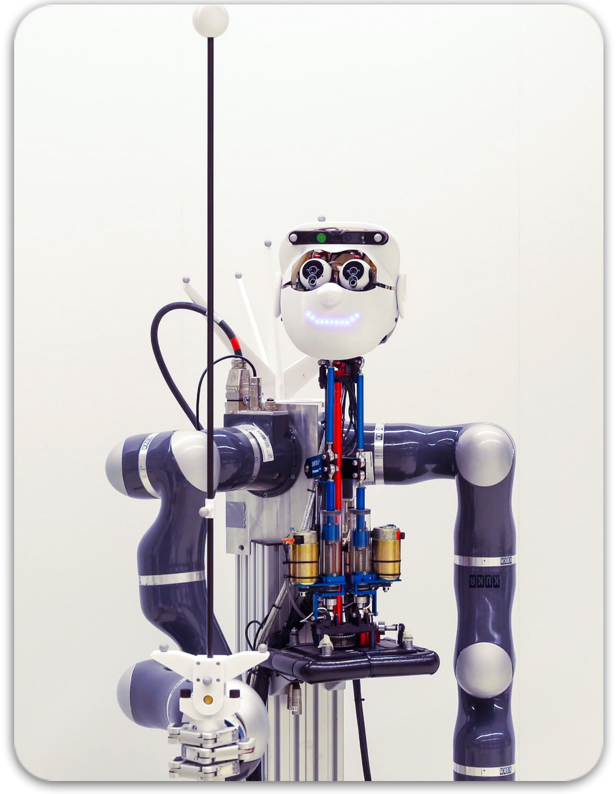
Learning Robot Controllers using Bayesian Optimization

Alonso Marco, J. Miguel Hernández-Lobato, Philipp Hennig and Sebastian Trimpe

LQR kernels for efficient Controller Learning [1]

- **Motivation**
  Tuning robot controllers *manually* is tedious, and time consuming
- **Goal**
  Learn *efficiently* LQR controller parameters from data
- **Inconveniences**
  Standard kernels agnostic to the learning problem
- **Our approach**
  Choose the correct prior for each control problem. Merge LQR controller structure into kernel: *LQR Kernel*



Automatic LQR tuning [3]

Unknown system $(\hat{a}, \hat{b})$

$$x_{t+1} = \hat{a}x_t + \hat{b}u_t + v_t, \; v_t \sim \mathcal{N}(0, v)$$

Cost $J(f) = \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=0}^{T-1} qx_t^2 + ru_t^2\right]$

Available model (a, b)

Feedback controller $u_t = fx_t$

LQR is suboptimal $f = \mathrm{lqr}(a, b, q, r)$

Goal: Find optimal f by collecting data from the unknown system. Include problem structure in kernel

LQR kernel construction

Cost: closed-form given model

$$J(f) = v\frac{(q + rf^2)}{1 - (a + bf)^2} := \phi_{(a,b)}(f)$$

Stochastic cost

$$J_{\mathrm{LQR}}(f) = w\,\phi_{(a,b)}(f), \quad w \sim \mathcal{N}(0, \sigma_w^2)$$

Parametric LQR kernel

$$k_{\mathrm{LQR}}(f, f') = \sigma_w^2\phi_{(a,b)}(f)\phi_{(a,b)}(f') = \\ = \sigma_w^2\frac{v^2(q + rf^2)(q + rf'^2)}{(1 - (a + bf)^2)(1 - (a + bf')^2)}$$

Parametric LQR kernel with m features

$$k_{\mathrm{pLQR,m}}(f, f') = \Phi^\top(f)\Sigma_w\Phi(f')$$

$$w \in \mathbb{R}^m, w \sim \mathcal{N}(0, \Sigma_w) \quad [\Phi(f)]_i = \phi_{(a_i,b_i)}(f)$$

