g^{-1}$, are differentiable maps. Denote by e the identity element of G and by $\mathfrak{g} := T_eG$ its Lie algebra.

We are interested on systems on Lie groups since they model a big quantity of mechanical systems. That is, these systems have as configuration space a Lie group.

An important opeation between elements of the Lie group G are left and right translations of elements on G. The left translation of an element $g \in G$ is given by $L_g(h) = gh$ for $h \in G$. The right translation of an element of $g \in G$ is given by $R_g(h) = hg$ for $h \in G$.

Agents with nonlinear dynamics

The special euclidean group in the space

Let $SE(3)$ be the special euclidean group in the space. Any $g \in SE(3)$ is represented as $g = \begin{pmatrix} R & q \\ 0 & 1 \end{pmatrix}$ where R is the orientation in $SO(3)$, the special orthogonal group in the space, and q is the position in $\mathbb{R}^3$.

The group operation is matrix multiplication. Therefore, in this representation $SE(3)$ is a subgroup of the general linear group $GL(3, \mathbb{R})$. The Lie algebra of $SE(3)$ is denoted by $\mathfrak{se}(3)$ and any $\xi \in \mathfrak{se}(3)$ is represented as

$$\xi = \begin{pmatrix} \hat{\Omega} & v \\ 0 & 0 \end{pmatrix},$$

$$\text{where } \Omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}, \hat{\Omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

With this notation, an element of $\mathfrak{se}(3)$ will be sometimes denoted as $\xi = (\hat{\Omega}, v)$ for $\hat{\Omega} \in \mathfrak{so}(3)$, and $v \in \mathbb{R}^3$, where $\mathfrak{so}(3)$ denotes the set of 3×3 skew-symmetric matrices.

Agents with nonlinear dynamics

Basics on Lie group actions

Let G be a finite dimensional Lie group with identity element $e \in G$. A left-action of G on a manifold Q is a smooth mapping $\Phi : G \times Q \to Q$ such that $\Phi(e,q) = q$ for all $q \in Q$, $\Phi(g, \Phi(h,q)) = \Phi(gh,q)$ for all $g, h \in G, q \in Q$ and for every $g \in G$, $\Phi_g : Q \to Q$ defined by $\Phi_g(q) \coloneqq \Phi(g,q)$ is a diffeomorphism.

Let $\mathfrak{g}$ be the Lie algebra associated to G, that is, $\mathfrak{g} = T_e G$. Let $\Phi : G \times G \to G$ be the left group action in the first argument defined as $\Phi(g,h) = L_g(h) = gh$ for all g, $h \in G$. The infinitesimal generator corresponding to $\xi \in \mathfrak{g}$ is $\xi_Q \in \mathfrak{X}(TQ)$ which is defined as $\xi_Q(q) = \frac{\mathrm{d}}{\mathrm{d}t}|_{t=0} \Phi(\exp(t\xi), q)$, where $\exp$ denotes the exponential map. The Lie bracket on $\mathfrak{g}$ is denoted by $[\cdot,\cdot]$. The adjoint map, $ad_\xi : \mathfrak{g} \to \mathfrak{g}$ for $\xi \in \mathfrak{g}$ is defined as $ad_\xi \eta \coloneqq [\xi, \eta]$ for $\eta \in \mathfrak{g}$.

Let $\mathbb{I} : \mathfrak{g} \to \mathfrak{g}^*$ be an isomorphism from the Lie algebra to its dual. The inverse is denoted by $\mathbb{I}^\sharp : \mathfrak{g}^* \to \mathfrak{g}$. The isomorphism $\mathbb{I}$ induces the inner product $\langle\!\langle \cdot,\cdot \rangle\!\rangle_{\mathbb{I}} : \mathfrak{g} \times \mathfrak{g} \to \mathbb{R}$ given by $\langle\!\langle \xi_1, \xi_2 \rangle\!\rangle_{\mathbb{I}} = \langle \mathbb{I}(\xi_1), \xi_2 \rangle_{\mathfrak{g}}$, for $\xi_1, \xi_2 \in \mathfrak{g}$ and where $\langle \cdot,\cdot \rangle_{\mathfrak{g}} : \mathfrak{g}^* \times \mathfrak{g} \to \mathbb{R}$ denotes the natural pairing between elements of $\mathfrak{g}^*$ and $\mathfrak{g}$.

