`dkpy`: Robust Control with Structured Uncertainty in Python

Timothy Everett Adams   Steven Dahdah   James Richard Forbes

McGill University, Department of Mechanical Engineering

July 9, 2026

Linear Robust Control

Motivation

- The models used in control design/analysis never perfectly represent the physical system.
  - Unmodeled dynamics
  - Inexact parameters
  - Varying operating conditions
- In certain applications, it is necessary to explicitly consider the model uncertainty.

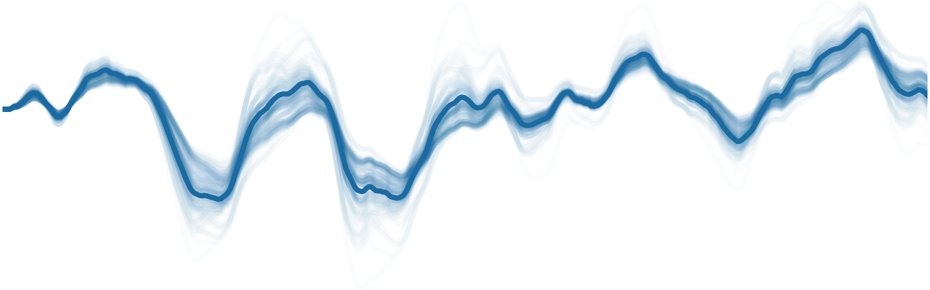


Figure 1: Illustration of uncertain model response.

Structured Uncertainty

Structured Uncertainty Robustness

- In complex interconnected systems, the uncertainty is structured.
  - Localized uncertainty in sub-systems.
- Robust stability and performance is assessed using the structured singular value μ (SSV) [1], [2], [3].

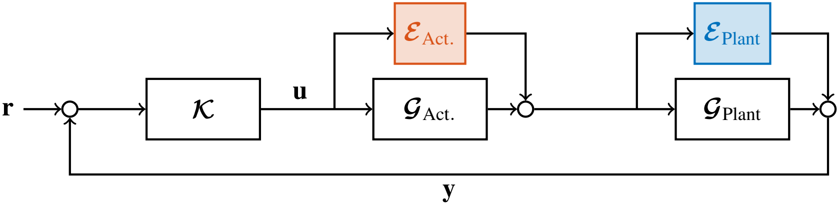


Figure 2: Uncertain feedback interconnection with structured uncertainty.

Robust Control Software

Contribution

- **Problem**: Software for robust control using μ-analysis and μ-synthesis.
  - MATLAB's *Robust Control Toolbox* [4]
- **Contribution**: **`dkpy`**
  - Open-source Python library for linear robust control design and analysis for structured uncertainty.
  - Integration with Python control ecosystem (`python-control` and `slycot`).

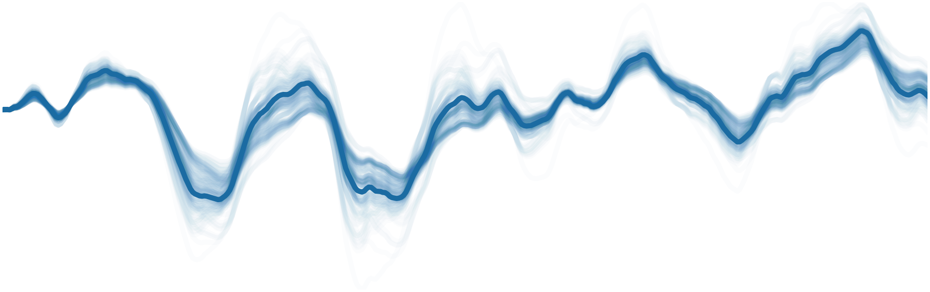


Figure 1: Illustration of uncertain model response.

Robust Stability Problem

Robust Stability Problem [2], [3]

Systems

- Generalized Plant $\mathcal{P}$
- Controller $\mathcal{K}$
- Normalized Perturbation $\boldsymbol{\Delta}$ ($\|\boldsymbol{\Delta}\|_\infty \leq 1$)

Uncertainty Structure

$$\boldsymbol{\Delta} = \begin{bmatrix} \boldsymbol{\Delta}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Delta}_M \end{bmatrix} \tag{1}$$

Robust Stability

- The interconnection between $\mathcal{P}$ and $\mathcal{K}$ is internally stable for all admissible perturbations $\boldsymbol{\Delta}$

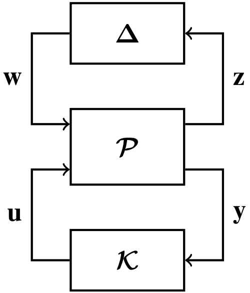


Figure 3: Robust stability problem.

