Safe Learning of Regions of Attraction for Uncertain, Nonlinear Systems with Gaussian Processes

Felix Berkenkamp, Riccardo Moriconi, Angela P. Schoellig, Andreas Krause

@CDC, December 2016

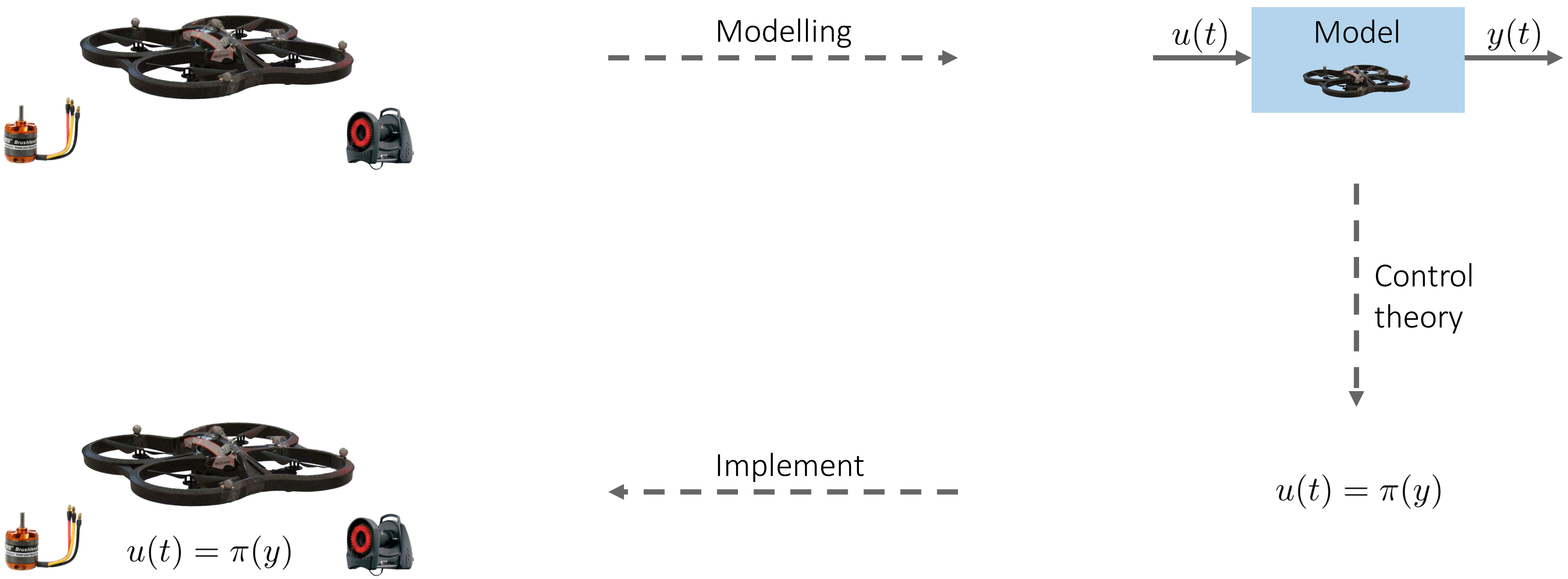
ETH zürich

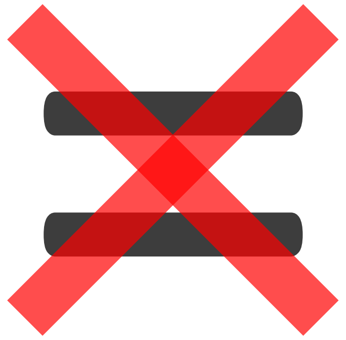
LAS

DYNAMIC SYSTEMS LAB

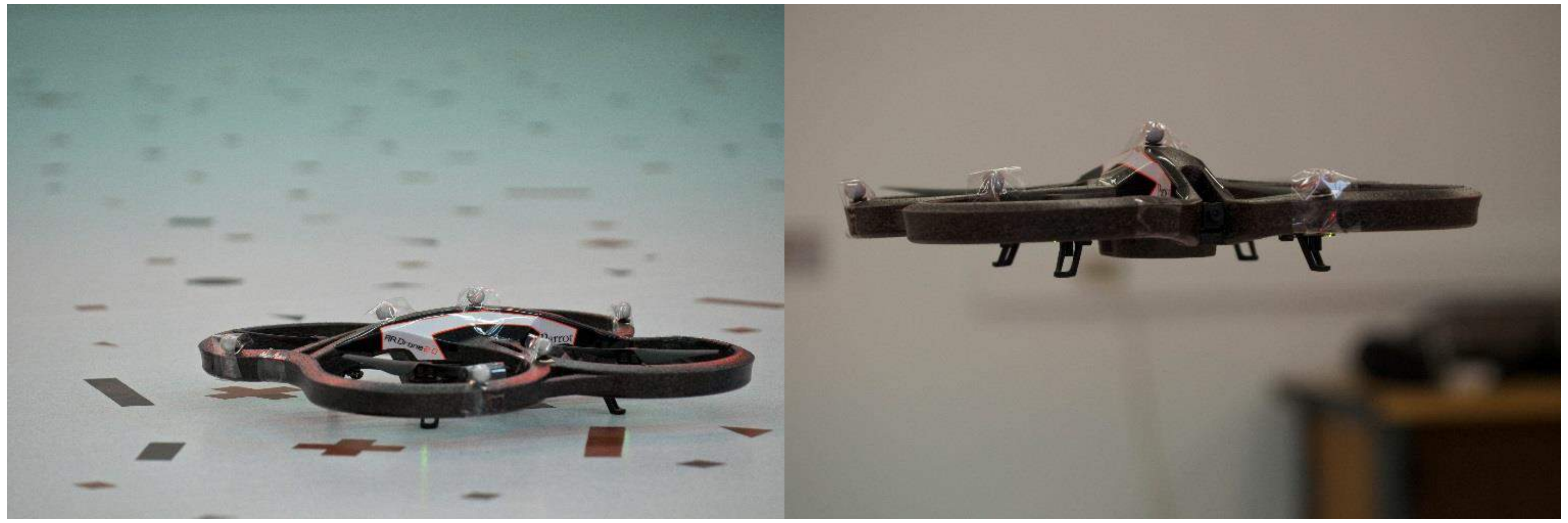
Institute for Aerospace Studies
UNIVERSITY OF TORONTO