Agents with nonlinear dynamics

Euler-Poincaré equations

The adjoint map $ad_{(\hat{\eta},v)}: \mathfrak{se}(3) \to \mathfrak{se}(3)$ is defined as

$$ad_{(\hat{\eta},v)}(\hat{\xi},\bar{v}) = (\eta \times \xi, \eta \times \bar{v} - \xi \times v),$$

for $(\hat{\eta},v),(\hat{\xi},\bar{v}) \in \mathfrak{se}(3)$. The dual of the adjoint map is $ad^*_{(\hat{\eta},v)}: \mathfrak{se}(3)^* \to \mathfrak{se}(3)^*$ defined as $\langle ad^*_{(\hat{\eta},v)}\alpha, \beta\rangle = \langle \alpha, ad_{(\hat{\eta},v)}\beta\rangle$ for $\alpha, \beta \in \mathfrak{se}(3)^*$, that is

$$ad^*_{(\xi,\alpha)}(\mu,\beta) = (\mu \times \xi - \alpha \times \beta, -\xi \times \beta),$$

for $(\xi,\alpha) \in \mathfrak{se}(3)$ and $(\mu,\beta) \in \mathfrak{se}(3)^*$.

Next, consider a multi-agent system with s agents. Each agent is modeled as a fully actuated mechanical system on $SE(3)$. This means we have 6 actuators on each agent denoted by u_i, with $i = 1,\dots,s$. The dynamics of agent i is given by the Euler-Poincaré equations

$$\begin{aligned} \xi_i &= T_{g_i}L_{g_i^{-1}}\dot{g}_i, \\ \dot{\xi}_i &= u_i + \mathbb{I}^{\sharp} ad^*_{\xi_i}(\mathbb{I}\xi_i). \end{aligned} \tag{16}$$

for $i = 1,\dots,s$ and $\mathbb{I}: \mathfrak{se}(3) \to \mathfrak{se}(3)^*$ is an isomorphism from the Lie algebra $\mathfrak{se}(3)$ to its dual $\mathfrak{se}(3)^*$ with inverse map denoted by $\mathbb{I}^{\sharp}$.

Underactuated vehicles

Underactuated vehicles

We assume a single underactuated rigid body with position $\boldsymbol{p} \in \mathbb{R}^3$ and orientation matrix $R \in SO(3)$. The body-fixed angular velocity is denoted by $\boldsymbol{\omega} \in \mathbb{R}^3$. The vehicle has mass $m \in \mathbb{R}_{>0}$ and rotational inertia tensor $J \in \mathbb{R}^{3\times 3}$. The state space of the vehicle is $S = SE(3) \times \mathbb{R}^6$ with $\boldsymbol{s} = ((R, \boldsymbol{p}), (\boldsymbol{\omega}, \dot{\boldsymbol{p}})) \in S$ denoting the state of the system.

Underactuated vehicles

The vehicle is actuated with torques $\boldsymbol{\tau} \in \mathbb{R}^3$ and a force $u \in \mathbb{R}$, which is applied in a body-fixed direction defined by a unit vector $\boldsymbol{e} \in \mathbb{R}^3$. We can model the system as

$$\begin{aligned} m\ddot{\boldsymbol{p}} &= R\boldsymbol{e}u + \boldsymbol{f}(\boldsymbol{p}, \dot{\boldsymbol{p}}), \quad \dot{R} = R\check{\boldsymbol{\omega}} \\ \dot{\boldsymbol{\omega}} &= J^{-1}\Big(J\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau} + \boldsymbol{f}_\omega(\boldsymbol{s})\Big), \end{aligned} \tag{17}$$

where the map $\check{(\cdot)}: \mathbb{R}^3 \to \mathfrak{so}(3)$ is given by

$$\check{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & -\omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{18}$$

with the components of the angular velocity $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^\top$. The functions $\boldsymbol{f}: \mathbb{R}^6 \to \mathbb{R}^3$ and $\boldsymbol{f}_\omega: S \to \mathbb{R}^3$ are state-depended disturbances and/or unmodeled dynamics. It is assumed that the full state $\boldsymbol{s}$ can be measured.

Underactuated vehicles

The general objective is to track a trajectory specified by the functions $(R_d, \boldsymbol{p}_d)\colon [0, T] \to SE(3)$. For simplicity, we focus here on position tracking only. The extension to rotation tracking is straightforward and will be discussed in the next lecture.

T. Beckers, L. Colombo, G. Pappas, S. Hirche. Online learning-based trajectory tracking for underactuated vehicles with uncertain dynamics. IEEE Control System Letters, 2021.

