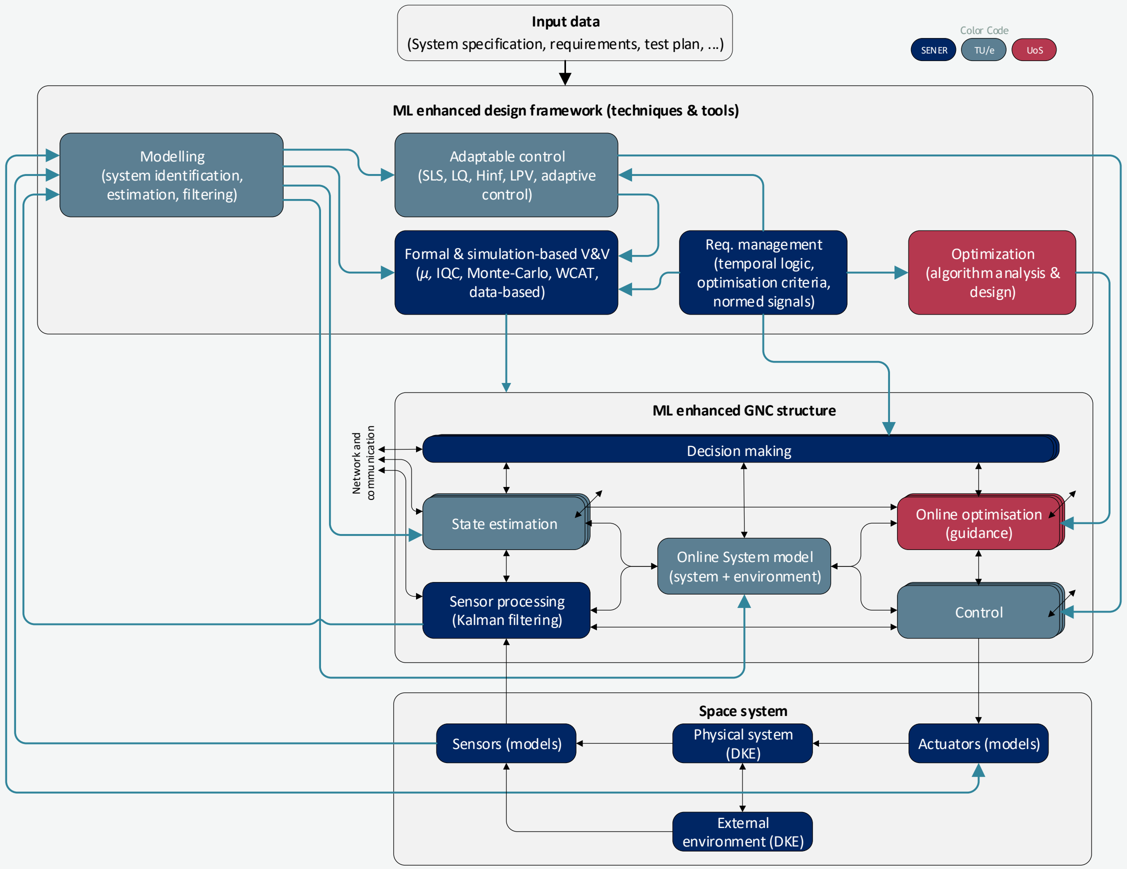
“Artificial Intelligence techniques for GNC design, implementation and verification”

Final Review

2022-12-19



AI4GNC - FR

Agenda

- 10:00 - 10:15 Welcome and status update
- 10:15 - 11:15 Technical Presentations (D7 Update)
  - 10:15 - 10:45 TUe – Model Augmentation
  - 10:45 - 11:15 Aachen/UoS – Robust Guidance
- 11:15 - 11:30 Buffer/Mini Break
- 11:30 - 12:30 SENER – (D8) Study Synthesis Presentation & Discussion
- 12:30 - 13:30 Lunch Break
- 13:30 - 15:00 Project Closure Discussion & Way Forward

TUe

Model Augmentation

Aachen/UoS

Robust Guidance

SENER

Study Synthesis

Study Synthesis

Introduction

1. Critical Analysis of Obtained Results
   - High-level technical summary
   - Recap of obtained results

2. Synthesis of New Capabilities
   - Results in larger context for GNC discipline
   - Trends and developments supported by AI4GNC project

3. Lessons Learned & Discussion
   - Ways to bring techniques into application/adoption
   - Conclusions / Discussion

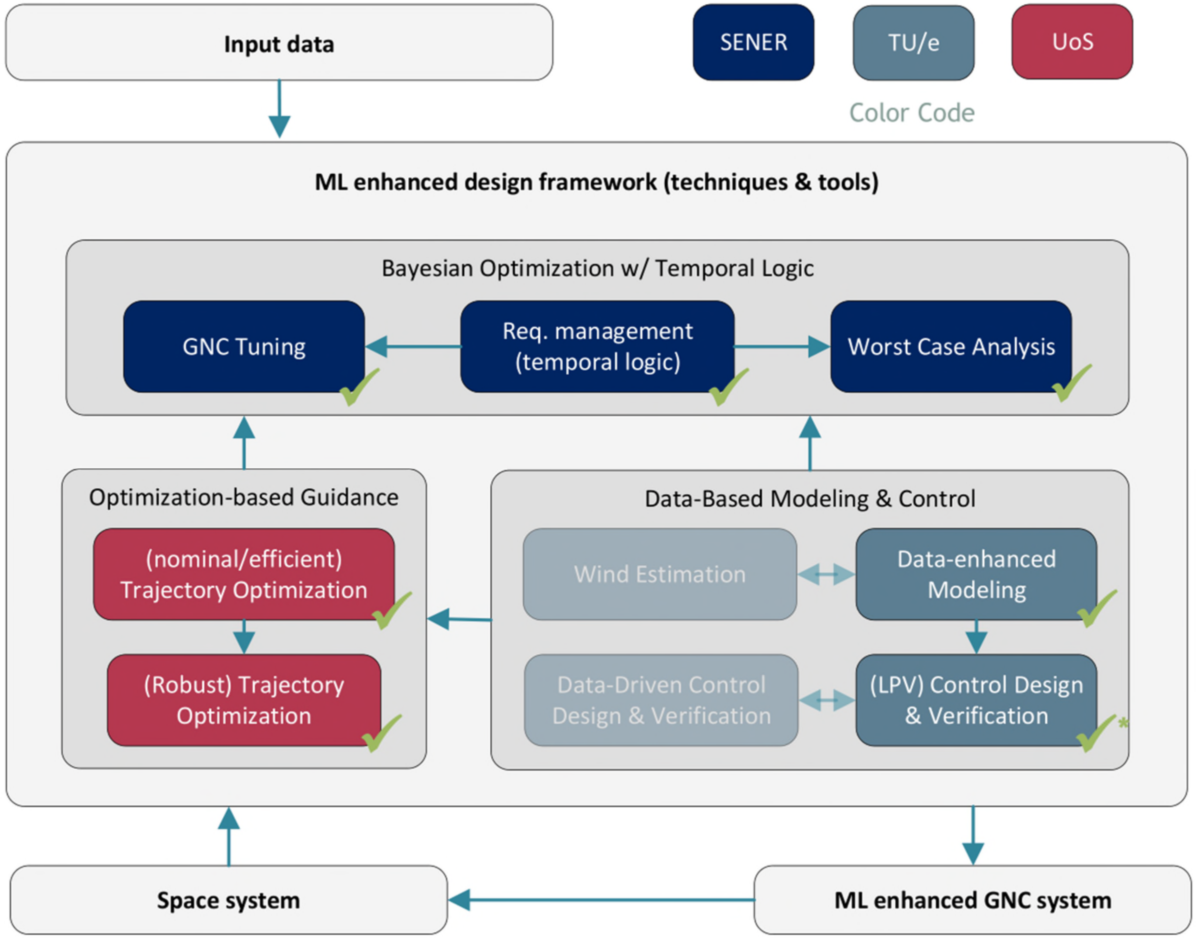
SENER

Critical Analysis of Obtained Results

Critical Analysis of Results

Project Execution

- Project executed as planned
- High-level goals realized
- Some adjustments throughout the project
    - Scope clarified during execution
    - Focus on most promising ideas
- Explorative nature of project at times difficult



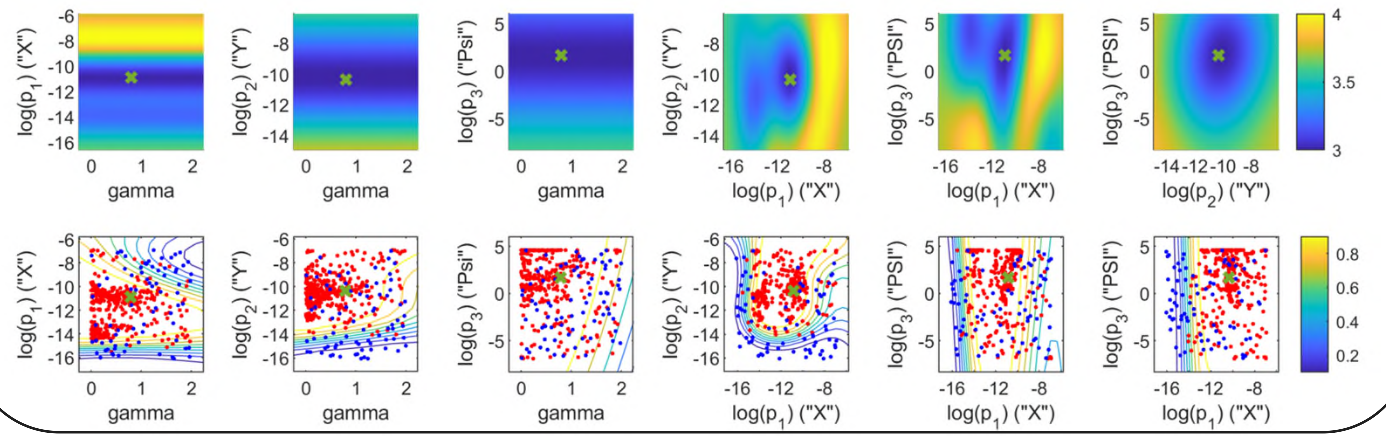
Critical Analysis of Results

Bayesian Optimization for Controller Tuning

Focus points

- Noisy simulations/function evaluations
  → seamless integration with MC
- Temporal logic constraints
  → direct optimization of requirements
- Interpretability & engineering insights
  → design tool with “human-in-the-loop”