What is control?

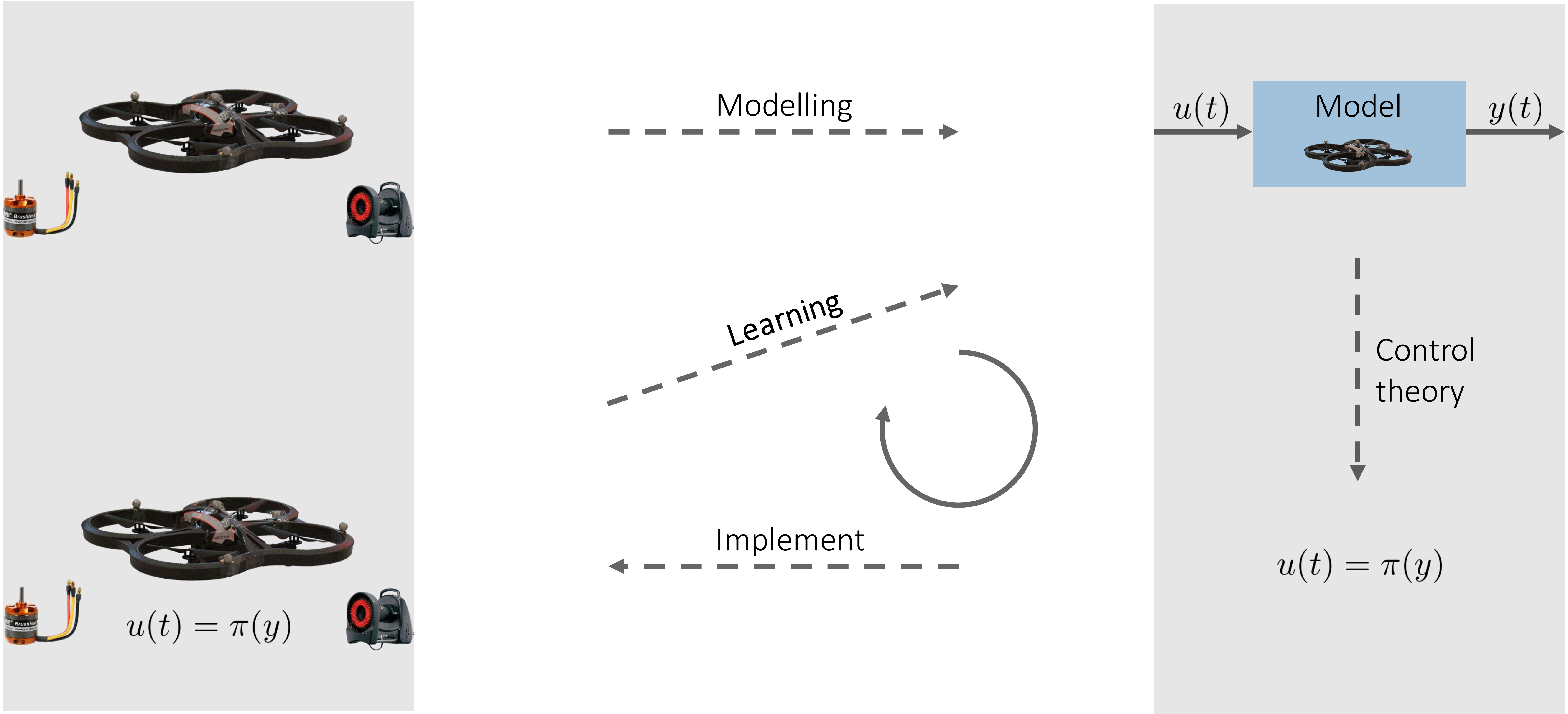




Degraded performance
Instability



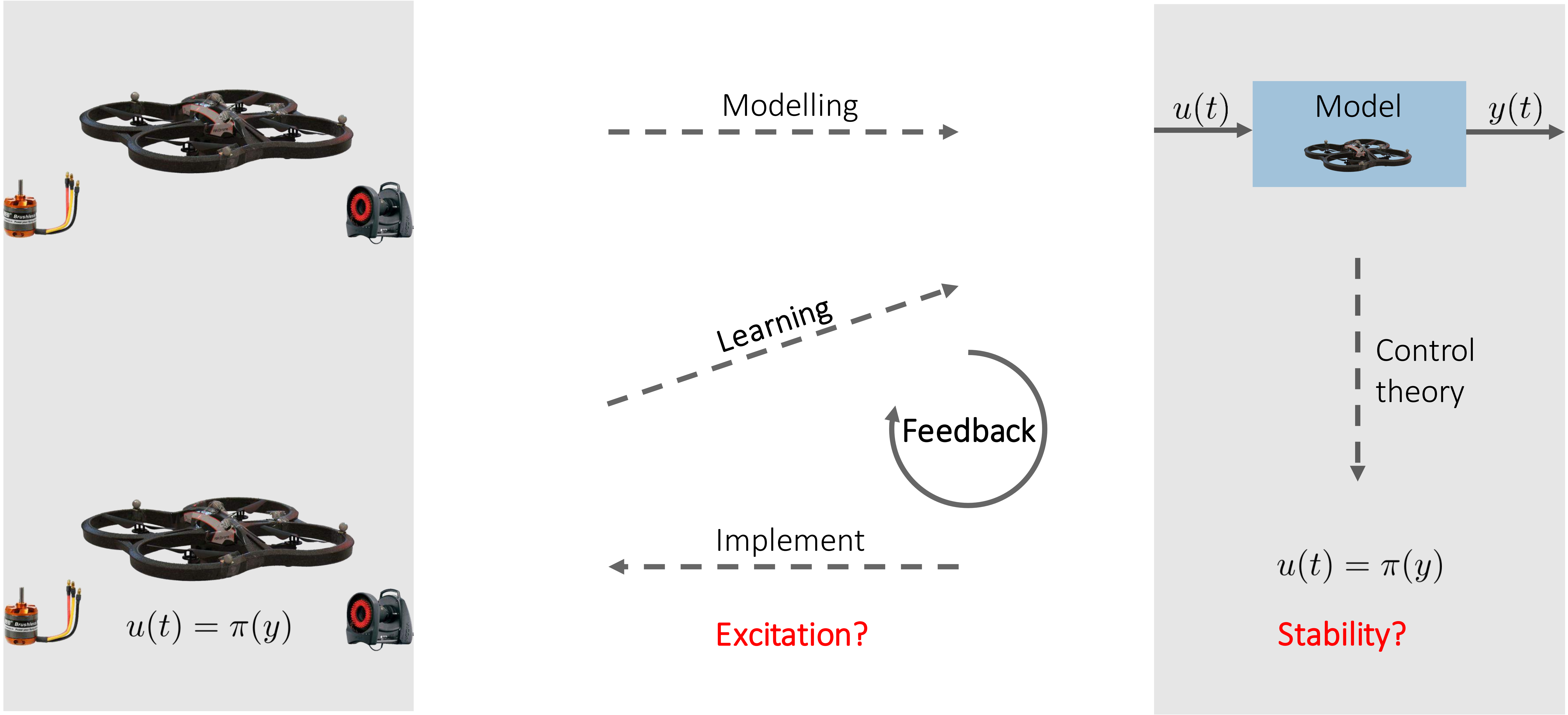
What is control?