T. Beckers, L. Colombo, S. Hirche. Safe trajectory tracking for underactuated vehicles with partially unknown dynamics. Journal of Geometric Mechanics 14 (4), 491-505, 2022.

Equivalent system

In preparation for the learning and control step, we transform the system dynamics (17) in an equivalent form. For the unknown dynamics $\boldsymbol{f}$ and $\boldsymbol{f}_\omega$, we use the estimates $\hat{\boldsymbol{f}}\colon \mathbb{R}^6 \to \mathbb{R}^3$ and $\hat{\boldsymbol{f}}_\omega\colon S \to \mathbb{R}^3$, respectively, of an oracle. The estimation error is moved to $\boldsymbol{\rho}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x})$ and $\boldsymbol{\rho}_\omega(\boldsymbol{s}) = \boldsymbol{f}_\omega(\boldsymbol{s}) - \hat{\boldsymbol{f}}_\omega(\boldsymbol{s})$, where $\boldsymbol{x} = [\boldsymbol{p}^\top, \dot{\boldsymbol{p}}^\top]^\top \in \mathbb{R}^6$, $\boldsymbol{s} \in S$. With the system matrix $A \in \mathbb{R}^{6\times6}$ and input matrix $B \in \mathbb{R}^{6\times3}$ given by

$$A = \begin{bmatrix} 0 & I_3 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} I_3 \end{bmatrix}, \tag{19}$$

and $I_3 \in \mathbb{R}^{3\times3}$ as identity matrix, we can rewrite (17) as

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\big(\boldsymbol{g}(R,u) + \hat{\boldsymbol{f}}(\boldsymbol{x}) + \boldsymbol{\rho}(\boldsymbol{x})\big) \\ \dot{R} &= R\check{\boldsymbol{\omega}} \\ \dot{\boldsymbol{\omega}} &= J^{-1}\big(J\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau} + \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}) + \boldsymbol{\rho}_\omega(\boldsymbol{s})\big), \end{aligned} \tag{20}$$

where $\boldsymbol{g}\colon SO(3) \times \mathbb{R} \to \mathbb{R}^3$ is a virtual control input with $\boldsymbol{g}(R,u) \coloneqq R\boldsymbol{e}u$. As consequence, (20) is equivalent to (17) without loss of generality.

Control Archichecture

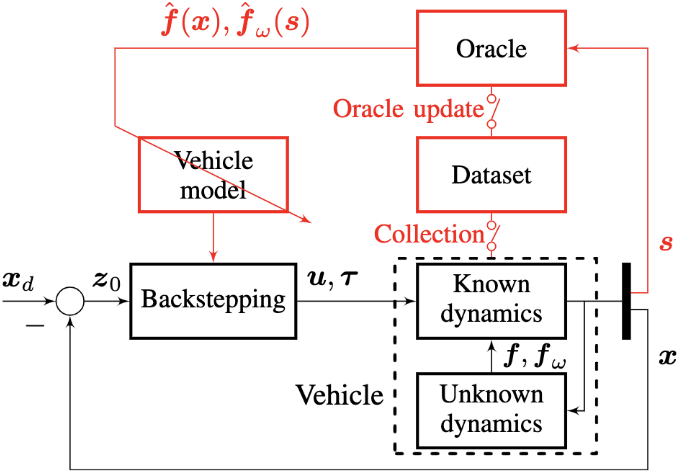


Figure: Control archichecture.

Learning

For the learning of the unknown dynamics of (17), we consider an oracle which predicts the values of $\boldsymbol{f}, \boldsymbol{f}_\omega$ for a given state $\boldsymbol{s}$. For this purpose, the oracle collects $N(n)\colon \mathbb{N} \to \mathbb{N}$ training points of the system (17) to create a data set

$$\mathcal{D}_{n(t)} = \{\boldsymbol{s}^{\{i\}}, \boldsymbol{y}^{\{i\}}\}_{i=1}^{N(n)}. \tag{21}$$

The output data $\boldsymbol{y} \in \mathbb{R}^6$ are given by $\boldsymbol{y} = [(m\ddot{\boldsymbol{p}} - Re u)^\top, (J(\dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times \boldsymbol{\omega}) - \boldsymbol{\tau})^\top]^\top$ such that the first three components of $\boldsymbol{y}$ correspond to $\boldsymbol{f}$ and the remaining to $\boldsymbol{f}_\omega$. The data set $\mathcal{D}_{n(t)}$ with $n\colon \mathbb{R}_{\geq 0} \to \mathbb{N}$ can change over time t, such that at time $t_1 \in \mathbb{R}_{\geq 0}$ the data set $\mathcal{D}_{n(t_1)}$ with $N(n(t_1))$ training points exists.