Robust Performance Problem

Equivalence of Robust Performance and Robust Stability [2, Theorem 10.8]

Robust performance of Figure 4

$$\|\mathbf{w}_2 \mapsto \mathbf{z}_2\|_\infty < 1, \quad \forall \, \|\boldsymbol{\Delta}\|_\infty \leq 1$$

$\iff$ Robust stability of Figure 5

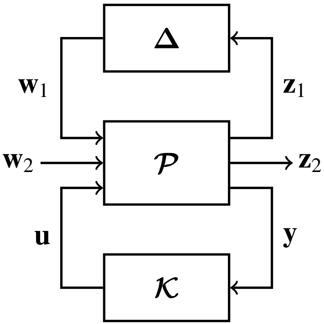


Figure 4: Robust performance problem.

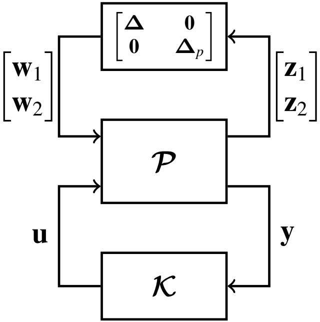


Figure 5: Robust performance problem in robust stability form.

Robust Stability Analysis

Robust Stability Condition

Robust Stability Condition [2, Theorem 10.6]

$$\sup_{\omega \in \mathbb{R}} \mu_{\boldsymbol{\Delta}}(\mathbf{F}_\ell(\mathbf{P}, \mathbf{K})(j\omega)) < 1 \tag{2}$$

Structured Singular Value (SSV) Upper Bound

$$\mu_{\boldsymbol{\Delta}}(\omega) \le \bar{\mu}_{\boldsymbol{\Delta}}(\omega) = \min_{\mathbf{D}_\omega} \bar{\sigma}(\mathbf{D}_\omega \mathbf{F}_\ell(\mathbf{P}, \mathbf{K})(j\omega)\mathbf{D}_\omega^{-1}) \tag{3}$$

$$\mathbf{D}_\omega \boldsymbol{\Delta} = \boldsymbol{\Delta} \mathbf{D}_\omega \tag{4}$$

Practical Robust Stability Condition

$$\bar{\mu}_{\boldsymbol{\Delta}}(\omega) < 1 \quad \forall \omega \in \{\omega_1, \dots, \omega_N\} \tag{5}$$

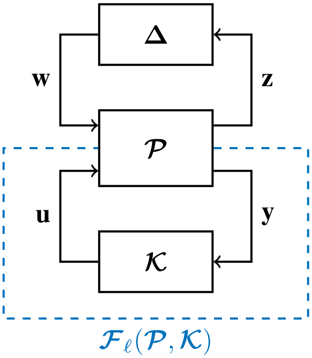


Figure 6: Closed-loop robust stability problem.

Robust Controller Synthesis

μ-Synthesis

μ-Synthesis Problem

$$\min_{\mathbf{K}(s)} \left(\min_{\mathbf{D}(s)} \left\| \mathbf{D}(s)\mathbf{F}_\ell(\mathbf{P},\mathbf{K})(s)\mathbf{D}^{-1}(s) \right\|_\infty \right) \tag{6}$$

$$\mathbf{D}(s)\boldsymbol{\Delta}(s) = \boldsymbol{\Delta}(s)\mathbf{D}(s) \tag{7}$$

- Issue: Not jointly convex in $\mathbf{D}(s)$ and $\mathbf{K}(s)$

DK-iteration
- Heuristic robust controller design procedure
- Minimize $\mu_{\boldsymbol{\Delta}}$ by alternating minimizing with respect to $\mathbf{D}(s)$ and $\mathbf{K}(s)$

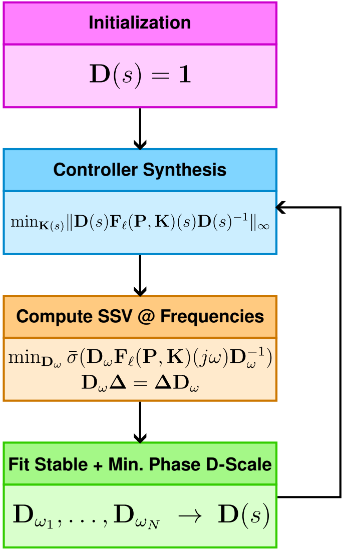


Figure 7: DK-iteration flow chart.