Why is learning not commonly used?

Because safety matters!

What can go wrong?



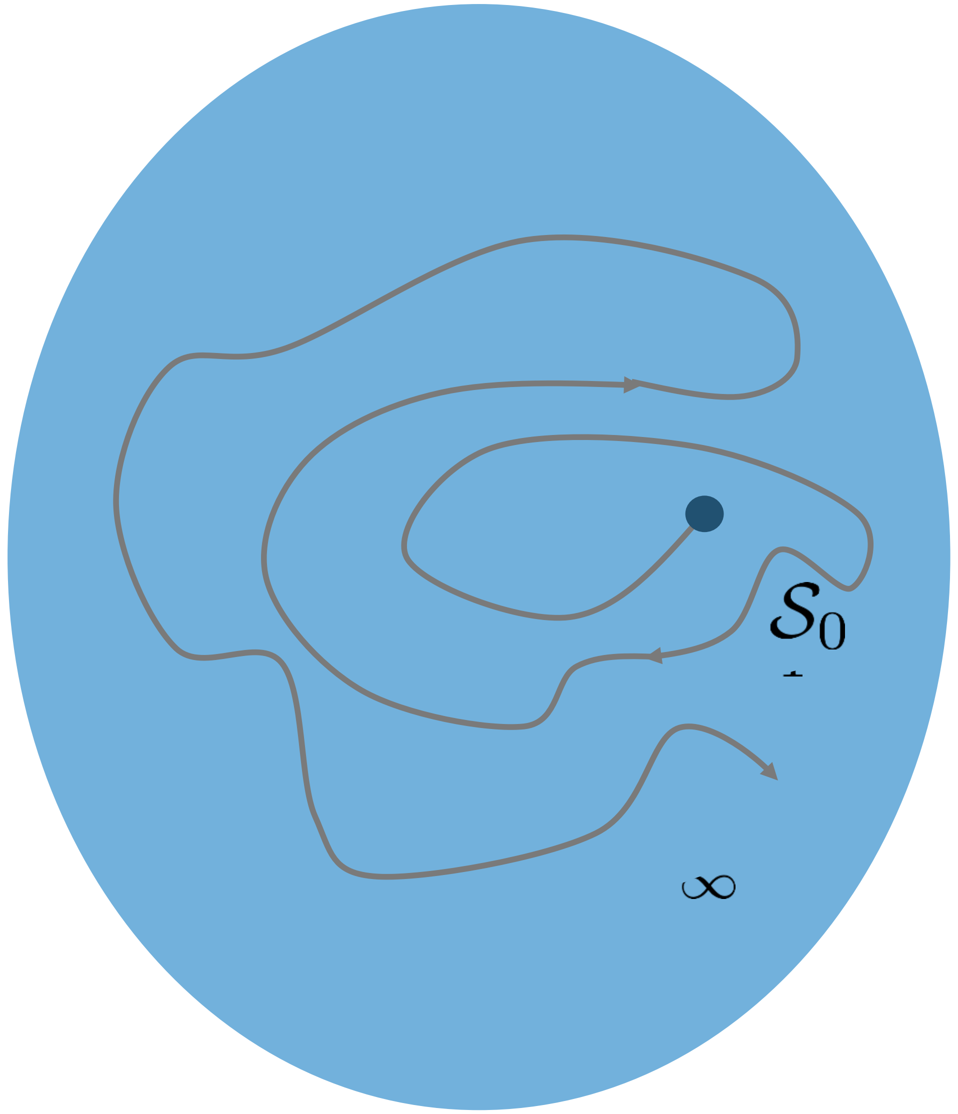
Can we learn about dynamics while remaining stable?

