Learning-based Robust Control: Guaranteeing Stability while Improving Performance

Felix Berkenkamp and Angela P. Schoellig

IROS 2014 – Machine Learning in Planning and Control of Robot Motion Workshop (14. Sep.)

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

DYNAMIC
SYSTEMS LAB

Why is model-based control not always sufficient?

- Model inaccuracies limit achievable performance!
  - Surface Material
  - Topography
  - Unknown dynamics

How these problems have been tackled so far

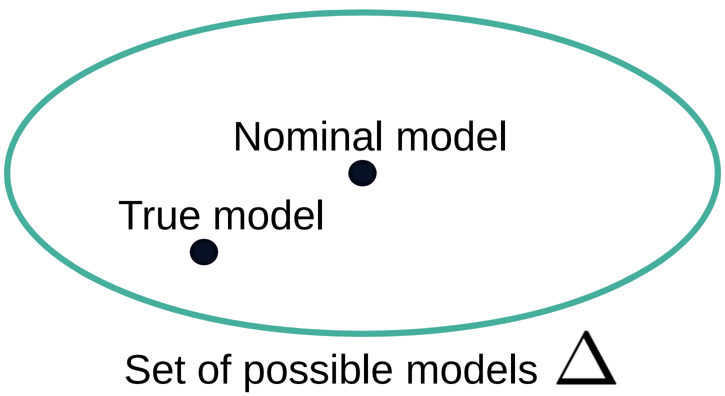
DYNAMIC SYSTEMS LAB

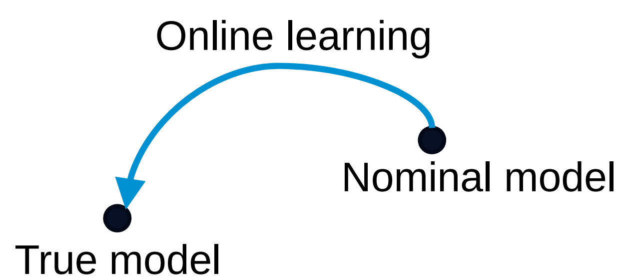
Robust control

- Specify prior uncertainty Δ in model
- Guarantee stability and performance for all possible models

Online learning

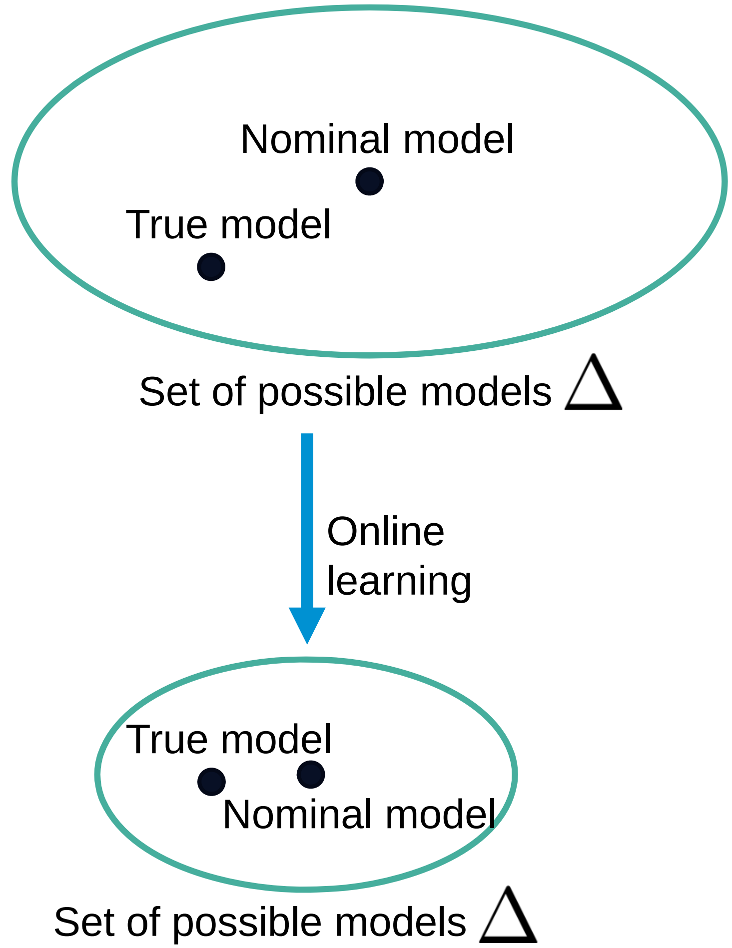
- Learn from online data
- Improve the model





DYNAMIC SYSTEMS LAB

	Robust Control	Online learning	Learning-based Robust Control
Models uncertainty	✓	✓	✓
Guarantees stability	✓	✗	✓
Improves online	✗	✓	✓



S. Schaal and C. G. Atkeson, “Learning control in robotics,”
IEEE Robotics & Automation Magazine, vol. 17, no. 2, pp. 20–29, 2010.