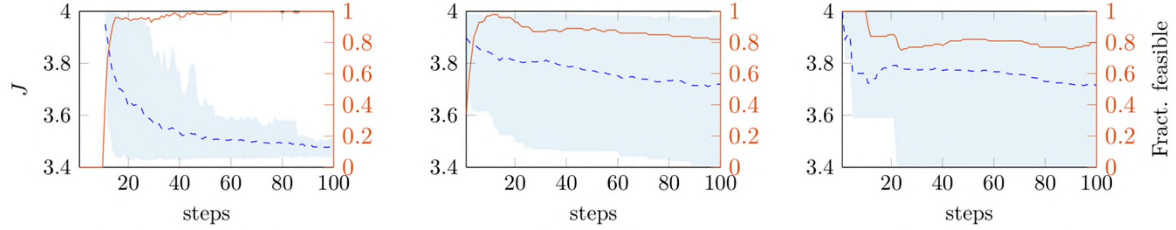
Cost Function Optimization



Results

→ highly competitive performance
→ applicable & effective for “real” tasks
→ Loses reliability in higher dimensions
→ Problem needs to be formulated “well”

Algorithm Performance



(a) Bayesian Optimization (b) Random (c) Surrogate optimization

esa SENER Aeroespacial TU/e Universität Stuttgart

Critical Analysis of Results

Robust Optimization-Based Guidance

- Extends robust control techniques to guidance level
- Can integrate model uncertainties
    - show-cased in ML-based scenarios
    - Generates distinct *robust* behaviour

  → Could generate insights for guidance strategies

Results

→ Significant performance improvements
- Over baseline solution
- Over non-robust variants

→ High design flexibility

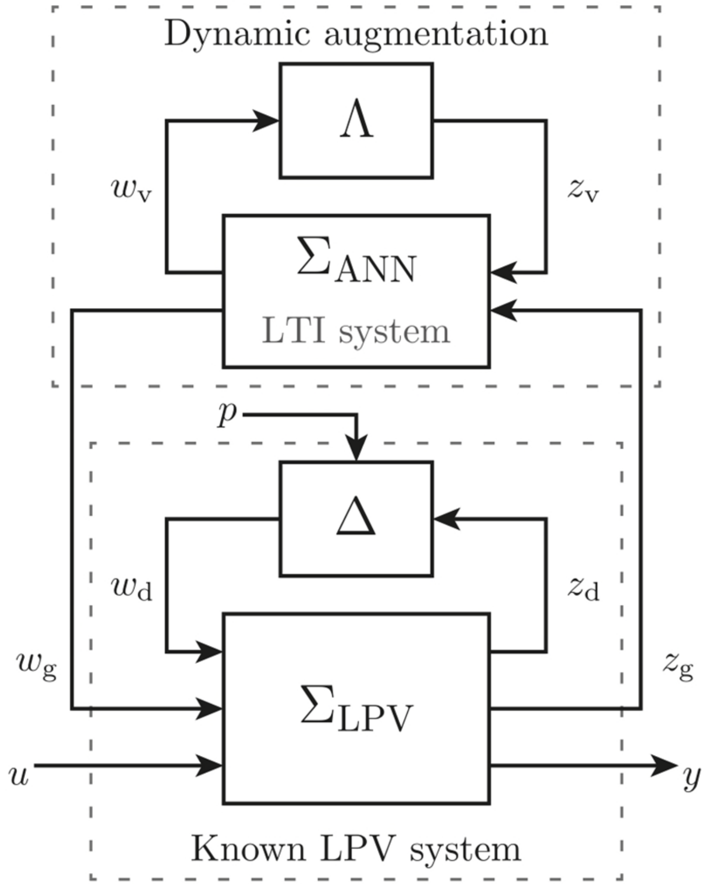
→ Complex (both computationally & design)
- Significant steps taken towards implementability
- Essentially real-time capable on Laptop

algorithm	wind error	90% quantiles	mean accuracy	median accuracy
Baseline	0m/s	27.8517	27.5154	16.1469
Baseline	$\frac{2.5}{3}\text{m/s}$	39.9353	34.9835	17.594
Baseline	$\frac{5}{3}\text{m/s}$	97.5034	59.1223 (39.3975)	21.4595
Baseline	$\frac{7.5}{3}\text{m/s}$	185.8093	108.3531	26.7415
DDP	0m/s	24.7212	20.2993	11.7073
DDP	$\frac{2.5}{3}\text{m/s}$	38.3248	28.8612	14.4037
DDP	$\frac{5}{3}\text{m/s}$	82.8809	52.1305	20.4581
DDP	$\frac{7.5}{3}\text{m/s}$	145.9398	94.1244	27.9901
robust DDP	0m/s	21.4472	19.8503	11.6286
robust DDP	$\frac{2.5}{3}\text{m/s}$	24.3172	21.7991	12.7784
robust DDP	$\frac{5}{3}\text{m/s}$	49.698	38.3896	16.2578
robust DDP	$\frac{7.5}{3}\text{m/s}$	127.2579	80.0665	22.0501

Critical Analysis of Results

Data-Driven Model Augmentation

- Valuable tool for complex (closed loop) dynamics,
  - Also with simulation data
  - LPV (or LTI) also provide good performance
  - Nonlinear/ML methods extend range of validity
- The performance depends on formulation
  - model structure
  - quality of the data
- Discussion point: How important/helpful is augmentation?
  - Black-box learning/ID easier?
  - Interpretability



SENER

Synthesis of New Capabilities

Synthesis of New Capabilities

Model-based Design and Decision Making

- Main models used in current GNC software
  - Linear (control design & analysis)
  - DKE high-fidelity functional simulator (V&V)

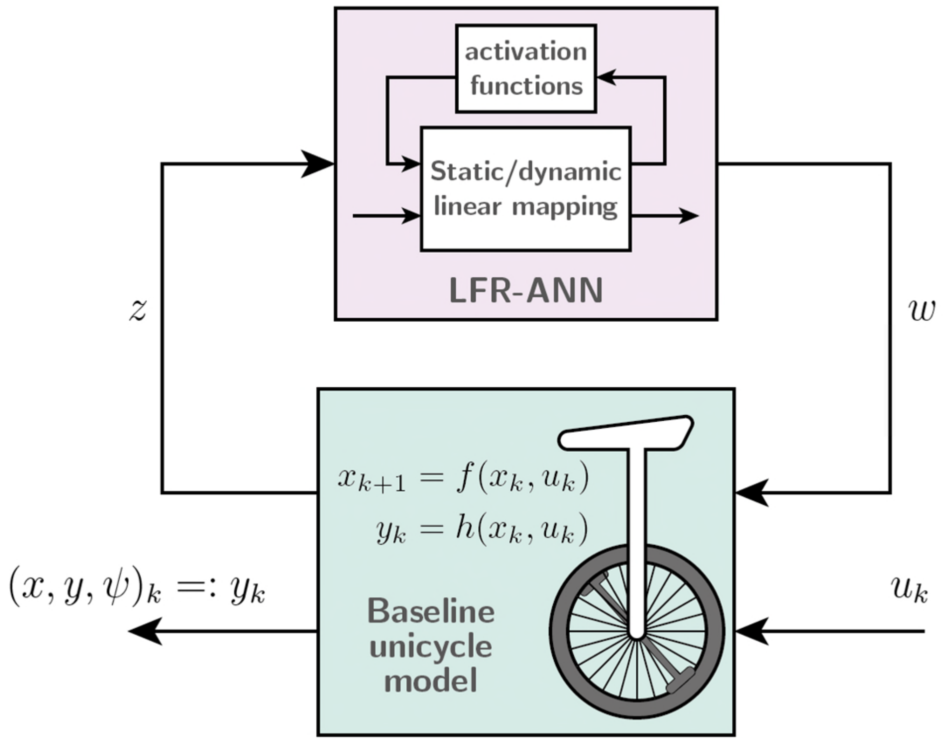
- Trend towards additional models for **decision making**
  - Exemplified by optimization-based guidance, data-driven model learning for guidance
  - Applies to most potential techniques for advanced capabilities & autonomy

- Advantages envisioned & demonstrated in AI4GNC

  Performance

  Design Flexibility

  Increase Autonomy



Synthesis of New Capabilities

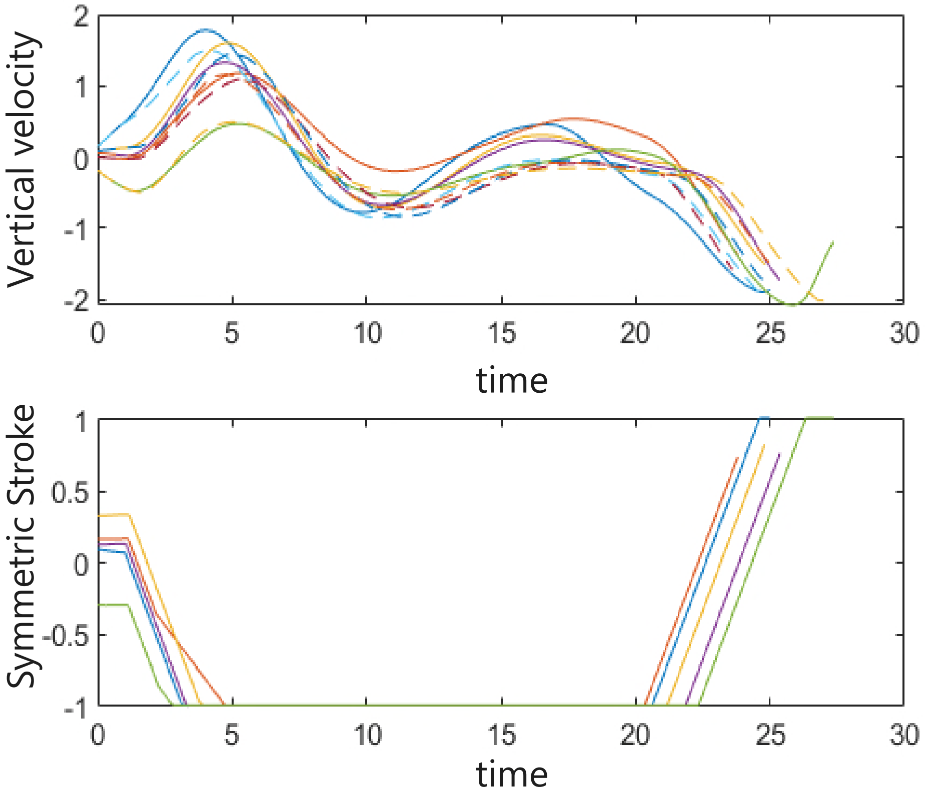
Data-Driven GNC Design

1. **Data-Driven Modelling**
   - Current designs focused on first principle models
   - Data-driven approaches may be underrepresented
     - Possibly improved model fit since complicated effects can be approximately captured
     - significantly reduced engineering effort

   → actively explored in SR project (linear models)

2. **Simulation for GNC design**
   - Currently the focus of simulations lies on V&V
   - Bayesian optimization, RL and related techniques shift focus
     - Significant potential to streamline tuning process
     - Overhead in their application needs to be reduced

*Dashed: Linear tf fit



SENER

Lessons Learned & Discussion

Lessons-Learned and Discussion

Part I: Technical

1. **Potential for advanced techniques**: Opportunities need to be clearly identified
   - Also in this project/benchmark -> what can advanced techniques improve? How?
   - What is the performance/complexity trade-off? Quantifiable?
     (E.g. linear augmentation vs ANN-based, Optimization problem classes)

2. **Use of techniques has significant overhead**
   - Can be reduced by good software integration
   - Techniques often inherently complex and challenging to design
     - Toolboxes and software packages can help to some extent
     - Still likely requires expert personnel

Lessons-Learned

Part II: Project Execution

3. Format of the project right for this type of study?
   (PRR → CRR → PDR → DDR → VR → FR)
   - (out of necessity) we took large liberties with the format
   - May give structure to development efforts

4. Use of SENER benchmark simulator
   - Has caused significant challenges
     - Reduced flexibility/agility, large overhead for development
     - Using different environment, we could have produced more results
   - **But:** Provides significant value by enforcing realistic use cases
     - Avoids demonstration on academic toy problems
   - Difficult tradeoff to be carefully considered

Lessons-Learned

Further discussion points?