$$\dot{x}(t) = \underbrace{f(x(t), u(t))}_{a\ priori\ \text{model}} + \underbrace{g(x(t), u(t))}_{\text{unknown model}} \qquad \text{with } u(t) = \pi(x(t))$$

Lipschitz continuous   Bounded RKHS norm

Where is this control policy safe to use?

You can experiment, but no system failures!



Stability certificates (robustness)

✓ Linear controllers [F.Berkenkamp et al, ECC'15]

? Nonlinear systems [A.K.Akametalu et al, CDC'14]

This paper:

Lyapunov stability (nonlinear, uncertain systems) with high probability

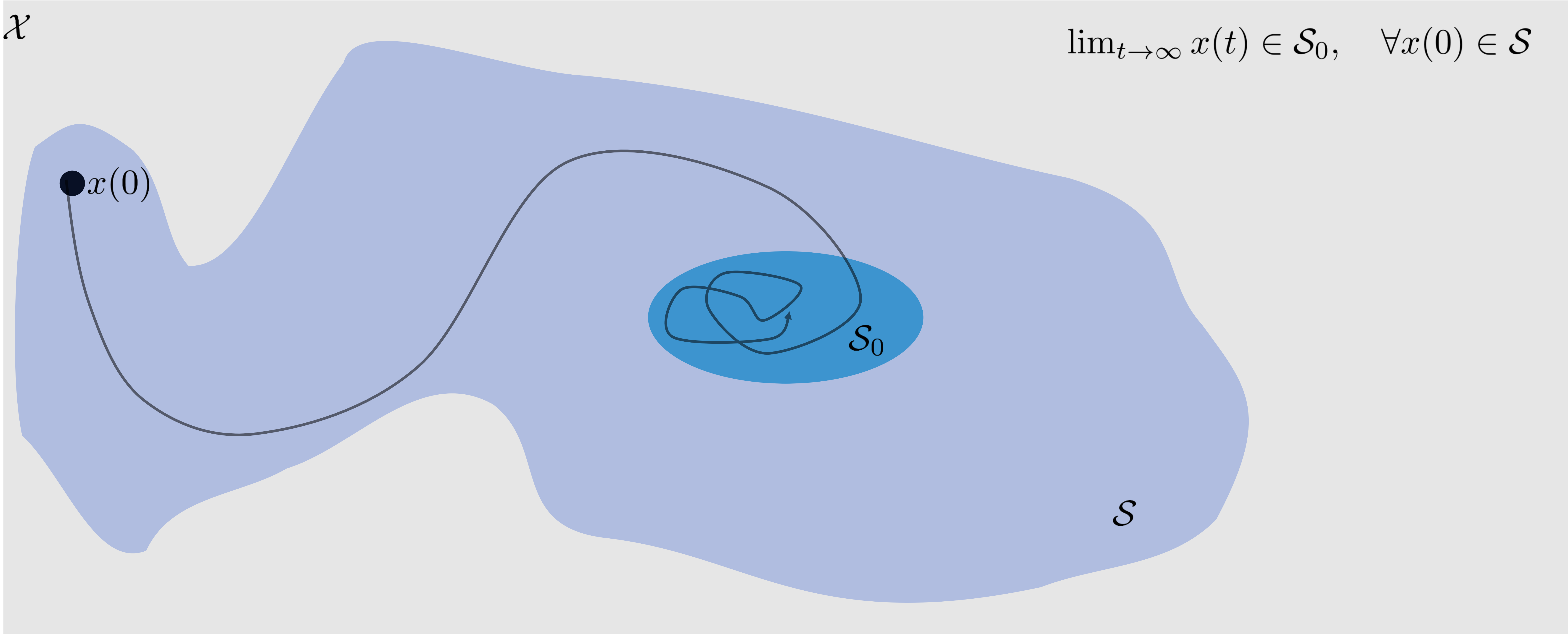
Exploration (excitation)

✓ Linear systems [L. Jung, SAP'98]

✓ Finite domains [R.I.Brafman et al, JMLR'02]

? Nonlinear, continuous

Use ideas from sensor placement


$\mathcal{X}$
$\lim_{t\to\infty} x(t) \in \mathcal{S}_0, \quad \forall x(0) \in \mathcal{S}$
$x(0)$
$\mathcal{S}_0$
$\mathcal{S}$