Learning

Assumption 1: Consider an oracle with the predictions $\hat{\boldsymbol{f}}_n \in \mathcal{C}^2$ and $\hat{\boldsymbol{f}}_{\omega,n} \in \mathcal{C}^0$ based on the data set $\mathcal{D}_n$ for:dataset. Let $S_{\mathcal{X}} \subset (SE(3) \times (\mathcal{X} \subset \mathbb{R}^6))$ be a compact set where the derivatives of $\hat{\boldsymbol{f}}_n$ are bounded on $\mathcal{X}$. There exists a bounded function $\bar{\rho}_n \colon S_{\mathcal{X}} \to \mathbb{R}_{\geq 0}$ such that, if $\|\cdot\|$ denotes the Euclidean norm, the prediction error is bounded by

$$P\left\{ \left\| \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}_n(\boldsymbol{x}) \\ \boldsymbol{f}_\omega(\boldsymbol{s}) - \hat{\boldsymbol{f}}_{\omega,n}(\boldsymbol{s}) \end{bmatrix} \right\| \leq \bar{\rho}_n(\boldsymbol{s}) \right\} \geq \delta \tag{22}$$

with probability $\delta \in (0, 1]$ for all $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{s} \in S_{\mathcal{X}}$.

Assumption 2: The number of data sets $\mathcal{D}_n$ is finite and there are only finitely many switches of $n(t)$ over time, such that there exists a time $T \in \mathbb{R}_{\geq 0}$ where $n(t) = n_{\text{end}}, \forall t \geq T$

Assumption 3: The kernel k is selected such that $\boldsymbol{f}, \boldsymbol{f}_\omega$ have a bounded reproducing kernel Hilbert space (RKHS) norm on $\mathcal{X}$ and $S_{\mathcal{X}}$, respectively, i.e. $\|f_i\|_k < \infty, \|f_{\omega,i}\|_k < \infty$ for all $i = 1, 2, 3$.

Model error

Consider the unknown functions $\boldsymbol{f}, \boldsymbol{f}_\omega$ and a GP model satisfying as:rkhs. The model error is bounded by

$$P\left\{\left\| \boldsymbol{\mu}\left(\begin{bmatrix} \hat{\boldsymbol{f}}_n(\boldsymbol{x}) \\ \hat{\boldsymbol{f}}_{\omega,n}(\boldsymbol{s}) \end{bmatrix} \middle| \boldsymbol{s}, \mathcal{D}_n \right) - \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}) \\ \boldsymbol{f}_\omega(\boldsymbol{s}) \end{bmatrix} \right\| \right.$$

$$\left. \le \left\| \boldsymbol{\beta}_n{}^\top \Sigma^{\frac{1}{2}} \left(\begin{bmatrix} \hat{\boldsymbol{f}}_n(\boldsymbol{x}) \\ \hat{\boldsymbol{f}}_{\omega,n}(\boldsymbol{s}) \end{bmatrix} \middle| \boldsymbol{s}, \mathcal{D}_n \right) \right\| \right\} \ge \delta$$

for $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{s} \in S_{\mathcal{X}}, \delta \in (0,1)$ with $\boldsymbol{\beta}_n \in \mathbb{R}^6$ as before

With Assumption 3 and the fact, that universals kernels exist which generate bounded predictions with bounded derivatives, GP models can be used as oracle to fulfill Assumption 1. In this case, the prediction error bound is given by $\bar{\rho}_n(\boldsymbol{s}) \coloneqq \|\boldsymbol{\beta}_n{}^\top \Sigma^{\frac{1}{2}}([\hat{\boldsymbol{f}}_n(\boldsymbol{x})^\top, \hat{\boldsymbol{f}}_{\omega,n}(\boldsymbol{s})^\top]^\top | \boldsymbol{s}, \mathcal{D}_n)\|$.

Tracking control

For the tracking control, we consider a given desired trajectory $\boldsymbol{x}_d(t)\colon \mathbb{R}_{t\geq 0} \to \mathcal{X}, \boldsymbol{x}_d \in \mathcal{C}^4$. The tracking error is denoted by $\mathbf{z}_0(t) = \boldsymbol{x}(t) - \boldsymbol{x}_d(t)$.