Robust trajectory generation

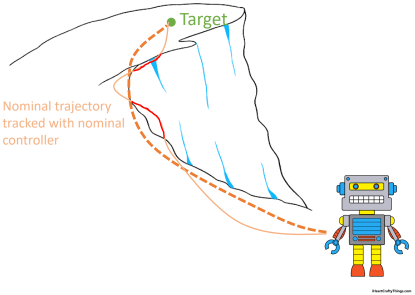
Final Review

Dennis Gramlich, Carsten W. Scherer, Christian Ebenbauer

Chair of Intelligent Control Systems, RWTH Aachen
Chair for Mathematical Systems Theory, University of Stuttgart

December 19, 2022

Why we are interested in robust trajectory generation



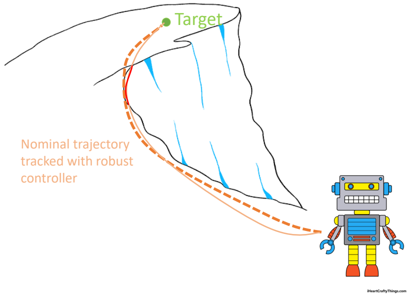
Robust control in aerospace engineering:

- Generate a nominal trajectory.
- Track the trajectory with a robust controller.

We propose:

- The integration of robust control into trajectory generation.

Why we are interested in robust trajectory generation



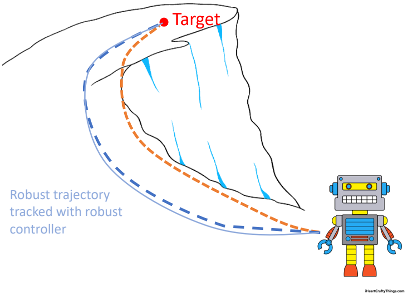
Robust control in aerospace engineering:

- Generate a nominal trajectory.
- Track the trajectory with a robust controller.

We propose:

- The integration of robust control into trajectory generation.

Why we are interested in robust trajectory generation



Robust control in aerospace engineering:

- Generate a nominal trajectory.
- Track the trajectory with a robust controller.

We propose:

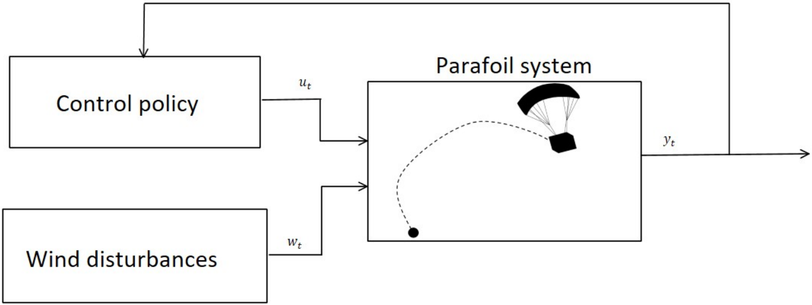
- The integration of robust control into trajectory generation.

Parafoil payload landing scenario

Goal: Find a control policy that minimizes the functional

$$\int_0^{t_f} u^2(t)\,\mathrm{d}t + \left\| \begin{pmatrix} x(t_f) - x_f \\ y(t_f) - y_f \\ \psi(t_f) - \psi_f \end{pmatrix} \right\|_{\boldsymbol{F}}^2$$

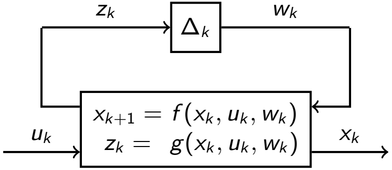
while controlling the system below and compensating wind disturbances.



Outline

Robust Differential Dynamic Programming

Robust Differential Dynamic Programming



The developped robust Differential Dynamic Programming (robust DDP) algorithm

- is applicable to nonlinear generalized plants as above and
- provides guarantees for affine linear time-varying systems.

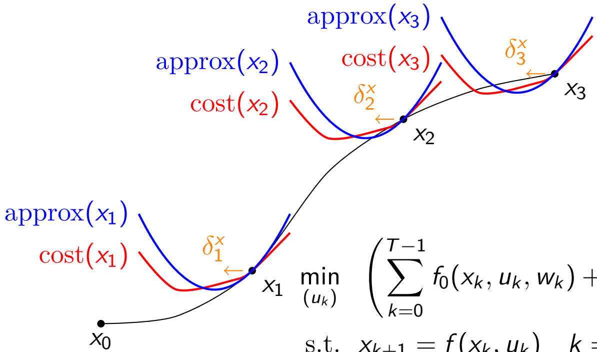
Robust Differential Dynamic Programming

For a nonlinear genearlized plant, we solve the robust Dynamic Program

$$
\begin{aligned}
\underset{(\pi_k)}{\text{minimize}}\ \underset{(\Delta_k)\subseteq\mathcal{D}}{\text{maximize}} \quad & \left(\sum_{k=0}^{T-1} f_0(x_k, u_k, w_k) + V_T(x_T)\right) \\
\text{s.t.} \quad & x_{k+1} = f(x_k, u_k, w_k) && k = 0, \dots, T-1 \\
& x_0 = x_s \\
& u_k = \pi_k(x_k) && k = 0, \dots, T-1 \\
& w_k = \Delta_k(g(x_k, u_k, w_k)) && k = 0, \dots, T-1,
\end{aligned}
$$

in a sequential convex programming fashion.

Robust Differential Dynamic Programming



$$\min_{(u_k)} \left(\sum_{k=0}^{T-1} f_0(x_k, u_k, w_k) + V_T(x_T) \right)$$

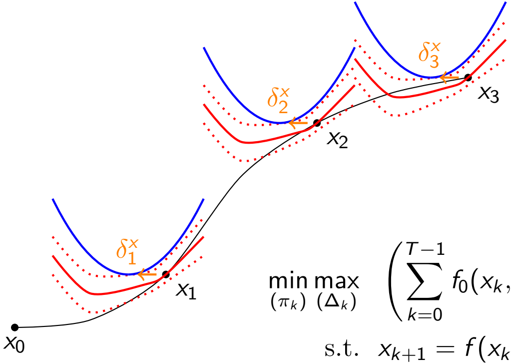
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k) \quad k = 0, \ldots, T-1$$

$$x_0 = x_s.$$

Classical DDP strategy:

- Approximate the cost around the reference trajectory $(x_k, u_k)_{k=0}^{T}$.
- Compute local update (δ_k^x, δ_k^u) that reduces the cost of the trajectory.

Robust Differential Dynamic Programming

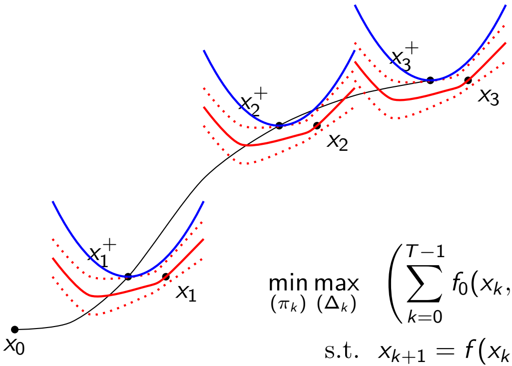


$$
\begin{aligned}
\min_{(\pi_k)} \max_{(\Delta_k)} \quad & \left(\sum_{k=0}^{T-1} f_0(x_k, u_k, w_k) + V_T(x_T) \right) \\
\text{s.t.} \quad & x_{k+1} = f(x_k, \pi_k(x_k), \Delta_k(x_k)) \\
& x_0 = x_s.
\end{aligned}
$$

Robust DDP strategy:

- Utilize uncertainty multipliers to bound the cost around the reference trajectory $(x_k, u_k)_{k=0}^{T}$.
- Compute local update (δ_k^x, δ_k^u) that reduces the cost upper bound of the trajectory.

Robust Differential Dynamic Programming



$$\min_{(\pi_k)} \max_{(\Delta_k)} \left(\sum_{k=0}^{T-1} f_0(x_k, u_k, w_k) + V_T(x_T) \right)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, \pi_k(x_k), \Delta_k(x_k))$$

$$x_0 = x_s.$$

Robust DDP strategy:

- Utilize uncertainty multipliers to bound the cost around the reference trajectory $(x_k, u_k)_{k=0}^{T}$.
- Compute local update (δ_k^x, δ_k^u) that reduces the cost upper bound of the trajectory.

Robust DDP: Robust performance benchmark

Robust DDP: Robust performance benchmark

Example	uncertainties
Ex. 1	wind field 1 with multiplicative uncertainty
Ex. 2	uncertain input delay
Ex. 3	uncertain velocity
Ex. 4	uncertain velocity and input delay
Ex. 5	uncertain velocity, input delay and wind
Ex. 6	wind field 2 with multiplicatie uncertainty
Ex. 7	wind field 2 with constant uncertainty
Ex. 8	wind field 2 with LIDAR uncertainty
Ex. 9	wind field 3 with multiplicative uncertainty

Table: List of example configurations for the Monte Carlo simulation. Note that in Example 2, there is an exception. In this case, we study the 4 DOF model with the prescribed uncertainties, but the six DOF model with no uncertainties.

Robust DDP: Robust performance benchmark

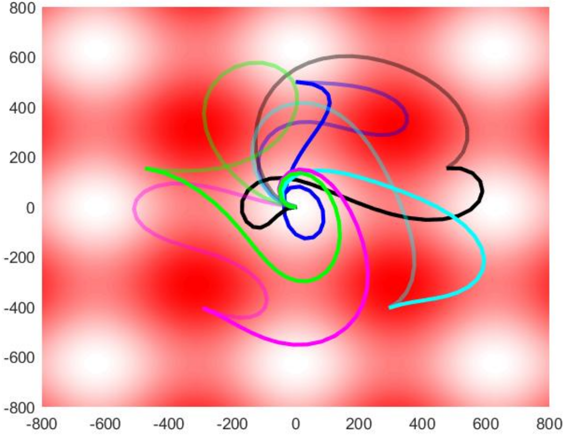


Figure: Illustration of nominal (transparent) and robust (dark) trajectories and the intensity of the uncertainty (red background).