Model for our approach

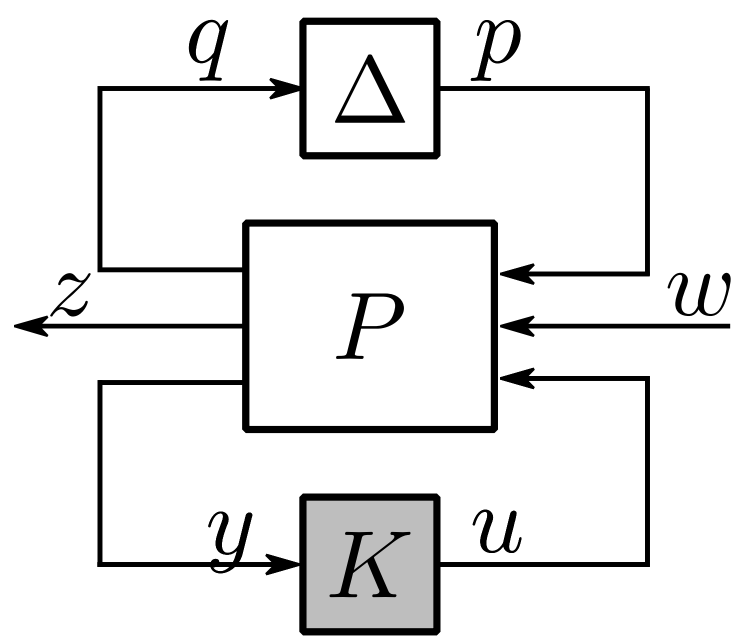
$$\mathbf{x}_{k+1} = \underbrace{\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)}_{a\ priori \text{ model}} + \underbrace{\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k)}_{\text{to be learned}}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \boldsymbol{\omega}_k,$$

- Stabilization of an operating point
- Robust Control: Guaranteed stability / performance
- Gaussian Process: Online learning

- Extended linear plant P
- Output $\mathbf{y} = \mathbf{C}\mathbf{x} = \mathbf{x}$
- Uncertainty $\Delta = \text{diag}(\delta_1, \ldots, \delta_f)$
  $|\delta_i| \leq 1$
- Disturbance $\mathbf{w}$
- Error signal $\mathbf{z}$

Robust Controller Design Framework



Goal: Find K such that

- Stability: P stable for all allowed Δ
- Performance: $\mathbf{z}$ minimized

→Convex optimization problem

Gaussian Process (GP) learning

- Objective: learn the model error with uncertainties from input/output data

$$\mathbf{x}_{k+1} = \underbrace{\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)}_{a\ priori \text{ model}} + \underbrace{\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k)}_{\text{to be learned}}$$