Interfaces for μ-Analysis and μ-Synthesis in `dkpy`

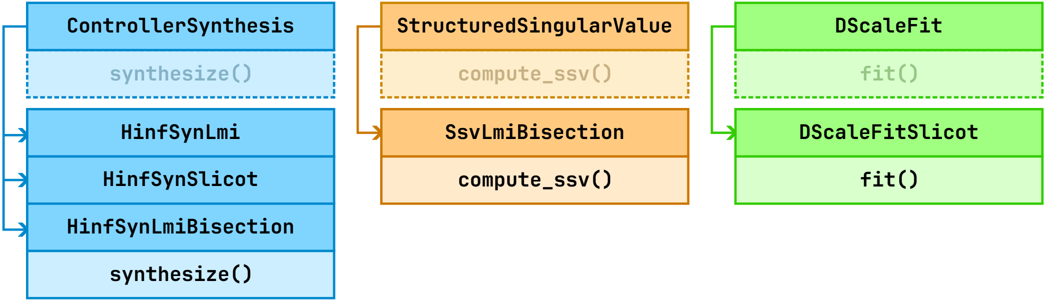


Figure 8: Base interfaces in `dkpy` and their implementations.

Interface for DK-iteration in `dkpy`

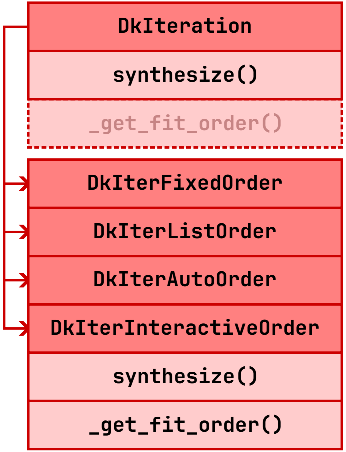


Figure 9: DK-iteration interface in `dkpy` and its implementations.

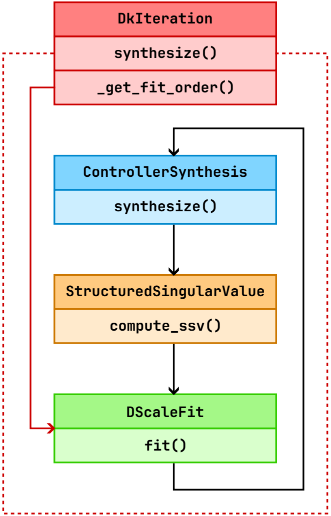


Figure 10: Overview of the DK-iteration `synthesize` function.

Example Problem [6, § 14.1]

Lateral Aircraft Dynamics Control

- Model
  - Linearized lateral aircraft dynamics [5, § 7.3, § B.4]
  - Uncertain actuator dynamics
- Robust Performance
  - Track roll angle reference signals
  - Minimize sideslip angle, roll rate, yaw rate, and actuator effort
  - Mitigate disturbance and sensor noise effects

Figure 11: Douglas DC-8 aircraft[1].

[1] https://en.wikipedia.org/wiki/Douglas_DC-8

Example Problem [6, § 14.1] (cont.)

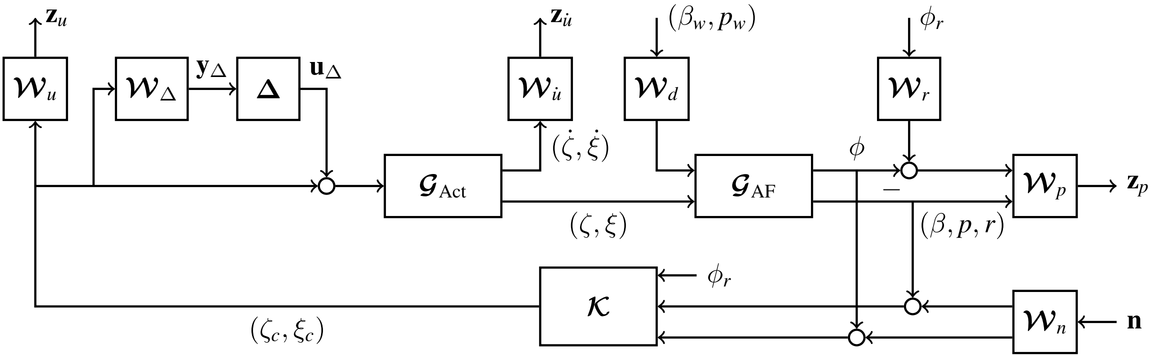


Figure 12: Linearized lateral aircraft dynamics control problem block diagram.