Robust DDP: Robust performance benchmark

Experiment	Algorithm	Terminal Cost	Missed Distance	Missed Heading
Example 1	nominal DDP	1.8862	43.255 m	2.2311°
Example 1	robust DDP	0.21558	14.2683 m	1.9848°
Example 2	nominal DDP	0.22312	13.6583 m	3.4648°
Example 2	robust DDP	0.42834	13.0901 m	9.1851°
Example 3	nominal DDP	0.029267	3.5746 m	2.3266°
Example 3	robust DDP	0.025571	4.7954 m	0.9195°
Example 4	nominal DDP	0.39618	19.8599 m	0.7608°
Example 4	robust DDP	0.071304	8.2416 m	1.0535°
Example 5	nominal DDP	-	-	-
Example 5	robust DDP	-	-	-
Example 6	nominal DDP	0.16619	12.8245 m	0.75264°
Example 6	robust DDP	0.11103	10.4436 m	0.80161°
Example 7	nominal DDP	4.2123	64.7013 m	2.9226°
Example 7	robust DDP	0.51526	22.4701 m	1.8431°
Example 8	nominal DDP	0.44822	21.0997 m	0.99691°
Example 8	robust DDP	0.28449	16.6594 m	1.5105°
Example 9	nominal DDP	2.2398	47.2049 m	1.9455°
Example 9	robust DDP	2.3306	47.8559 m	3.643°

Color code: best, worst.

Robust DDP: Sener simulator benchmark

Simulator Monte Carlo benchmark

1D wind scenario without constraints:

- Parafoil payload landing scenario.
- Disturbances by a height dependent wind field.

3D wind scenario with constraints:

- Parafoil payload landing scenario.
- Disturbances by a wind field depending on all space coordinates.
- Valley constraints.

1D wind scenario without constraints

algorithm	wind error	90% quantiles	mean accuracy	median accuracy
Baseline	$0 \mathrm{m/s}$	27.8517 (27.6395)	27.5154 (19.5091)	16.1469 (19.3674)
Baseline	$\frac{2.5}{3} \mathrm{m/s}$	39.9353 (35.45)	34.9835 (22.5425)	17.594 (20.651)
Baseline	$\frac{5}{3} \mathrm{m/s}$	97.5034 (100.233)	59.1223 (39.3975)	21.4595 (22.7324)
Baseline	$\frac{7.5}{3} \mathrm{m/s}$	185.8093 (252.9832)	108.3531 (104.7632)	26.7415 (30.2102)
DDP	$0 \mathrm{m/s}$	24.7212 (27.2009)	20.2993 (15.7265)	11.7073 (17.0487)
DDP	$\frac{2.5}{3} \mathrm{m/s}$	38.3248 (47.2506)	28.8612 (22.0739)	14.4037 (16.3496)
DDP	$\frac{5}{3} \mathrm{m/s}$	82.8809 (97.7824)	52.1305 (36.6703)	20.4581 (18.8947)
DDP	$\frac{7.5}{3} \mathrm{m/s}$	145.9398 (182.7431)	94.1244 (64.4572)	27.9901 (32.1246)
robust DDP	$0 \mathrm{m/s}$	21.4472 (28.3516)	19.8503 (17.7563)	11.6286 (14.1737)
robust DDP	$\frac{2.5}{3} \mathrm{m/s}$	24.3172 (36.6898)	21.7991 (23.5991)	12.7784 (15.4815)
robust DDP	$\frac{5}{3} \mathrm{m/s}$	49.698 (88.8988)	38.3896 (44.01)	16.2578 (20.0285)
robust DDP	$\frac{7.5}{3} \mathrm{m/s}$	127.2579 (147.6379)	80.0665 (69.2864)	22.0501 (31.0435)

Color code: best, middle, worst.

3D wind scenario with constraints

algorithm	wind error	90% quantiles	mean accuracy	median accuracy
Baseline	$0\mathrm{m/s}$	95.7631 (71.0752)	51.9363	25.2396
Baseline	$\frac{2.5}{3}\mathrm{m/s}$	130.8552 (91.2114)	67.8491	27.0211
Baseline	$\frac{5}{3}\mathrm{m/s}$	215.6164 (190.0191)	93.7852	31.8473
Baseline	$\frac{7.5}{3}\mathrm{m/s}$	328.166 (314.3696)	131.7685	43.6718
DDP	$0\mathrm{m/s}$	39.5696 (60.384)	50.5451	19.3913
DDP	$\frac{2.5}{3}\mathrm{m/s}$	73.7915 (147.7287)	67.9683	21.9754
DDP	$\frac{5}{3}\mathrm{m/s}$	146.9501 (160.418)	92.1789	26.9091
DDP	$\frac{7.5}{3}\mathrm{m/s}$	259.5951 (221.2534)	116.3121	38.8167
robust DDP	$0\mathrm{m/s}$	40.3008 (139.2251)	56.2837	16.8733
robust DDP	$\frac{2.5}{3}\mathrm{m/s}$	73.7171 (152.2869)	59.6432	19.7793
robust DDP	$\frac{5}{3}\mathrm{m/s}$	152.6772 (-)	83.5045	24.6162
robust DDP	$\frac{7.5}{3}\mathrm{m/s}$	271.3544 (229.1841)	131.6672	34.4236

Color code: best, middle, worst.

Robust DDP & Learning (Joint work with Lukas Hewing)

Robust DDP & Learning (Joint work with Lukas Hewing)

Learned system models can be inaccurate!

- Model errors exist even in the proximity of the training set.
- Model errors can increase significantly, outside of the *support* of the training set.

Idea: Use robust planning for learned models.

- Improves performance in the data domain.
- Avoids leaving the data domain.

Robust DDP & Learning (Joint work with Lukas Hewing)

Studied setup:

Nominal model

$$\dot{x}(t) = v(t)\cos(\gamma(t))\cos(\psi(t))$$
$$\dot{y}(t) = v(t)\cos(\gamma(t))\sin(\psi(t))$$
$$\dot{\psi}(t) = \frac{L(v(t))\sin(u(t))}{mv(t)\cos(\gamma(t))}$$
$$\dot{v}(t) = -D(v(t))/m - g\sin(\gamma(t))$$
$$\dot{\gamma}(t) = \frac{L(v(t))\cos(u(t)) - mg\cos(\gamma(t))}{mv(t)}$$
$$\dot{z}(t) = v(t)\sin(\gamma(t))$$
$$\dot{u}(t) = \frac{u_{\mathrm{com}} - u(t)}{\tau_u}.$$

Learned model

$$\dot{x}(t) = \bar{v}\cos(\psi(t))$$
$$\dot{y}(t) = \bar{v}\sin(\psi(t))$$
$$\dot{\psi}(t) = \bar{c}u(t) + w(t)$$
$$\dot{u}(t) = \mu(u(t), u_{\mathrm{com}})$$
$$\|w(t)\| \leq 2\sigma(u(t), u_{\mathrm{com}}).$$

Here, $\mu(u, u_{\mathrm{com}})$ and $\sigma(u, u_{\mathrm{com}})$ are the mean and standard deviation of a gaussian process.

Robust DDP & Learning (Joint work with Lukas Hewing)

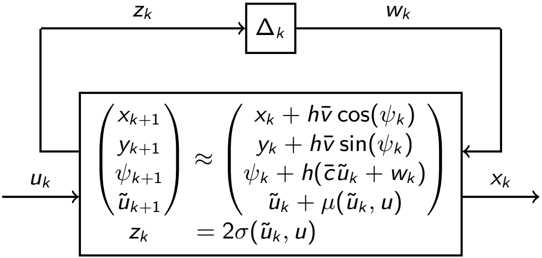
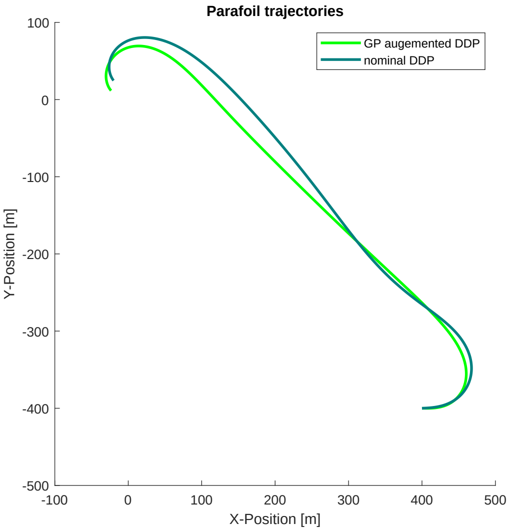


Figure: Generalized plant from the GP augemented model.

The uncertainties of the above generalized plant are characterized by $\|\Delta_k\| \leq 1$, i.e., the family of multipliers

$$\begin{pmatrix} \Delta_k \\ I \end{pmatrix}^\top \begin{pmatrix} -\lambda I & 0 \\ 0 & \lambda I \end{pmatrix} \begin{pmatrix} \Delta_k \\ I \end{pmatrix} \succeq 0 \qquad \forall \lambda \in \mathbb{R}_{\geq 0}.$$

Robust DDP & Learning (Joint work with Lukas Hewing)



Robust DDP & Learning (Joint work with Lukas Hewing)

Statistics	GP augmented DDP	nominal DDP
90% Terminal Cost Quantile	0.31043	1.4633
Terminal Cost Median	0.060813	0.24776
Terminal Cost Mean	0.1607	0.69561
90% Position Error Quantile	15.2541	21.5877
Position Error Median	7.2427	8.8074
Position Error Mean	8.2827	11.0035
90% Heading Error Quantile	4.6454	19.4048
Heading Error Median	0.77642	6.4613
Heading Error Mean	2.0768	9.0401

Color code: best, worst.