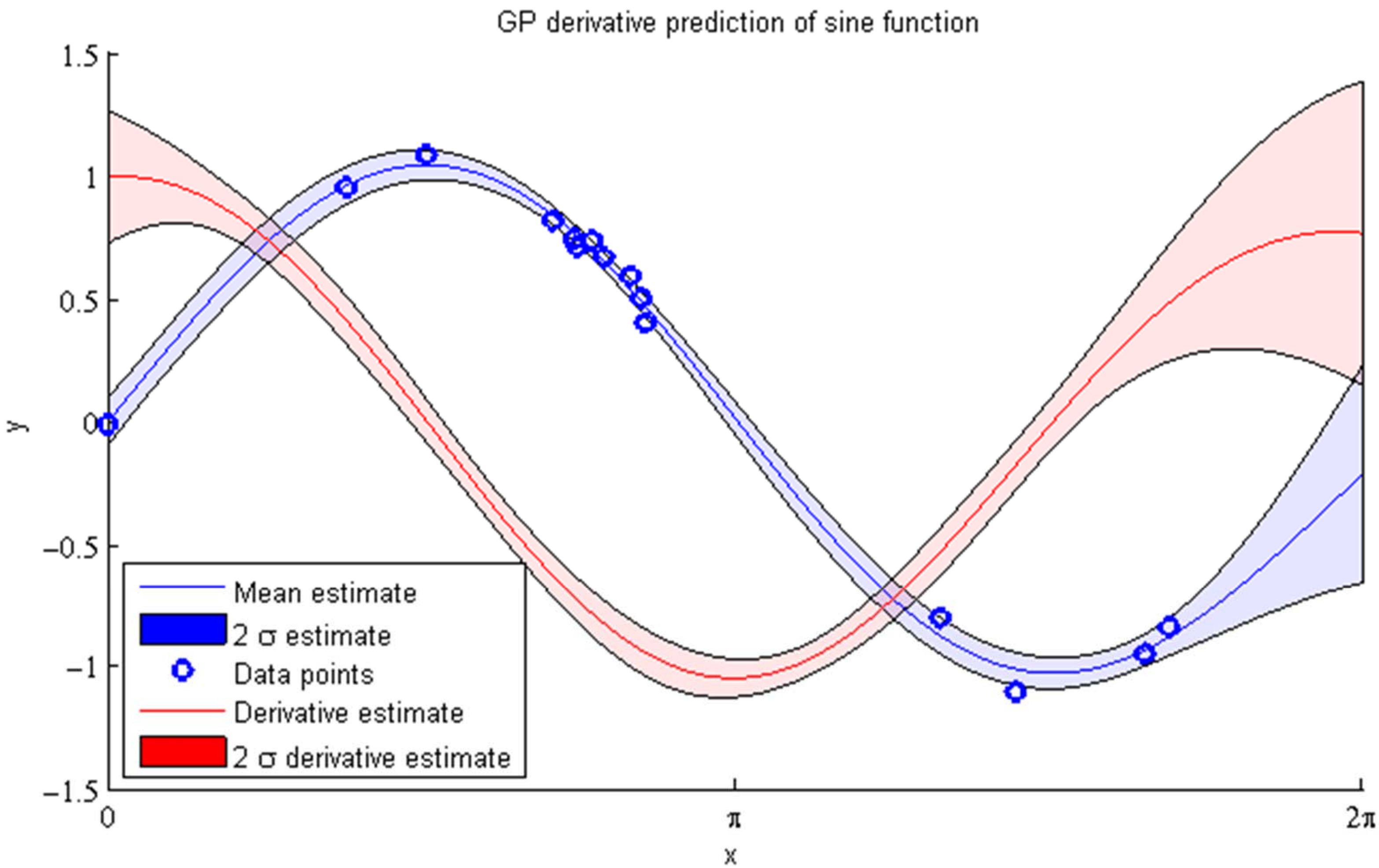
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \boldsymbol{\omega}_k ,$$

- Assumption: Correlation of data given by a kernel function

  Similar inputs will lead to similar outputs

- Hyperparameters (noise and scaling): learned from data

Gaussian Process learning



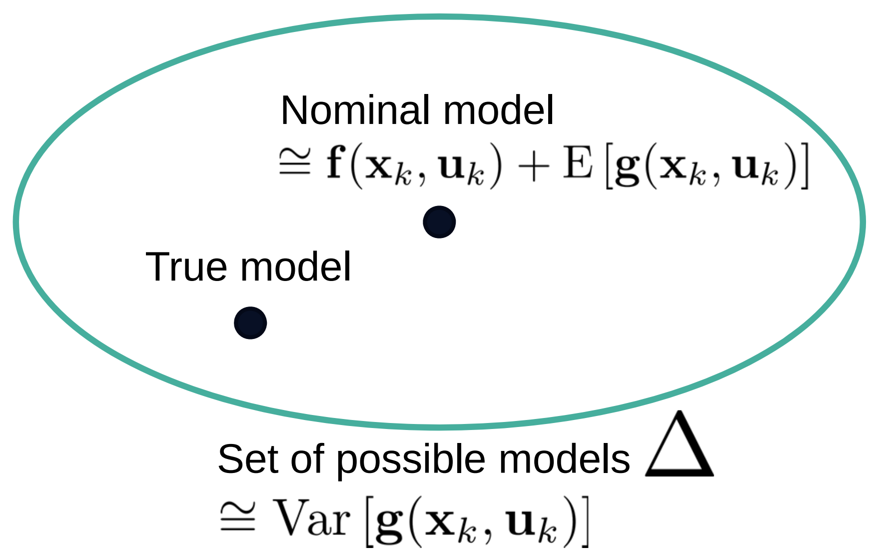
Gaussian Process in our context

- Model:

$$\mathbf{x}_{k+1} = \underbrace{\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)}_{a\ priori\ \text{model}} + \underbrace{\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k)}_{\text{to be learned}}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \boldsymbol{\omega}_k,$$