$\dot{x} = f_\pi(x)$ $V(x)$

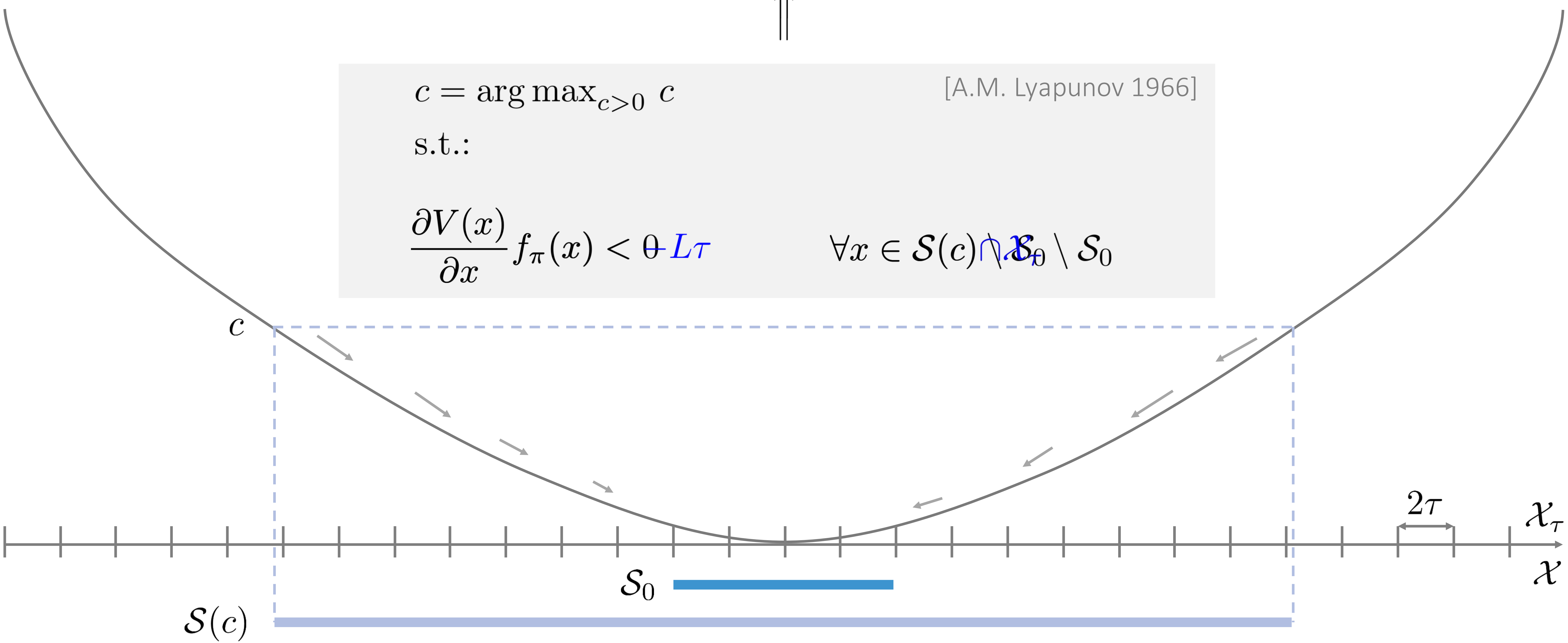
$\lim_{t\to\infty} x(t) \in \mathcal{S}_0 \qquad \forall x(0) \in \mathcal{S}(c)$

$\Uparrow$

$c = \arg\max_{c>0}\ c$ [A.M. Lyapunov 1966]

s.t.:

$\frac{\partial V(x)}{\partial x} f_\pi(x) < \not{0}\, L\tau \qquad \forall x \in \mathcal{S}(c) \cap \mathcal{X}_\tau \setminus \mathcal{S}_0$



$$\dot{x} = f_\pi(x) + \underbrace{g_\pi(x)}_{\text{unknown model}}$$

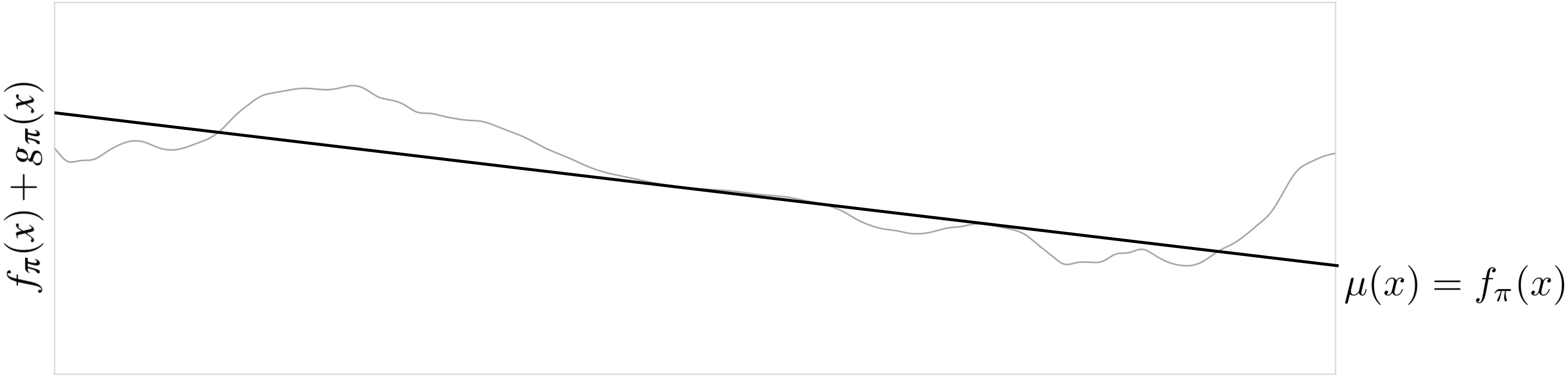
$$c = \arg\max_{c>0} \; c$$

s.t.:

$$\frac{\partial V(x)}{\partial x} f_\pi(x) < -L\tau \qquad \forall x \in \mathcal{S}(c) \cap \mathcal{X}_\tau \setminus \mathcal{S}_0$$

known systems: [R. Bobiti, M. Lazar, CDC 2016]

$$\dot{x} = f_\pi(x) + \underbrace{g_\pi(x)}_{\text{unknown model}}$$



high probability confidence intervals

Lipschitz continuous

What about unknown dynamics?