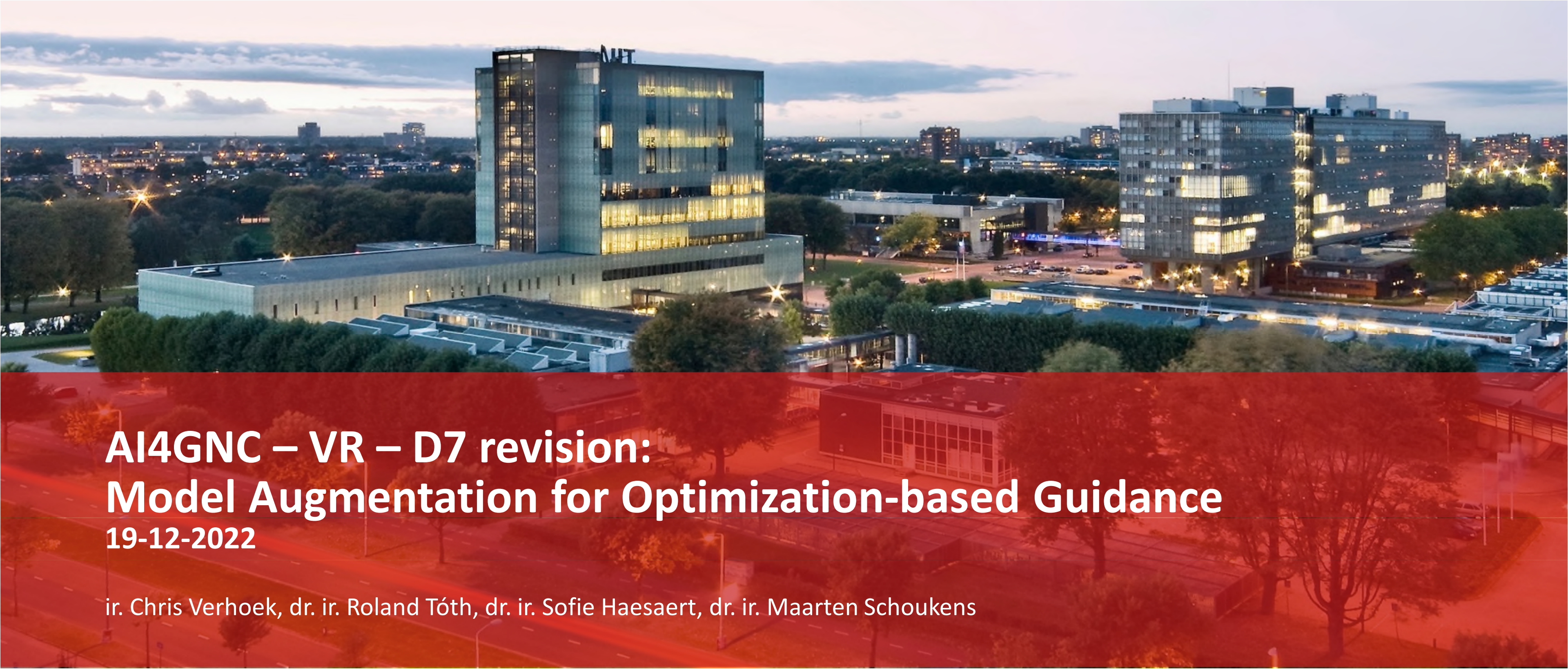
Conclusion

- Robust planning can improve robustness over a robust controller.
- Robust DDP has demonstrated the advantages of robust planning in extensive benchmarks.
- Robust planning can be a useful addition to learning based control methods.

Conclusion

Thank you:
Valentin Preda,
Samir Bennani,
Lukas Hewing and
Sener.

For all the support during our work for AI4GNC.



AI4GNC – VR – D7 revision:
Model Augmentation for Optimization-based Guidance

19-12-2022

ir. Chris Verhoek, dr. ir. Roland Tóth, dr. ir. Sofie Haesaert, dr. ir. Maarten Schoukens

Overview

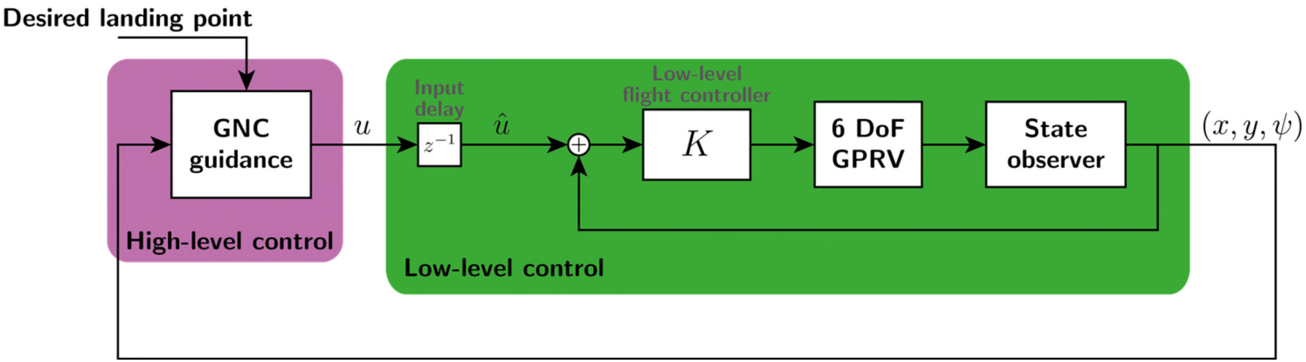
- Overviewed the state-of-the-art LPV identification and control (D3)
- Developed an LFR-ANN based model-augmentation concept (D4)
- Developed 6/12 DOF models of the parafoil return vehicle (de Lange, 2021) (D4)
- Applied model-augmentation concept on the developed models (D4)
    - LTI baseline augmentation → excellent results (BFRs of ~95%)
- Investigated augmentation of the unicycle baseline (D6)

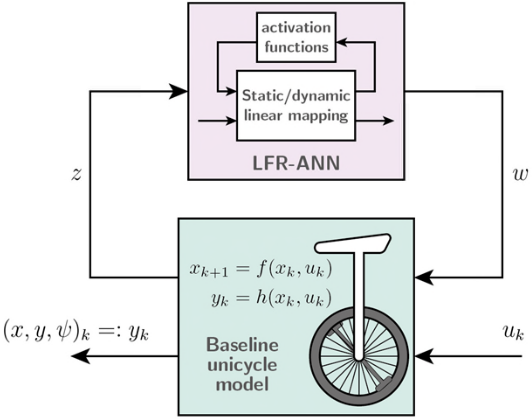
Next step?

Problem setting

Goal: Integrate the augmented model in guidance

- ➢ Capture closed-loop flight dynamics as an LFR-ANN augmented unicycle





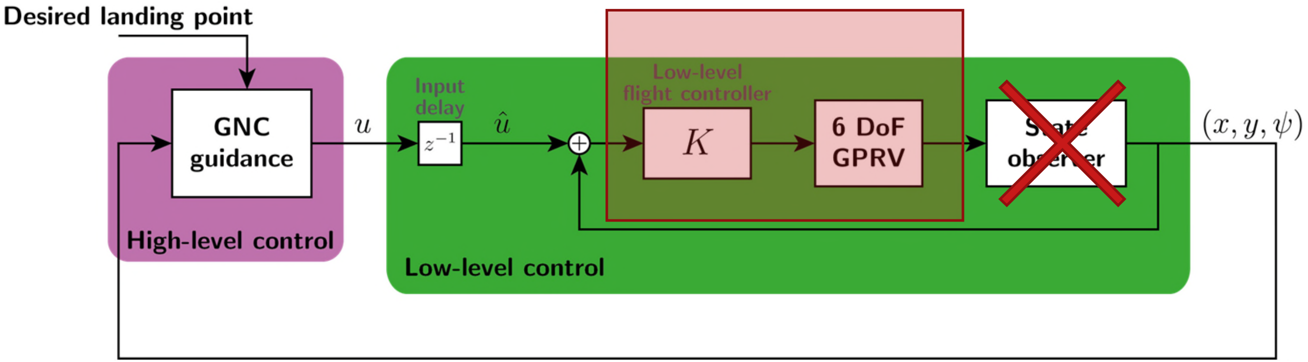
Contents

- **Augmentation scenario**
- Data-generation
- Preliminary analysis
- Training results
- Conclusions

Augmentation scenario

Simulation environment: based on TU/e simulator (developed in D4)

- 6 DOF vehicle model
- LPV controller designed for reference tracking of ψ
- Generated noise to match state observer error (based on recorded specturm)



Augmentation scenario

Velocity-based training on the unicycle model

$$\boxed{\begin{aligned}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\psi}(t)\end{aligned}} \begin{aligned}&= v\cos(\psi(t)) + w_x(t)\\ &= v\sin(\psi(t)) + w_y(t)\\ &= u(t)\end{aligned}$$

State derivatives → Outputs

$$\dot{\psi} = \dot{\psi}_{\text{ref}}$$

$$z = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} v_{\text{forward}} \sin\psi + d_{\text{wind},x} \\ v_{\text{forward}} \cos\psi + d_{\text{wind},y} \\ \dot{\psi}_{\text{ref}} \end{pmatrix}$$

Augmentation scenario

Velocity-based training – new baseline model

$$\dot{\psi} = \dot{\psi}_{\text{ref}}$$

$$z = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} v_{\text{forward}} \sin \psi + d_{\text{wind},x} \\ v_{\text{forward}} \cos \psi + d_{\text{wind},y} \\ \dot{\psi}_{\text{ref}} \end{pmatrix}$$

Approximation of $\dot{x}, \dot{y}$ by the model:

- Dependent on v_{forward} → norm of $\dot{x}, \dot{y}$
- Dependent on $d_{\text{wind},\bullet}$ → noise

➢ Only dependent on the approximation error of ψ , i.e., $\dot{\psi}$

Augmentation scenario

Quality of baseline model only dependent on the mapping

$$\dot{\psi} = \dot{\psi}_{\mathrm{ref}}$$

- Identity mapping

➢ Identification problem: Estimate $\hat{\dot{\psi}} = \mathcal{M}\dot{\psi}_{\mathrm{ref}}$ such that $\hat{\dot{\psi}} - \dot{\psi}$ is close to zero.