Before we propose the main theorem about the safe learning-based tracking control law, the feedback gain matrix G_n is introduced. As part of the controller, G_n penalizes the position tracking error and the result is fed back to both inputs, the force control u and the torque control $\boldsymbol{\tau}$ of the system.

The feedback gain matrix is allowed to be adapted with any update of the oracle based on a new data set $\mathcal{D}_n$ to lower the feedback gains when the oracle's accuracy is improved.

Property: The matrix $G_n \in \mathbb{R}^{3\times 6}$ is chosen such that there exist a symmetric positive definite matrix $P_n \in \mathbb{R}^{6\times 6}$ and a positive definite matrix $Q_n \in \mathbb{R}^{6\times 6}$ which satisfy the Lyapunov equation

$$P_n\big(A - BG_n\big) + \big(A - BG_n\big)^\top P_n = -Q_n \tag{23}$$

for each switch of $n(t)$.

Backstepping control

Backstepping method is a control theory approach used for controlling and stabilizing nonlinear dynamical systems.

This approach is achieved by recursively stabilizing the system origin in which the control process ends when the final external control is evaluated. Moreover, the backstepping method is a typical strategy for achieving control laws by defining virtually error variables and a Lyapunov function for each subsystem of the original system of ordinary differential equations to assure system stability.

Tracking control

Consider the underactuated rigid-body system given by (17) with unknown dynamics $\boldsymbol{f}, \boldsymbol{f}_\omega$ and the existence of an oracle satisfying Assumptions 2 and 3. Let $G_{z_1}, G_{z_2} \in \mathbb{R}^{3\times 3}$ be positive definite symmetric matrices. The control law

$$\begin{aligned} \boldsymbol{\tau} &= J(\boldsymbol{e}\times(R^\top \boldsymbol{g}_{\ddot{d}} - \breve{\boldsymbol{\omega}}^2 \boldsymbol{e} u - 2\breve{\boldsymbol{\omega}}\boldsymbol{e}\dot{u})u^{-1}) - J\boldsymbol{\omega}\times\boldsymbol{\omega} - \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}), \\ \ddot{u} &= \boldsymbol{e}^\top(R^\top \boldsymbol{g}_{\ddot{d}} - \breve{\boldsymbol{\omega}}^2 \boldsymbol{e} u - 2\breve{\boldsymbol{\omega}}\boldsymbol{e}\dot{u}), \end{aligned} \tag{24}$$

with the desired virtual control input derivative

$$\begin{aligned} \boldsymbol{g}_{\ddot{d}} = {} & m\boldsymbol{p}_d^{(4)} - G_n\left(\frac{\partial \dot{\hat{\boldsymbol{x}}}}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}} - \ddot{\boldsymbol{x}}_d\right) - BP_n(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) - G_{z_2}B^\top P_n \boldsymbol{z}_0 - \frac{\partial}{\partial \boldsymbol{x}}\Big[\frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}\Big]\dot{\hat{\boldsymbol{x}}} \\ & - (G_{z_1} + G_{z_2})\left(\dot{\boldsymbol{g}} - m\boldsymbol{p}_d^{(3)} + G_n(\dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d) + \frac{\partial \hat{\boldsymbol{f}}_n}{\partial \boldsymbol{x}}\dot{\hat{\boldsymbol{x}}}\right) \\ & - (G_{z_2}G_{z_1} + I_3)\left(\boldsymbol{g} - m\ddot{\boldsymbol{p}}_d + G_n\boldsymbol{z}_0 + \hat{\boldsymbol{f}}_n(\boldsymbol{x})\right) \end{aligned} \tag{25}$$

$$\dot{\hat{\boldsymbol{x}}} = A\boldsymbol{x} + B\left(\boldsymbol{g}(R,u) + \hat{\boldsymbol{f}}_n(\boldsymbol{x})\right) \tag{26}$$

guarantees that the tracking error is uniformly ultimately bounded by

$$\mathbb{P}\{\|\boldsymbol{z}_0(t)\| \leq \max_{\boldsymbol{s}\in S_{\mathcal{X}}} \bar{\rho}_{n_{\text{end}}}(\boldsymbol{s})b_{n_{\text{end}}}, \forall t \geq T\} \geq \delta \tag{27}$$

with $b_{n_{\text{end}}} = (\max\{eig(P_{n_{\text{end}}}), 1\}/\min\{eig(P_{n_{\text{end}}}), 1\})^{1/2}$, and time constant $T \in \mathbb{R}_{\geq 0}$ on $S_{\mathcal{X}}$.