- GP inputs: $\mathbf{a}_k = (\mathbf{x}_k, \mathbf{u}_k)$
- GP outputs: $\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$



Example: simple linear/affine system

- True model:

$$\mathbf{x}_{k+1} = \underbrace{\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)}_{a\ priori\ \text{model}} + \underbrace{\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k)}_{\text{to be learned}}$$

$$x_{k+1} = ax_k + bu_k + d \qquad \begin{aligned} a &= 0.6 \\ b &= 0.9 \\ d &= 0 \end{aligned}$$

- Without prior knowledge, use GP to learn system dynamics

$$f(x_k, u_k) = 0$$

Parameters estimated using GP

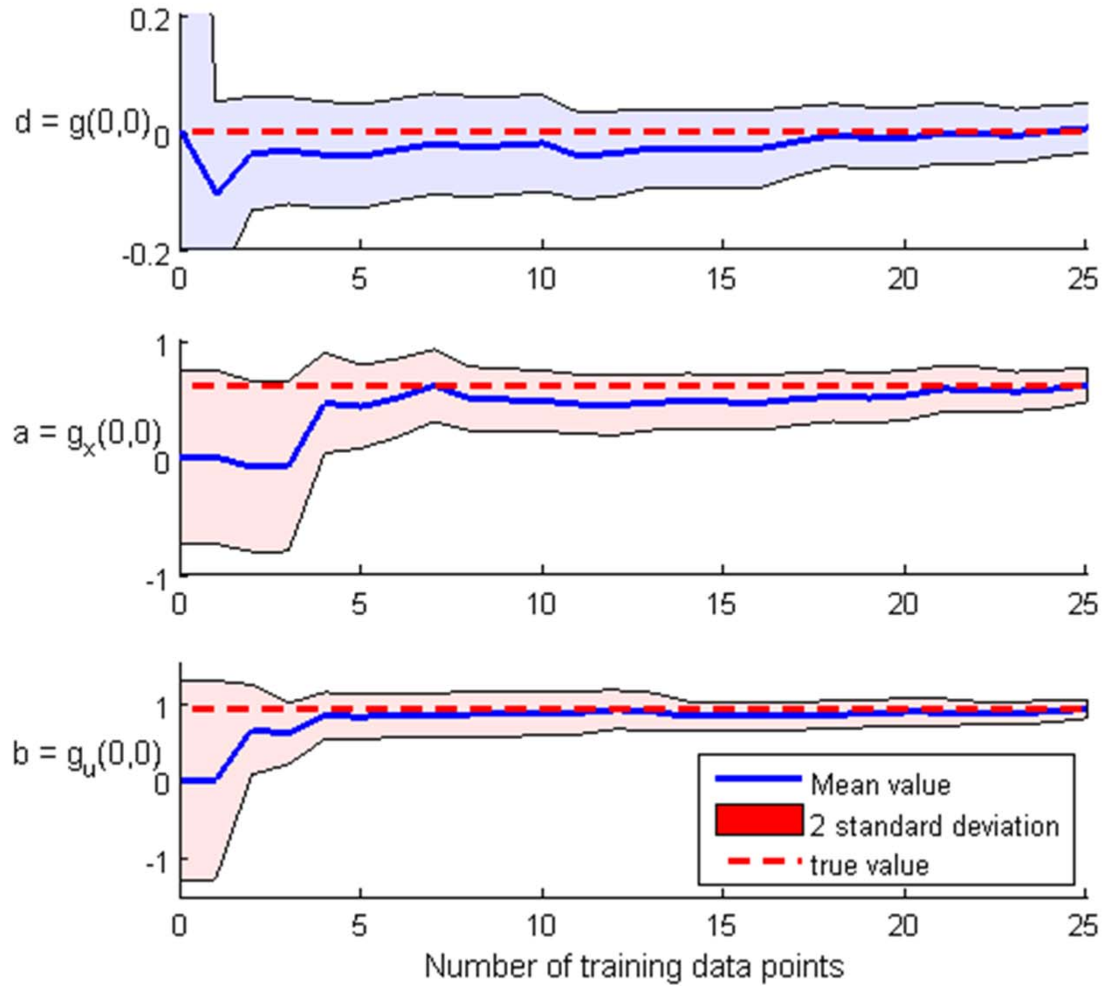
$$x_{k+1} = a x_k + b u_k + d$$

$$x_{k+1} = g(x_k, u_k)$$

$$a = 0.6$$
$$b = 0.9$$
$$d = 0$$

$$g_x = \frac{\partial g}{\partial x}, \;\; g_u = \frac{\partial g}{\partial u}$$