$$\dot{x} = f_\pi(x) + \underbrace{g_\pi(x)}_{\text{unknown model}}$$

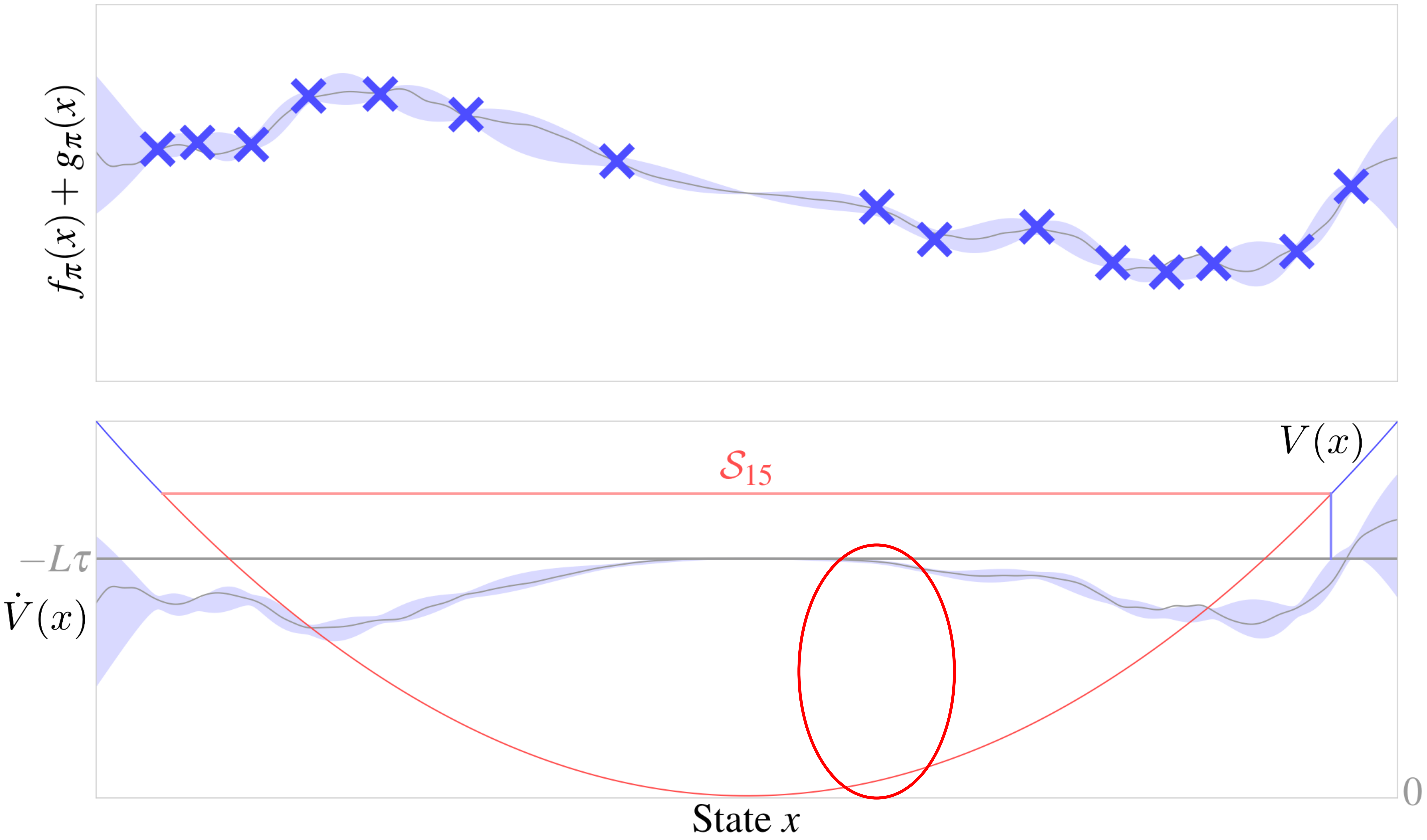
$$c = \arg\max_{c>0} \; c$$

s.t.:

$$\Pr\left\{ \frac{\partial V(x)}{\partial x} \left(f_\pi(x) + g_\pi(x) \right) < -L\tau \;\; \forall x \in \mathcal{S}(c) \cap \mathcal{X}_\tau \setminus \mathcal{S}_0 \right\} \geq 1 - \delta$$

True system is stable within $\mathcal{S}(c)$ with high probability!