Tracking control

- The control law does not depend on any state derivatives, which are typically noisy in measurements. The derivatives, i.e. the translational and angular accelerations, are only necessary for the training of the oracle, which can often deal with noisy data. For instance, GP models can handle additive Gaussian noise on the output.

- We prove the stability of the closed-loop with the proposed control law with multiple Lyapunov function, where the n-th function is active when the oracle predicts based on the corresponding training set $\mathcal{D}_n$. Note that due to a finite number of switching events, the switching between stable systems can not lead to an unbounded trajectory.

- The torque control law of (24) has a singularity at $u = 0$ as without control force u no tracking control is possible in general. To overcome the singularity, a reasonable trajectory planning can be performed, or the control torques are set to zero at this point. In practice, this leads to chattering that can be alleviated by a slight modification of the control law to remove the singularity.

- The proof shows that the bound of the tracking error depends on the prediction error $\bar{\rho}_n$ of the oracle.

Numerical Example 1

To demonstrate the application relevance of our proposed approach, we consider the task of an quadcopter to explore a terrain with unknown thermals.

The dynamics of the vehicle is described with mass $m = 1$ kilogram, inertia $J = I_3 kg/m^2$ and the direction $\boldsymbol{e} = [0, 0, 1]^\top$ of the force input u.

The data of the thermals is taken from publicly available paragliding data.

The thermals are assumed to act on the quadcopter as a disturbance in the direction of x_3, i.e., the altitude, as well as an angular momentum in the direction of ω_1. A GP model is then used as oracle to predict $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}_\omega(\boldsymbol{s})$ based on the collected data set with the squared exponential kernel.

Numerical Example 1

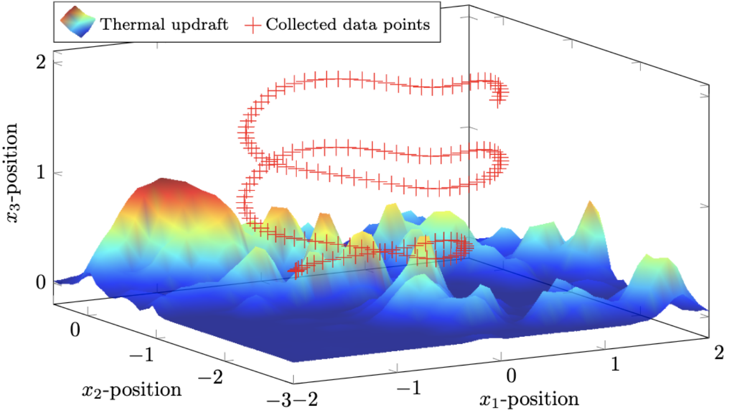


Figure: Visualization of the normalized magnitude of the thermal updraft acting on the quadcopter and the recorded training data points (red crosses).

Numerical Example 1

The prior knowledge about the existing gravity in $\boldsymbol{f}(\boldsymbol{x})$ is considered as estimate in the mean function of the GP with $m_3(\boldsymbol{s}) = -10$.

First, we start with the collection of training data for the GP model. For this purpose, the control inputs for the aerial vehicle are generated by a controller but without an oracle, i.e. $\hat{\boldsymbol{f}}(\boldsymbol{x}) = \hat{\boldsymbol{f}}_\omega(\boldsymbol{s}) = \boldsymbol{0}, \forall \boldsymbol{x} \in \mathbb{R}^6, \boldsymbol{s} \in \mathcal{S}$.

The feedback gain matrix is set to

$$G = \begin{bmatrix} 10 & 0 & 0 & 10 & 0 & 0 \\ 0 & 10 & 0 & 0 & 10 & 0 \\ 0 & 0 & 20 & 0 & 0 & 10 \end{bmatrix} \tag{28}$$

and $G_{z_1} = G_{z_2} = 2I_3$.

The desired trajectory is given by $x_{d,1}(t) = \sin(t), x_{d,2}(t) = \cos(t) - 1$, $x_{d,3}(t) = t/10$.

Every 0.1 second a training point has been recorded. Each training point consists of the actual state $\boldsymbol{s}$ and $\boldsymbol{y}$.