$$\begin{aligned} x_{k+1} = &\, (\mathrm{E}[g_x] + 2\sqrt{\mathrm{Var}[g_x]}\delta_a) x_k \\ &+ (\mathrm{E}[g_u] + 2\sqrt{\mathrm{Var}[g_u]}\delta_b) u_k \\ &+ (\mathrm{E}[g\;\,] + 2\sqrt{\mathrm{Var}[g\;\,]}\delta_d) \end{aligned}$$



$$\delta_a, \delta_b, \delta_d \in [-1, 1]$$

Bringing GPs and Robust Control together

$$x_{k+1} = \overbrace{\underbrace{\overbrace{\mathrm{E}[g_x]x_k}^{P_{nom}}}_{} + \overbrace{\overbrace{2\sqrt{\mathrm{Var}[g_x]}x_k}^{q}\ \overbrace{\delta_a}^{\Delta}}^{p}}$$

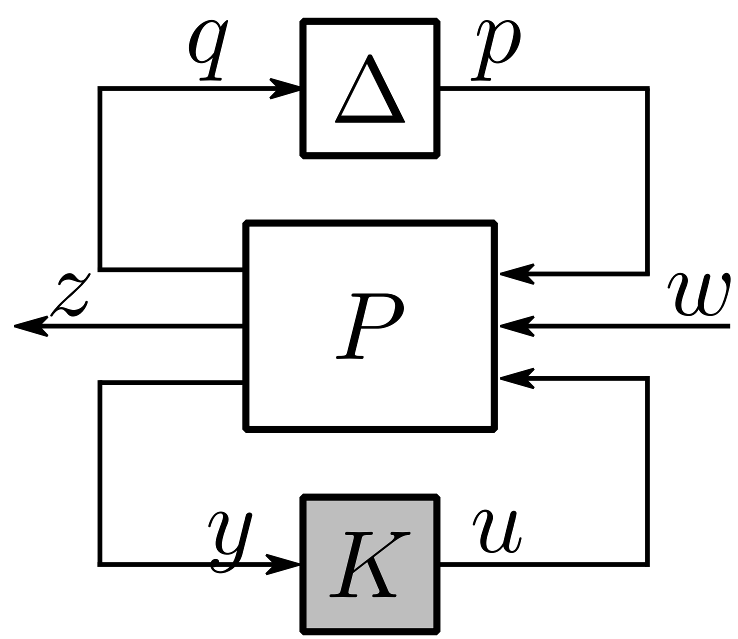
$$+ \mathrm{E}[g_u]u_k + 2\sqrt{\mathrm{Var}[g_u]}u_k \ \delta_b$$

$$+ \underbrace{\mathrm{E}[g]}_{\text{steady-state error}} + \underbrace{2\sqrt{\mathrm{Var}[g]}\,\delta_d}_{w}$$