Example: Robust Controller Synthesis Code

```
controller_synthesis=dkpy.HinfSynLmi(
    lmi_strictness=5e-7,
    solver_params=dict(
        solver="MOSEK",
        eps=5e-8,
    ),
)
```

```
structured_singular_value=dkpy.SsvLmiBisection(
    bisection_atol=1e-5,
    bisection_rtol=1e-5,
    max_iterations=1000,
    lmi_strictness=1e-7,
    solver_params=dict(
        solver="MOSEK",
        eps=1e-9,
    ),
)
```

```
d_scale_fit=dkpy.DScaleFitSlicot()
```

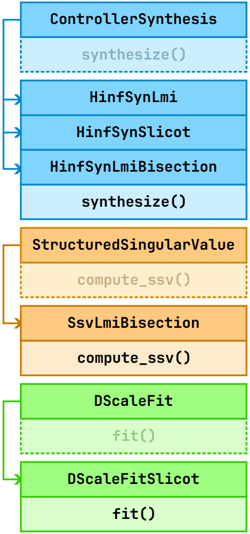


Figure 13: Base interfaces in `dkpy`.

Example: Robust Controller Synthesis Code (cont.)

```python
dk_iter = dkpy.DkIterListOrder(
    controller_synthesis,
    structured_singular_value,
    d_scale_fit,
    fit_orders=[4, 4, 4],
)
```

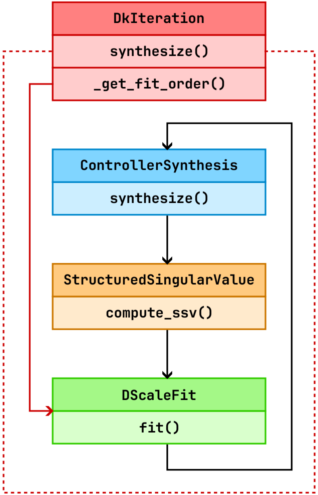


Figure 14: DK-iteration interface.

Example: Robust Controller Synthesis Code (cont.)

```python
# Frequency
freq_min = 0.01
freq_max = 100
num_freq = 100
freq = np.logspace(np.log10(freq_min), np.log10(freq_max), num_freq)
omega = 2 * np.pi * freq

# Uncertainty block structure
block_structure = [
    dkpy.ComplexFullBlock(n_y_delta, n_u_delta),
    dkpy.ComplexFullBlock(n_w, n_z),
]

# Synthesize robust controller with DK-iteration
controller, N, mu, iter_results, info = dk_iter.synthesize(
    plant_gen,
    n_y,
    n_u,
    omega,
    block_structure,
)
```

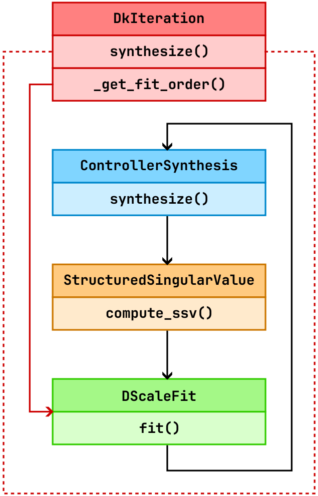


Figure 15: DK-iteration interface.