Numerical Example 1

Since the training points depend on the typically noisy measurement of the accelerations $\ddot{\boldsymbol{p}}$ and $\dot{\boldsymbol{\omega}}$, a Gaussian distributed noise $\mathcal{N}(0, 0.08^2 I_3)$ is added to the measurement. After the simulation time of 15 seconds, the data set $\mathcal{D}$ consists of 150 training points. Based on this data set, a GP model is trained and the hyperparameters are optimized by means of the likelihood function.

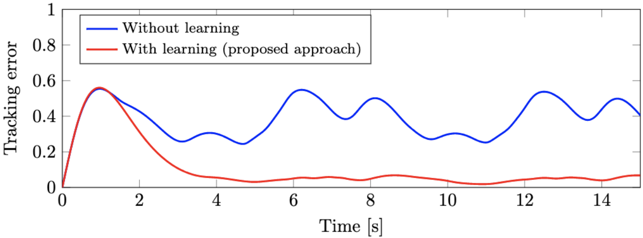


Figure: Tracking error of the quadcopter with control law $\boldsymbol{u}$ without learning (blue) and with our proposed learning-based approach (red).

Numerical Example 1

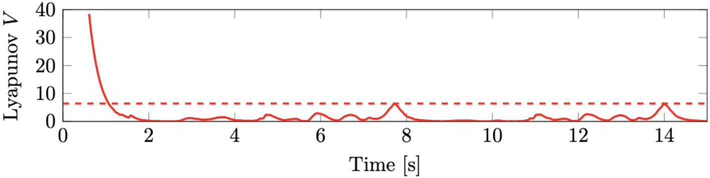


Figure: Lyapunov function (solid) converges to a tight set around zero (dashed line) and stays inside this set with high probability as guaranteed by Theorem.

Numerical Example 1

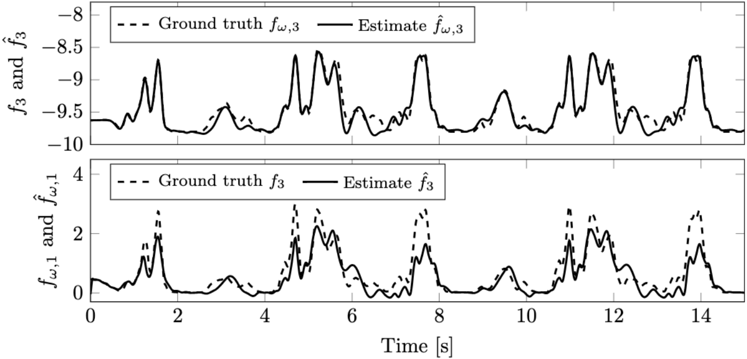


Figure: Ground truth (dashed) of the unknown dynamics and estimates of the GP (solid). Top: Unknown dynamics acting on the x_3-position of the quadcopter. Bottom: Unknown dynamics acting on the first component of the angular acceleration $\dot{\boldsymbol{\omega}}$ of the quadcopter.

Numerical Example 2

The dynamics of the quadrocopter are described by (17) with mass $m = 1$ kilogram, inertia $J = \mathrm{diag}(2,2,1) kg/m^2$ and the direction $\boldsymbol{e} = [0,0,1]^\top$ of the force input u. As unknown dynamics $\boldsymbol{f}$ and $\boldsymbol{f}_\omega$, we consider an arbitrarily chosen wind field and the gravity force given by

$$\boldsymbol{f}(\boldsymbol{x}) = [0, 0, 2\sin(x_1) + \exp(-5x_2^2) - 9.81]^\top \tag{29}$$

$$\boldsymbol{f}_\omega(\boldsymbol{s}) = [2\exp(-x_1^2 - x_2^2) + \omega_1 \cos(x_2)^2, 0, 0]^\top. \tag{30}$$

The posterior mean $\boldsymbol{\mu}(\boldsymbol{y}|\boldsymbol{s}, \mathcal{D}_n)$ of a GP model is used as oracle to predict the z-component of $\boldsymbol{f}(\boldsymbol{x})$ and the x-component of $\boldsymbol{f}_\omega(\boldsymbol{s})$ with the squared exponential kernel.

The prior knowledge about the existing gravity in $\boldsymbol{f}(\boldsymbol{x})$ is considered as estimate in the mean function of the GP with $m_3(\boldsymbol{s}) = -10$. At starting time $t = 0$, the data set $\mathcal{D}_n$ is empty such that the prediction is solely based on the mean function.