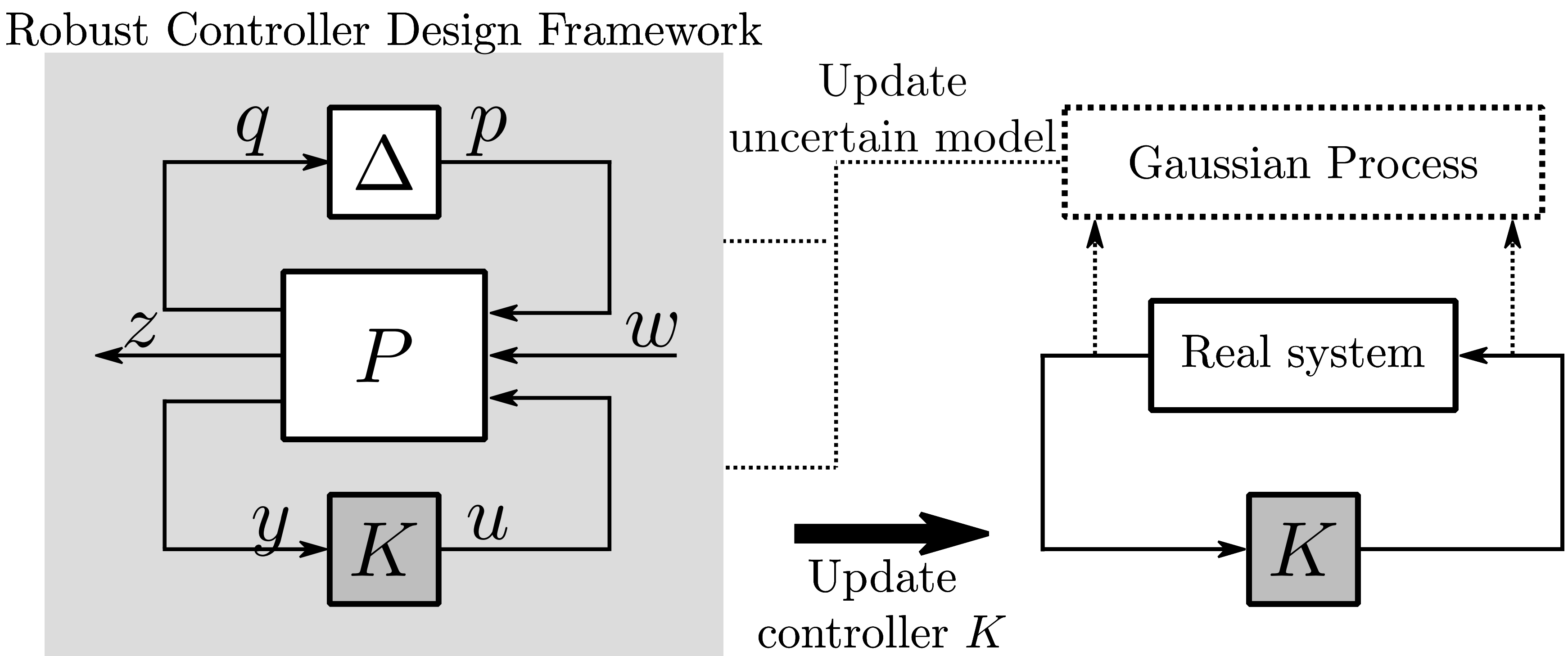
$$\delta_a, \delta_b, \delta_d \in [-1, 1]$$

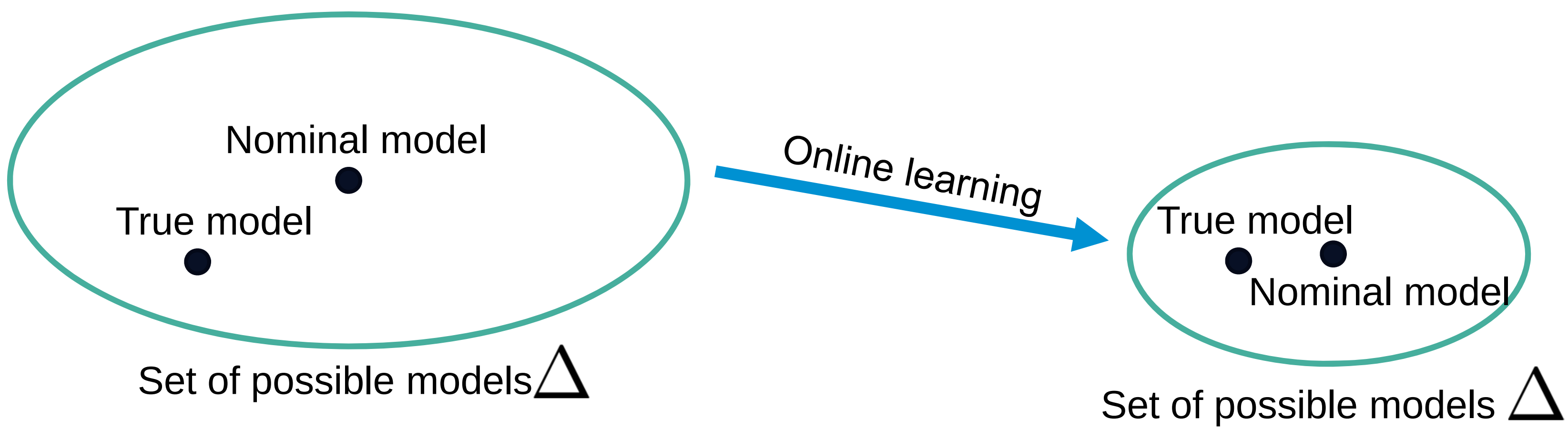
This generalizes to non-linear, non-scalar, MIMO systems using the same process for element-wise uncertainty