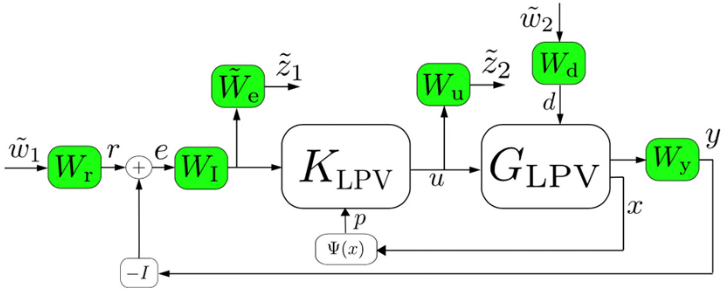
Solve this identification problem in the model-augmentation framework

Contents

Data-generation

Use our own model:

- 6 DoF GPRV
- Controlled with LPV controller (Matthis' MSc work)
  - Reference tracking scenario
  - Tracks ψ
- Use heading-rate reference from SENER simulator

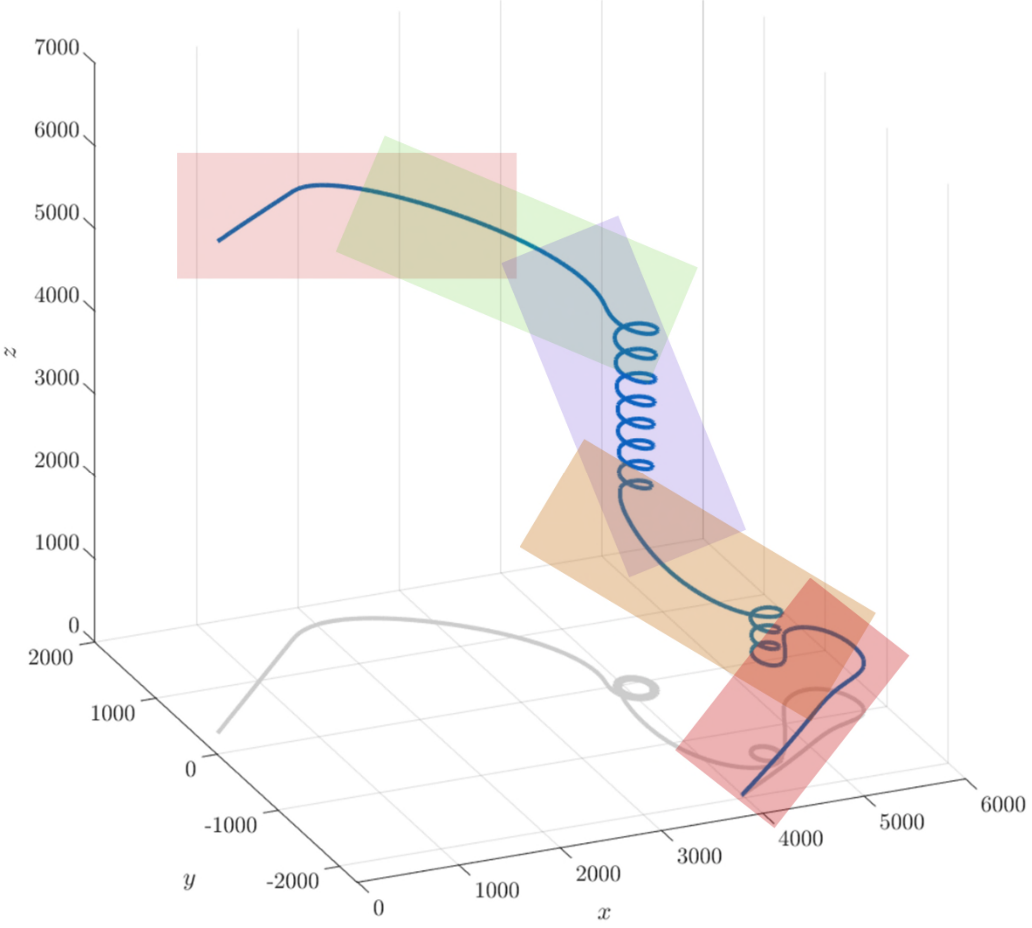


Divide reference in sub-windows

Data-generation

Divide reference in sub-windows

- Simulate sub-windows with ball of random initial conditions
- 12 sub-windows, 25 initial conditions
  - 300 trajectories, T_s = 0.1 [s], 250 [s]
  - 750k datapoints
  - Generate with/without wind

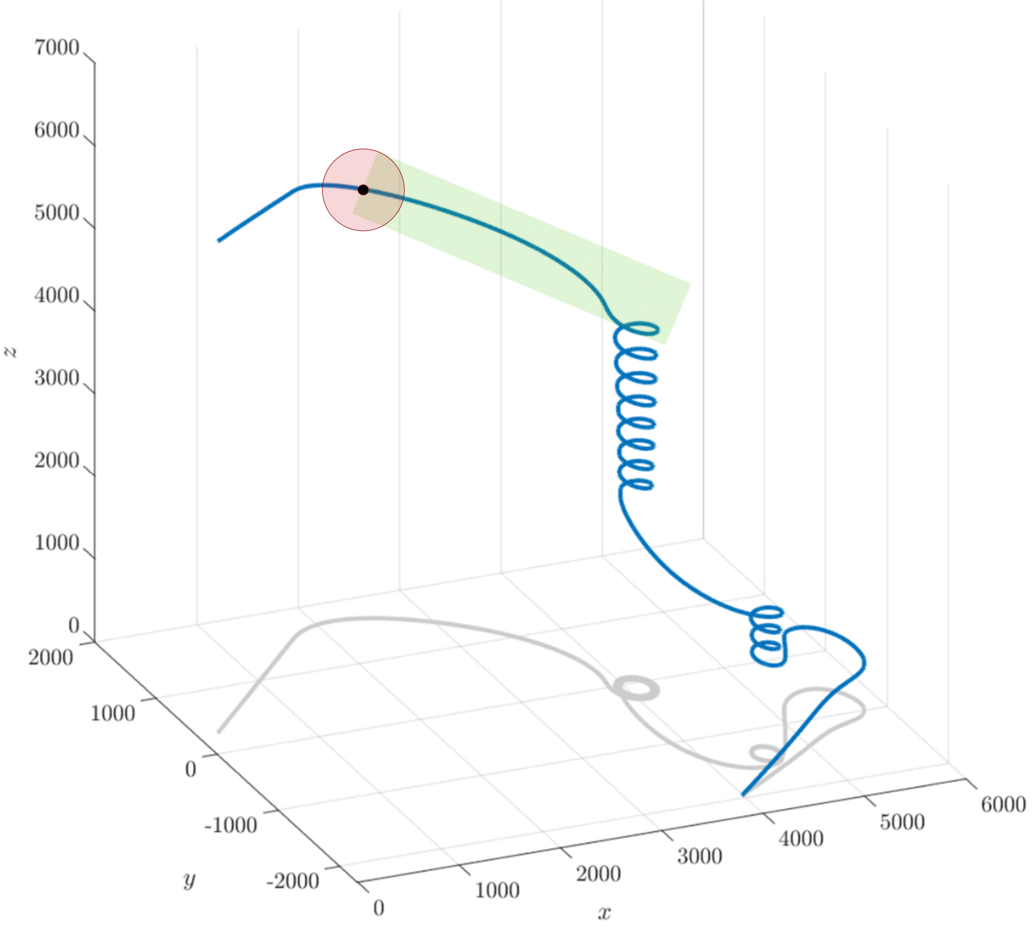


Data-generation

Divide reference in sub-windows

- Simulate sub-windows with ball of random initial conditions
- 12 sub-windows, 25 initial conditions
    - 300 trajectories
    - 750k datapoints
    - Generate with/without wind (seen as the noise source)

Size of ball initial conditions:
15% deviation from nominal



Contents

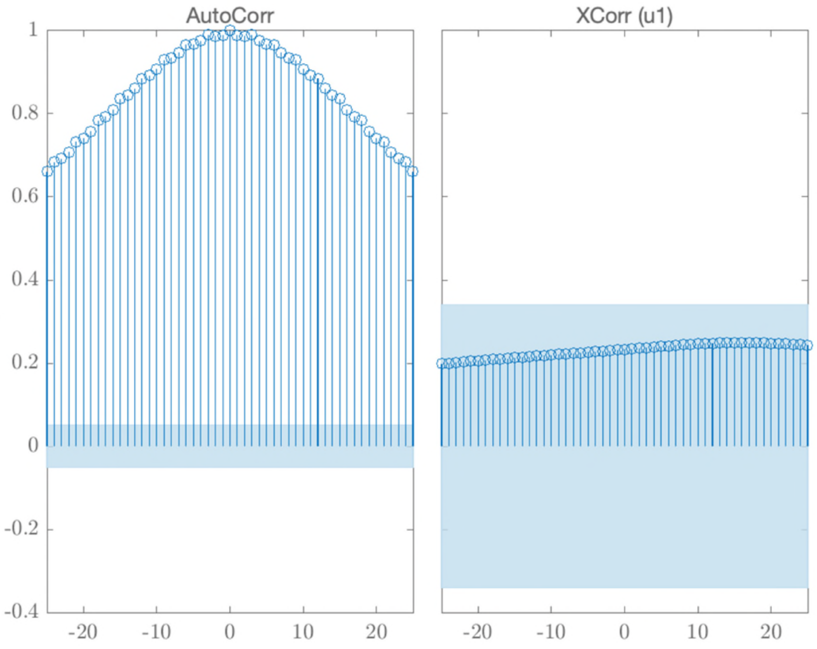
- Augmentation scenario
- Data-generation
- **Preliminary analysis**
- Training results
- Conclusions

Preliminary analysis

Q: How nonlinear the behavior is?

Identify the mapping $\hat{\dot{\psi}} = \mathcal{M}\dot{\psi}_{\mathrm{ref}}$ with LTI system identification techniques

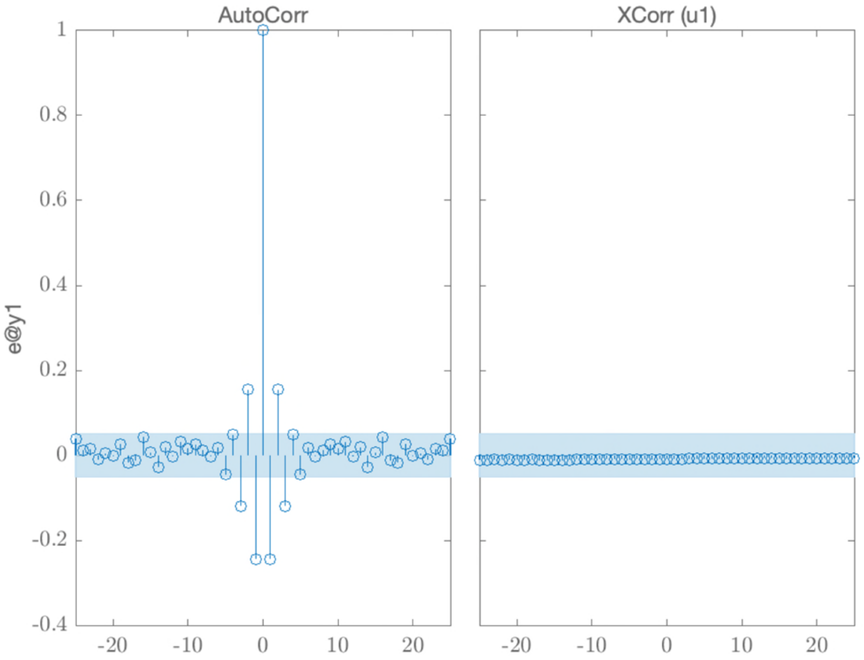
- Use wind/noise-free data → **OE-estimation**
- Process part estimated well with order 2 → mainly linear behavior with minor nonlinearities
- Best-Fit-Rate (BFR) of 99.43% on validation data
- With wind disturbance: BFR of 89.66%