Numerical Example 2

The initial position of the quadrocopter is $\boldsymbol{p}(0) = [0.1, -0.1, 0]^\top$ whereas the desired trajectory starts at $\boldsymbol{p}_d(0) = [0, 0, 0]^\top$ due to an assumed position measurement error. We employ an online learning approach which collects a new data point every 0.1 seconds such that the total number of data points is $N = 5n$.

The GP model is updated every second until $t = 12$ seconds, where the last 10 collected training points are appended to the set $\mathcal{D}_n$ and the hyperparameters are optimized by means of the likelihood function.

Numerical Example 2

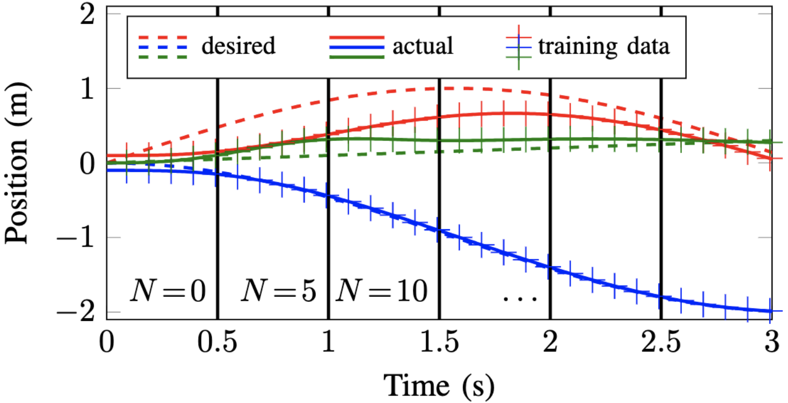


Figure: A segment of the desired (dashed) and actual trajectory (solid). Every 0.1 seconds a training point (cross) is recorded. Every 0.5 seconds the oracle is updated based on all collected training points N. The additional training data allows to refine the model such that the tracking error is decreasing.

Numerical Example 2

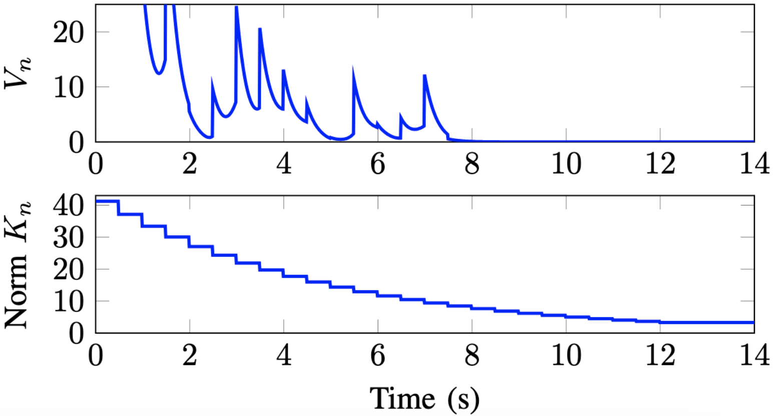


Figure: Top: Lyapunov function converges to a tight set around zero. The jumps occur when the oracle is updated. Bottom: Norm of the feedback gain matrix is decreasing due to improved accuracy of the oracle.

Numerical Example 2

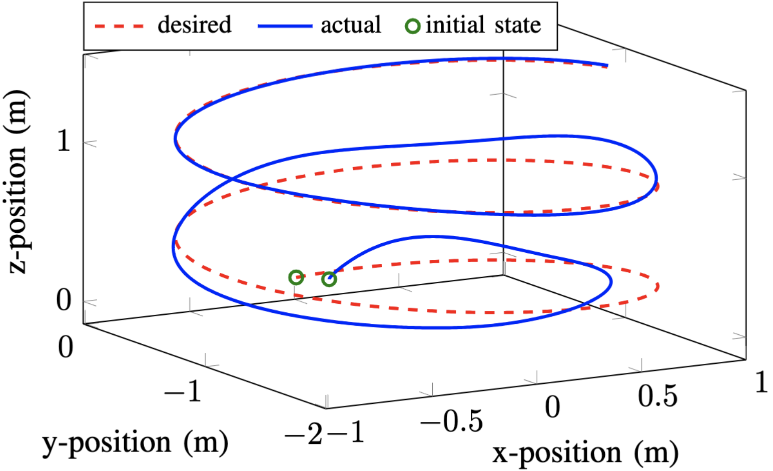


Figure: Actual trajectory converges to desired trajectory