Exploring the safe set



Stability certificates (robustness)

✓ Linear controllers [F.Berkenkamp et al, ECC'15]

? Nonlinear systems [A.K.Akametalu et al, CDC'14]

This paper:

Lyapunov stability (nonlinear, uncertain systems) with high probability

Exploration (excitation)

✓ Linear systems [L. Jung, SAP'98]

✓ Finite domains [R.I.Brafman et al, JMLR'02]

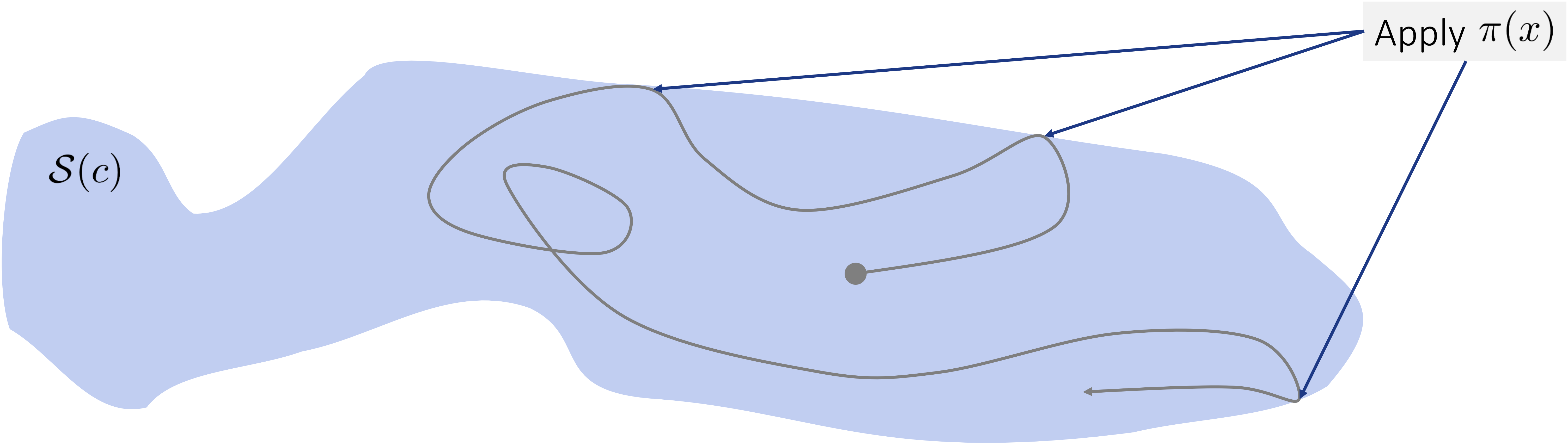
? Nonlinear, continuous

Use ideas from sensor placement

How to actively explore?

Do we converge to maximum safe set?

The policy $\pi(x)$ is safe: keeps us in $\mathcal{S}(c)$



Close-to-optimal measurements: [A.Krause, C.Guestrin, UAI'05]

$$x_n = \underset{x \in \mathcal{S}(c_n)}{\arg\max}\ \sigma_{n-1}(x)$$

Theorem: Guaranteed to *converge* to the maximum safe levelset up to a certain *accuracy* after a *finite* number of data points – *without leaving* this safe levelset with high probability.

Bound depends on

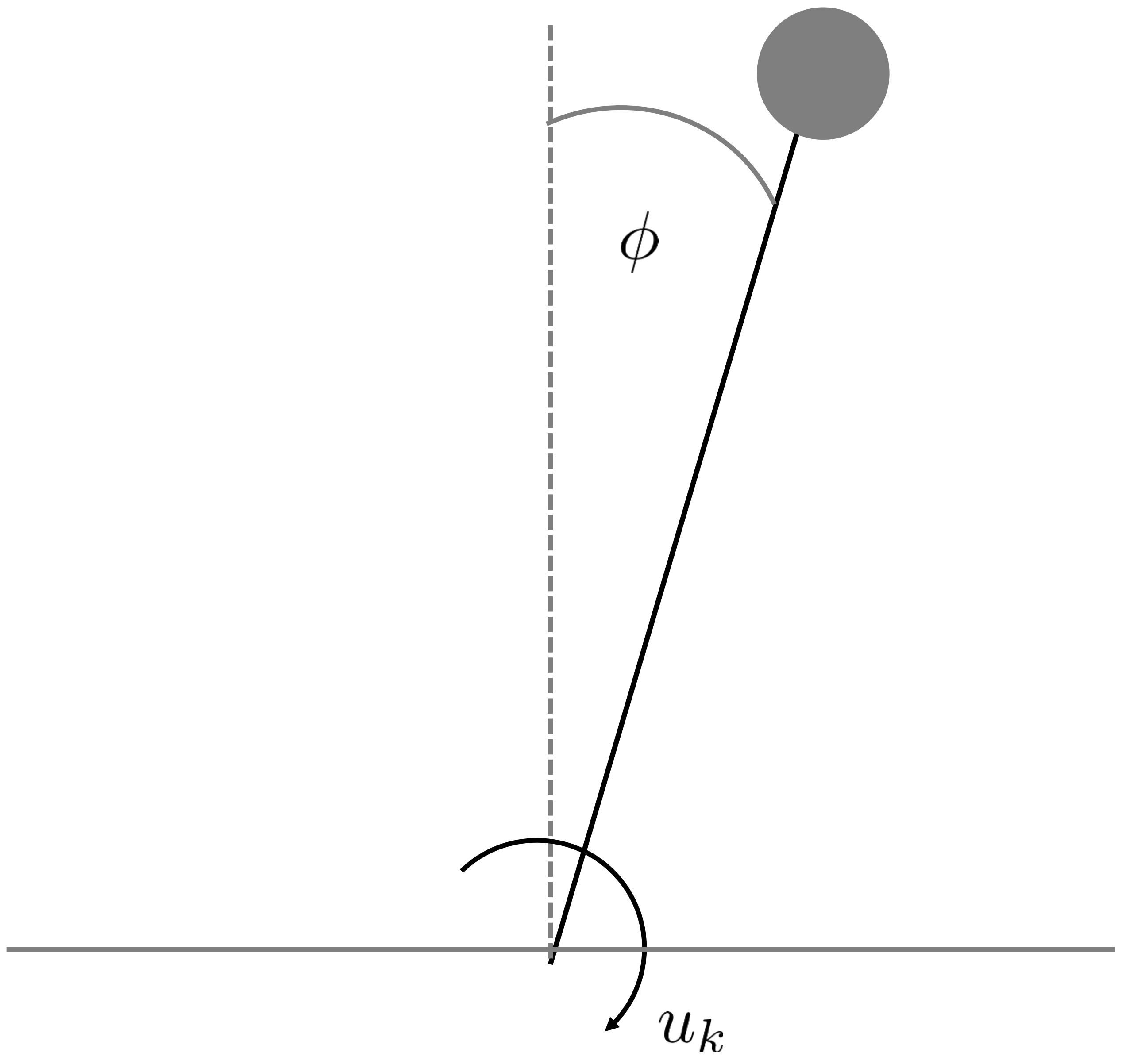
- Size of the maximum safe levelset
- Information capacity of the Gaussian process model
- Accuracy