Preliminary analysis

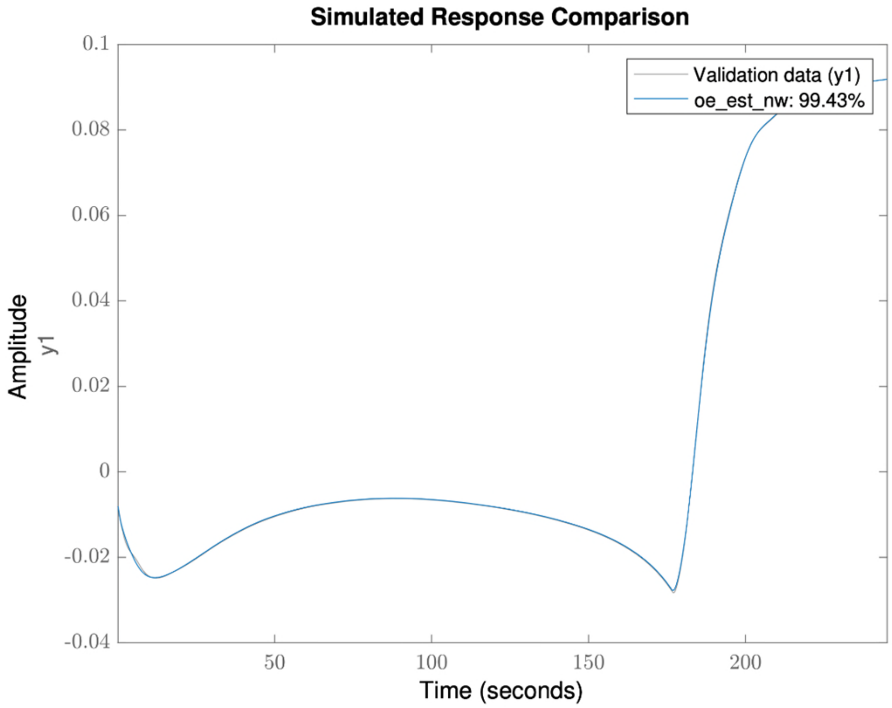
Identify the mapping $\hat{\dot{\psi}} = A\dot{\psi}_{\mathrm{ref}}$ with LTI system identification techniques

- Use wind-infected data → **BJ-estimation**
- Best-Fit-Rate (BFR) of 89.67% on validation data
- Autocorrelation can not be brought further down
- Best-Fit-Rate on noise/wind-free data: 99.37%

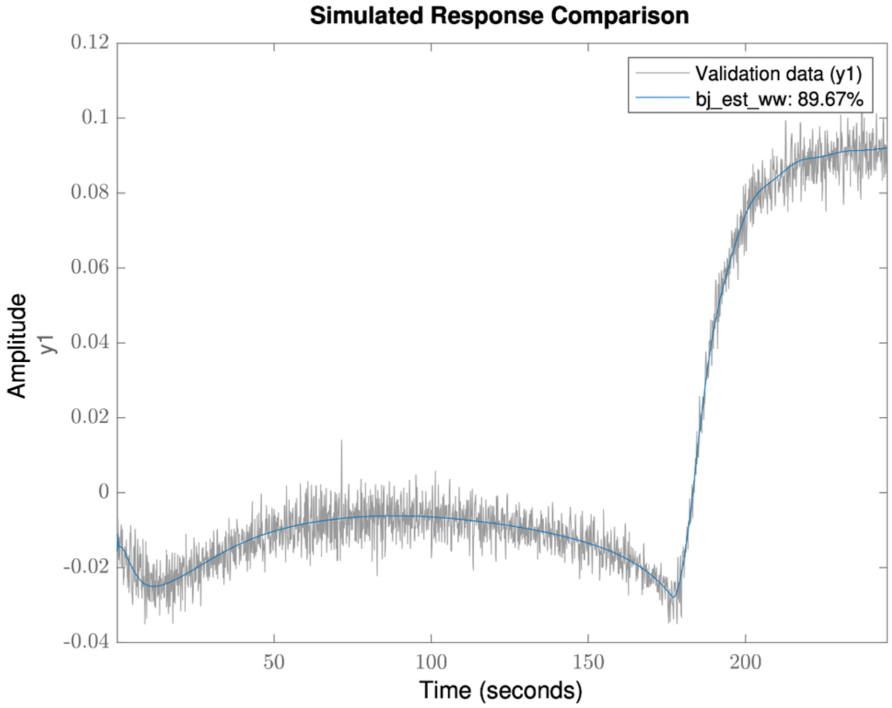


Preliminary analysis

OE simulation results



BJ simulation results



Preliminary analysis

However, there were a few data-sets where the fit drasticaly dropped.

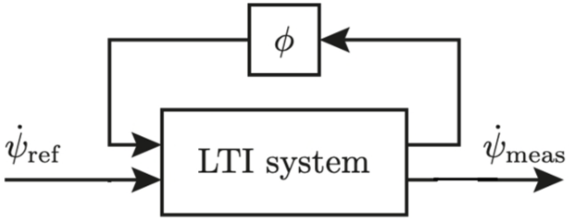
- Worst-case BFR: 65%
- The controller dropped out from its designed operating range
- Main reason for the dominant LTI behavior: linearizing effect of the LPV controller

Contents

Training results

We want to learn the residual nonlinearities in the mapping $\dot{\psi}_{\mathrm{ref}} \rightarrow \dot{\psi}$

- See the identified LTI model as baseline
- Augment with a static neural network



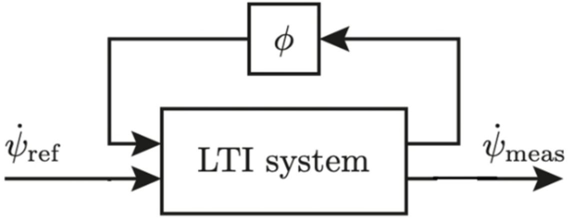
Goal: Accurately learn the residual nonlinearities in the mapping

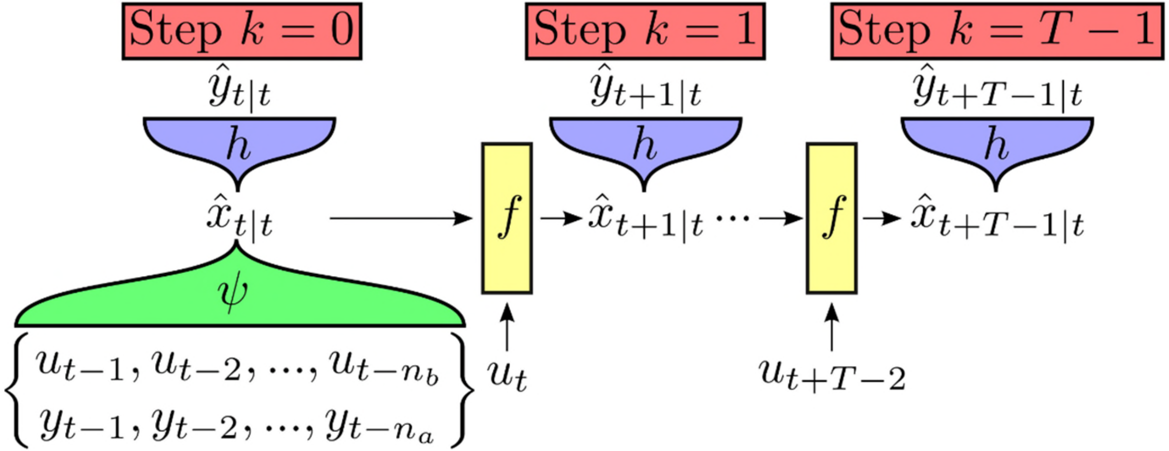
- ➢ Use noise and wind-free data
    - ➢ Avoid *learning* wind / noise

Training results

Train in SUBNET framework

- Number of inputs of the network: 2
- Number of outputs of the network: 2
- Number of hidden layers: 2
- Nodes per layer: 64
- Activation functions: tanh()