Example: Robust Controller Synthesis Results

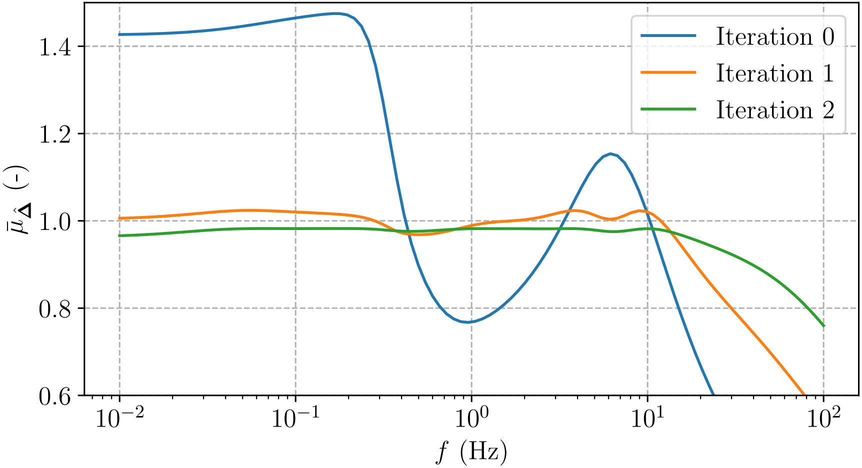
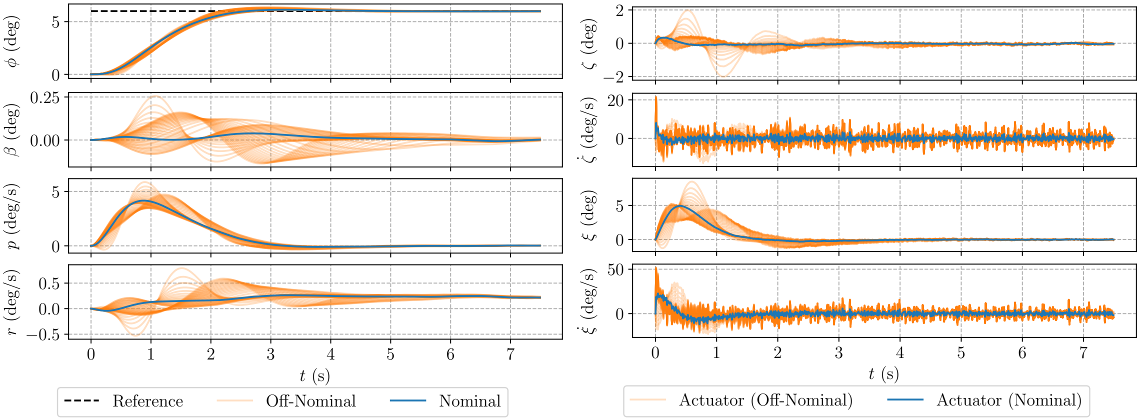


Figure 16: Structured singular value upper bound across iterations of DK-iteration.

Example: Robust Controller Synthesis Results (cont.)



(a) States.

(b) Inputs.

Figure 17: Closed-loop roll angle reference tracking simulation for nominal and perturbed systems.

Conclusion

Summary

- `dkpy` is a Python package for LTI robust control
  - μ-analysis
  - μ-synthesis using DK-iteration

Future Work

- μ-analysis and μ-synthesis for real and mixed real/complex uncertainties [2, § 18.2]
- Lower bound computation for μ to assess conservatism in upper bound [2, § 10.2]

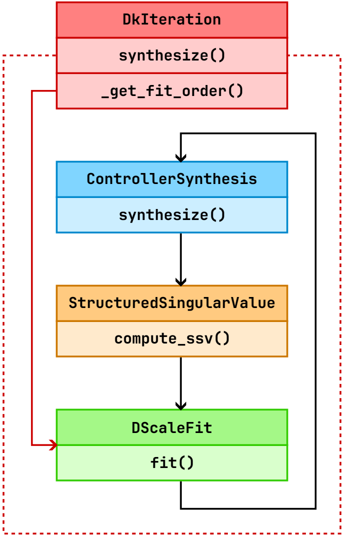


Figure 18: DK-iteration interface.

Questions?

timothy.adams@mail.mcgill.ca

james.richard.forbes@mcgill.ca

Paper

Code

References

[1] A. Packard and J. Doyle, “The complex structured singular value,” *Automatica*, vol. 29, no. 1, pp. 71–109, Jan. 1, 1993.

[2] K. Zhou and J. C. Doyle, *Essentials of Robust Control*, 1st ed. Upper Saddle River, NJ: Prentice Hall, 1998.

[3] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. Chichester, England: Wiley-Interscience, Nov. 4, 2005.

[4] *Robust Control Toolbox*, Mathworks. [Online]. Available: https://www.mathworks.com/products/robust.html.

[5] L. V. Schmidt, *Introduction to Aircraft Flight Dynamics*, 1st ed. Reston, VA: American Institute of Aeronautics and Astronautics, Jan. 1, 1998.

[6] U. Mackenroth, *Robust Control Systems*, 1st ed. Berlin, Heidelberg: Springer, 2004.