Maximum torque limited!

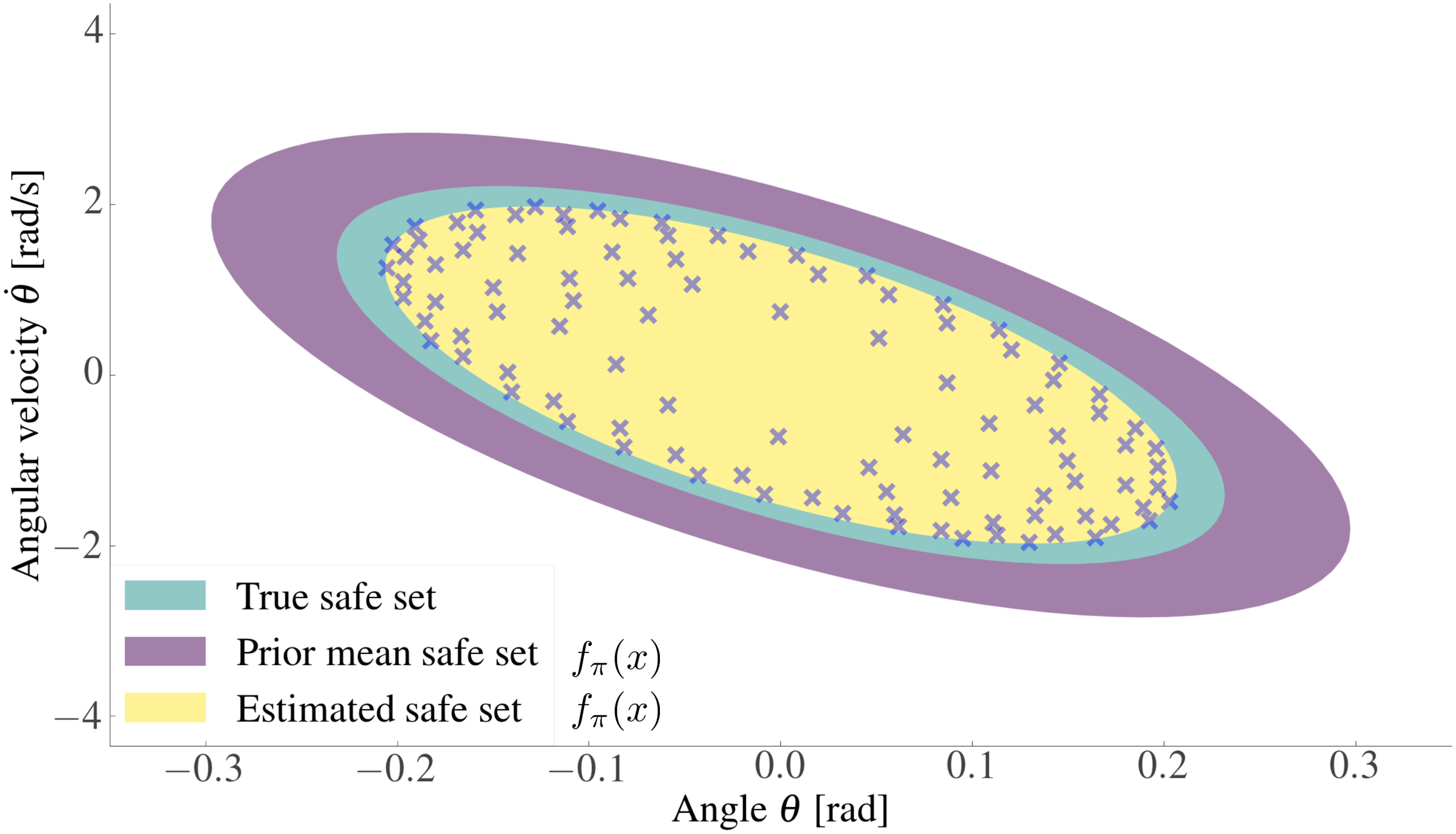
Safe exploration so that the pendulum doesn't fall.

Controller: LQR with prior mean model

Quadratic Lyapunov function



Safe learning for an inverted pendulum



Conclusion

Can **simultaneously learn** system dynamics and give **stability guarantees**

Lyapunov stability for **nonlinear, uncertain** systems (with high probability, discretization)

Convergence guarantees

There is hope for **safe reinforcement learning**!

Code is open source

Example notebooks

More safe learning at http://berkenkamp.me