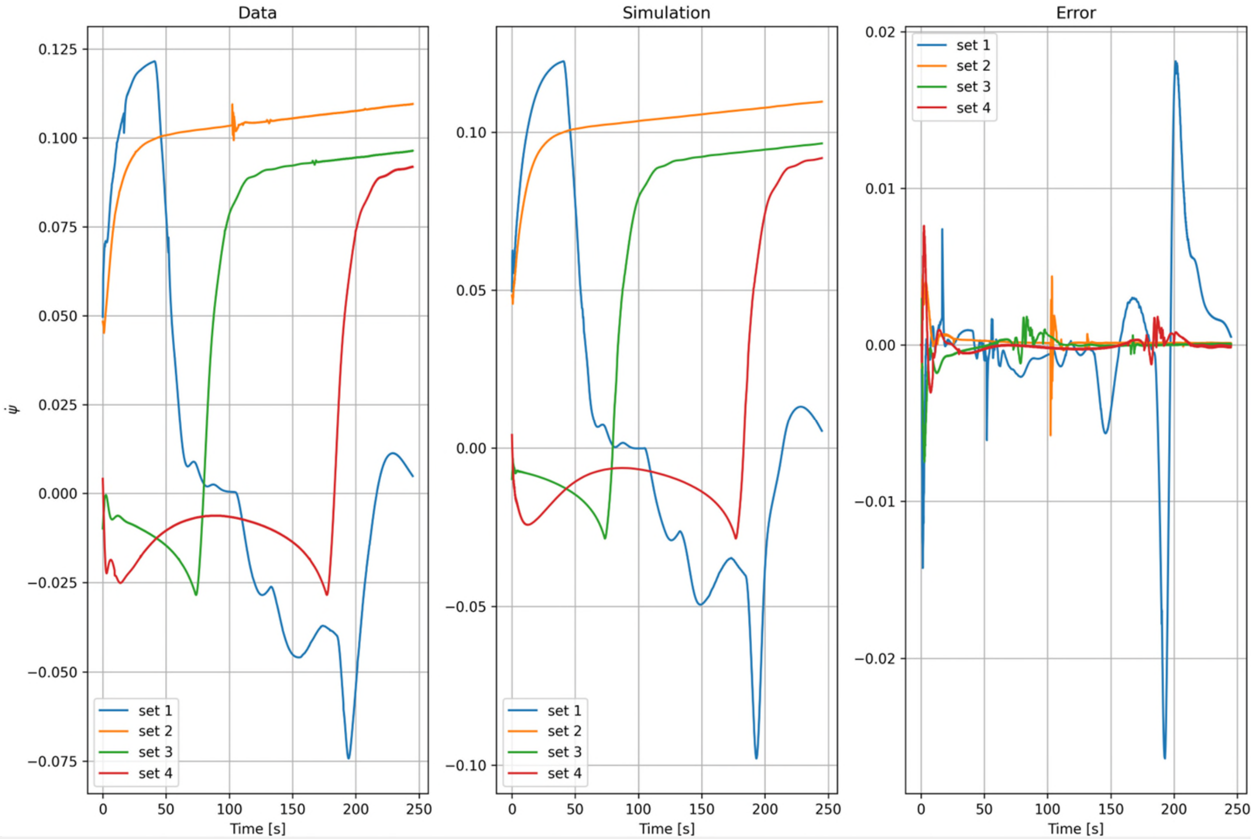




Training results

Augmented model:
worst-case BFR on
the validation set: **85%**

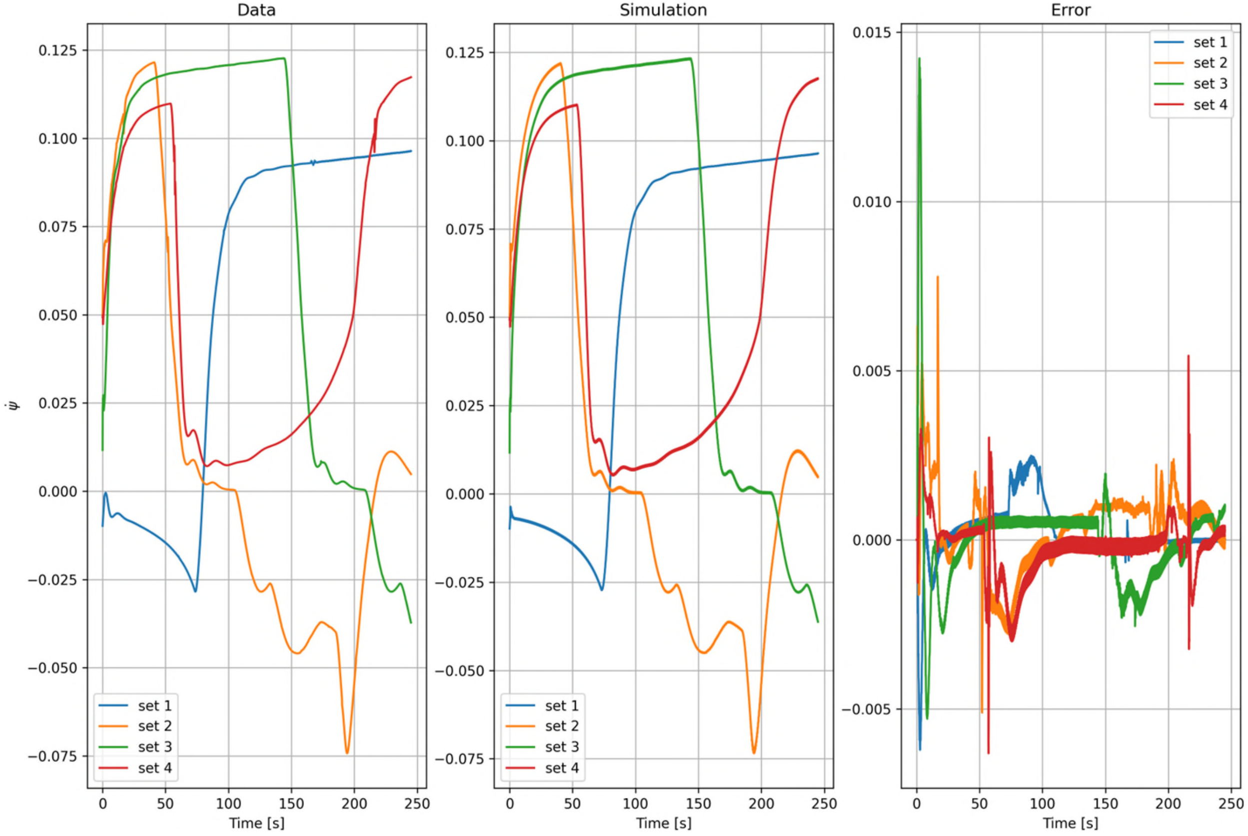
Simulation on 4 test sets:



Training results

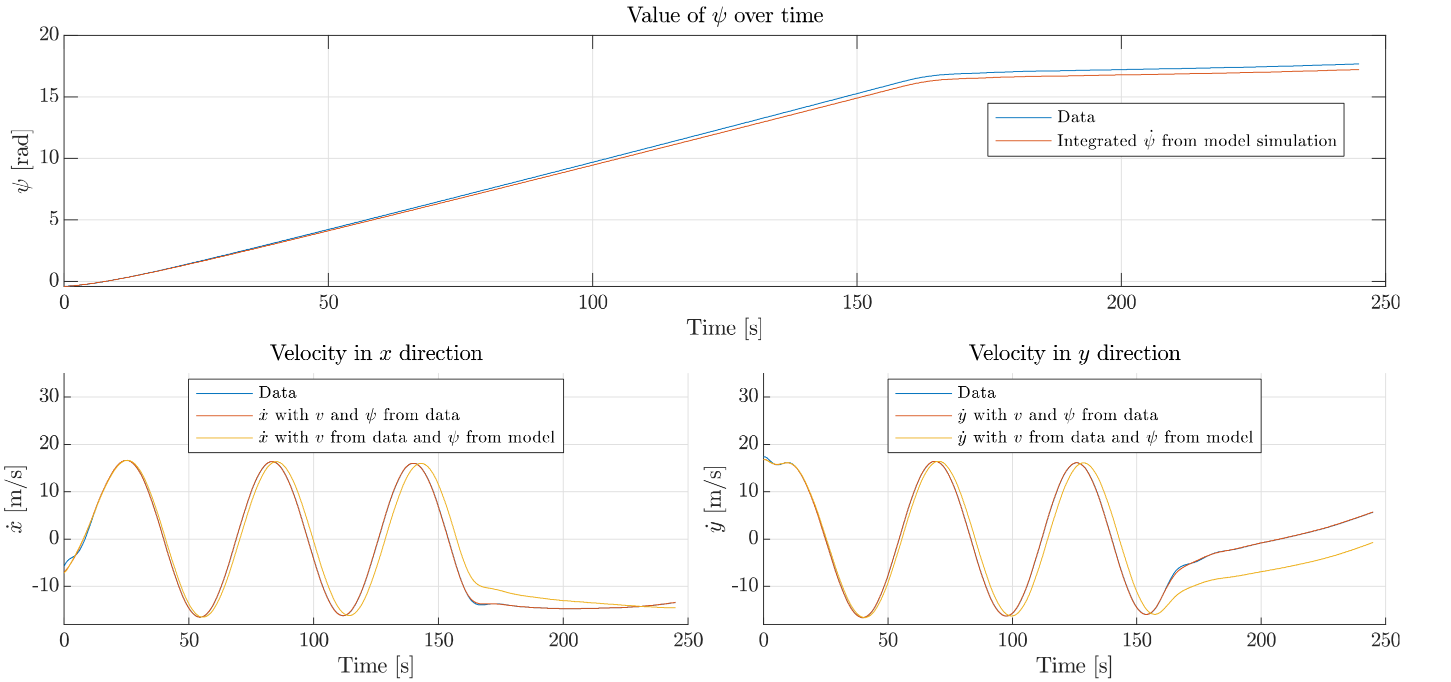
Full black-box model:
worst-case BFR on
the validation set **93%**

Simulation on 4 test sets:



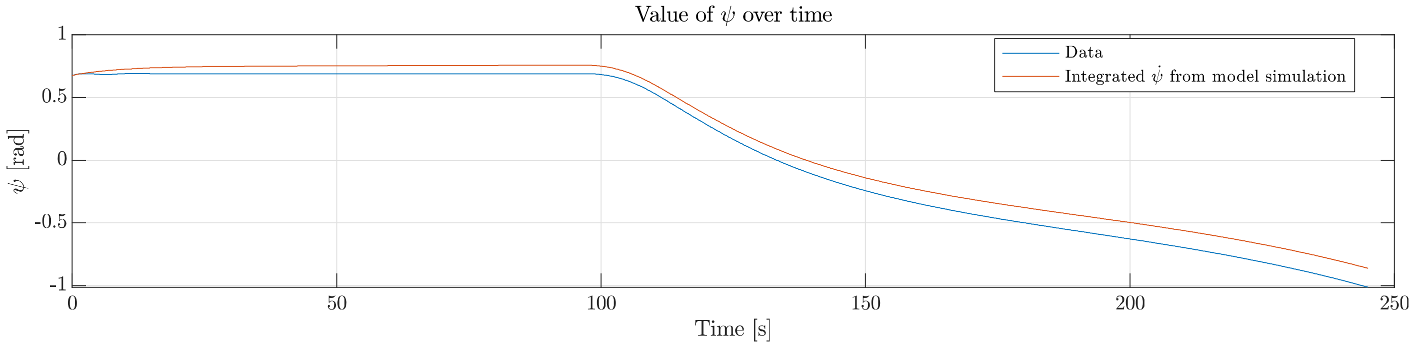
Training results

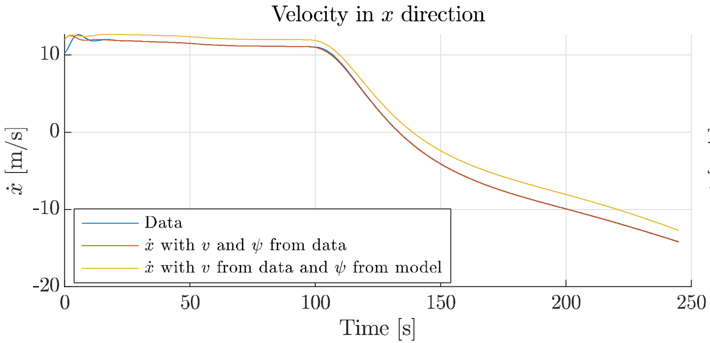
Velocity simulation (augmented model)

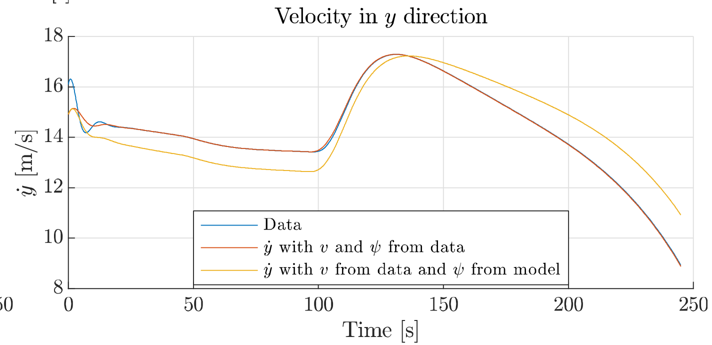


Training results

Velocity simulation (black-box model)







Training results

- For data-sets where LTI behavior is dominant, augmentation structure contributes: <1%
- Black-box approach reaches better validation results (effect of SGO + regularization)
- Long-term predictions (simulations) have less error with the augmented LTI model
- Performance of the augmented structure dependent on the initialization of ANN (use the encoder)
- Overall performance is excellent

Contents

Conclusions

- For training on velocity, identification problem resembles to learning $\dot{\psi}_{\mathrm{ref}} \rightarrow \dot{\psi}$
- We have shown that we can learn the residual nonlinear dynamics
- For small deviations from nominal trajectory, LTI augmentation is sufficient
- Outside of this region → ANNs are needed
- Due to LFR-ANN structure, augmentation can be used to define uncertainty
    - Working this properly out requires more time...