Robust Controller Design Framework

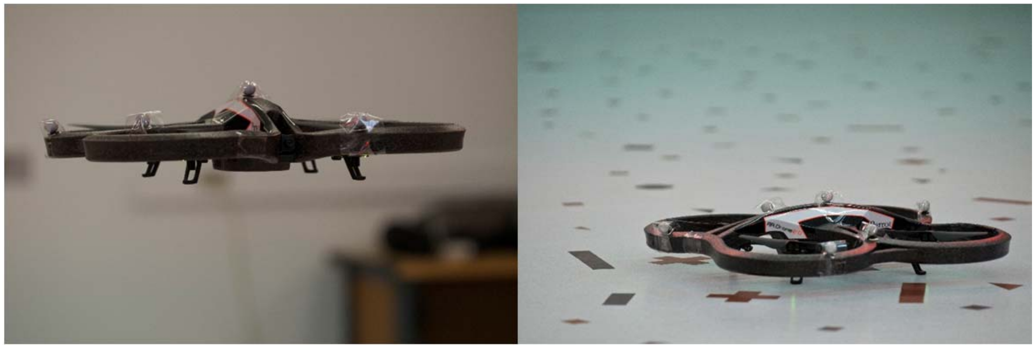


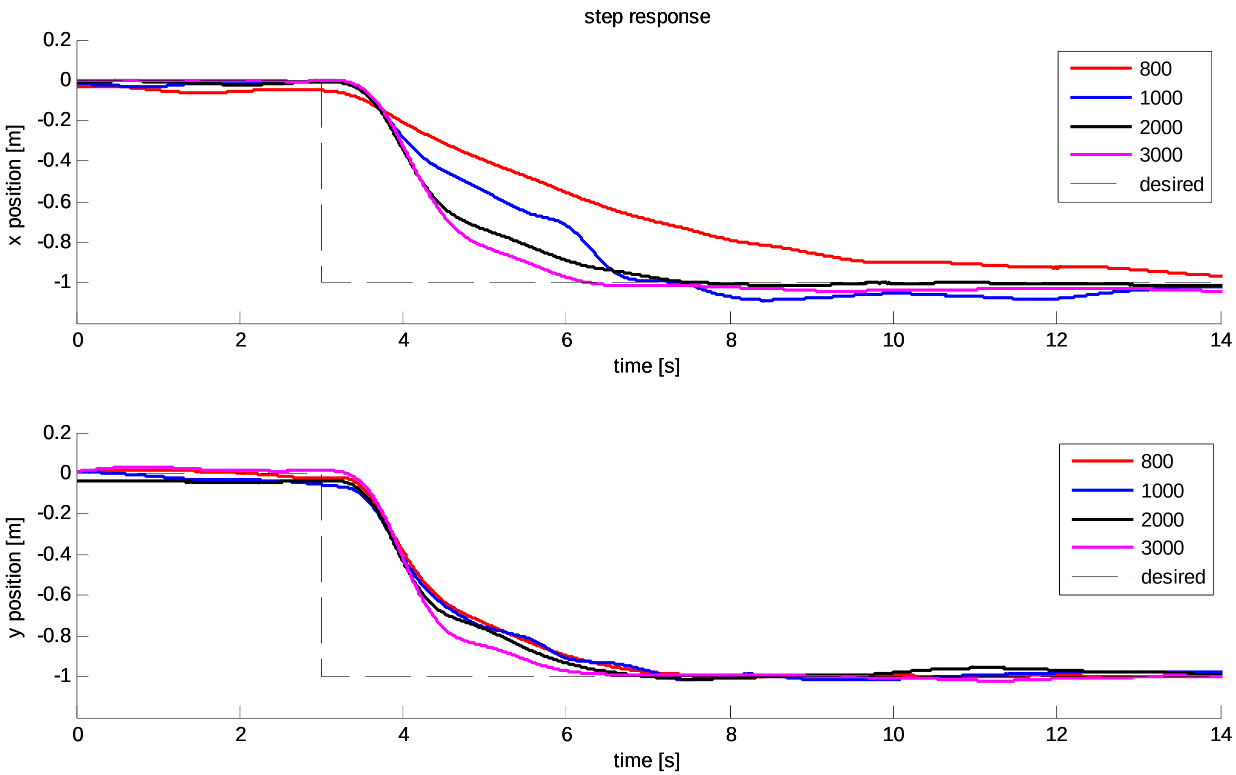
The complete process

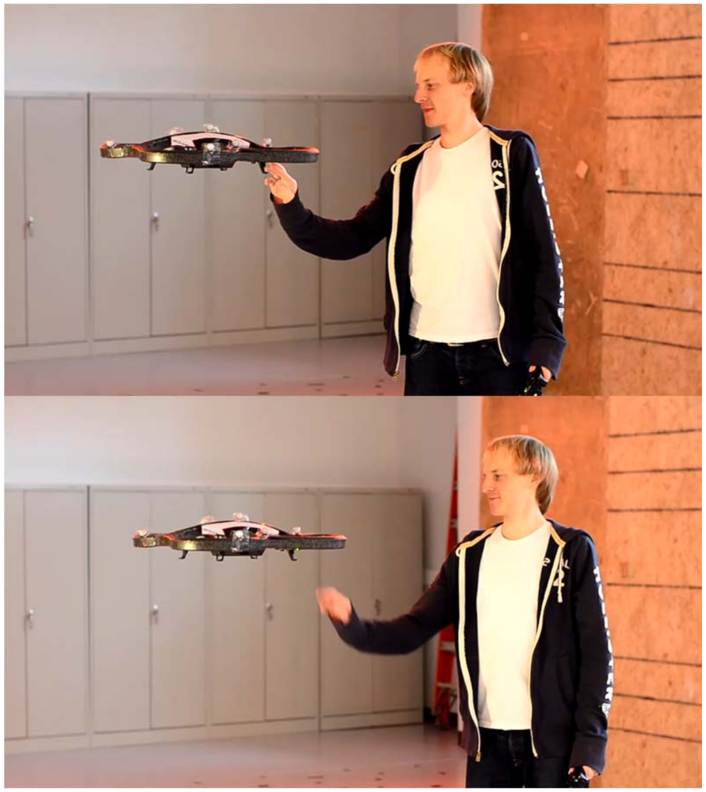




Experimental results: https://youtu.be/YqhLnCm0KXY?list=PLC12E387419CEAFF2

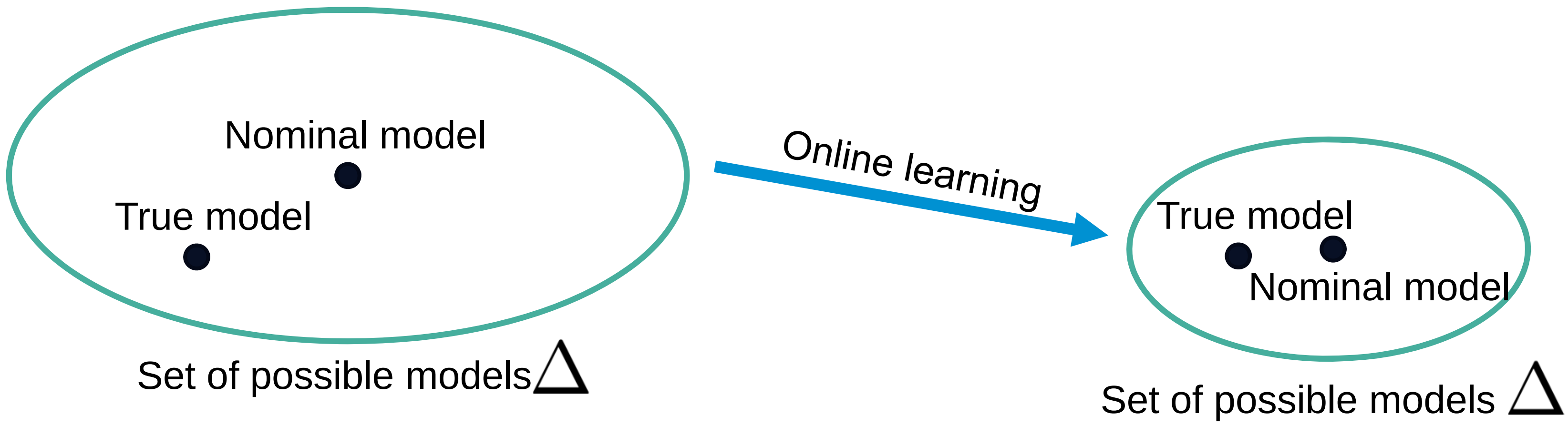






Summary

- **Combined** Gaussian Process **learning** with Linear **Robust Control**
- Enables **controller performance to improve online** while providing **stability guarantees**



Follow us!

Thank you!

Felix Berkenkamp
befelix@ethz.ch

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

DYNAMIC
SYSTEMS LAB