Kernel trick

$$\Sigma_w = \sigma_w^2 I, \;\; \sigma_w \propto 1/m \\ m \to \infty$$

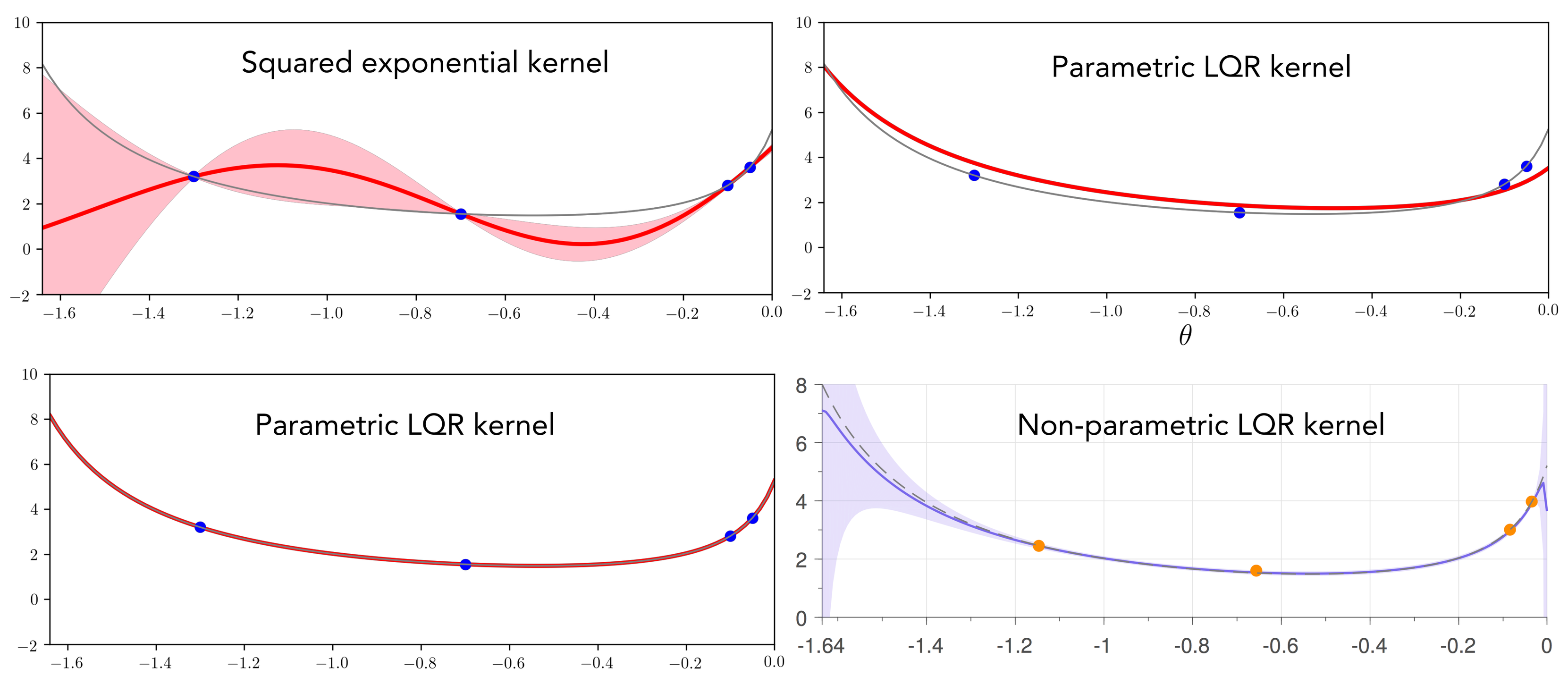
Non-parametric LQR kernel

$$k_{\mathrm{nLQR}}(f, f') = \\ = \sigma_{\mathrm{n}}^2\int_{b_{\min}}^{b_{\max}}\int_{a_{\min}}^{a_{\max}} \phi_{(a,b)}(f)\phi_{(a,b)}(f')\,da\,db$$

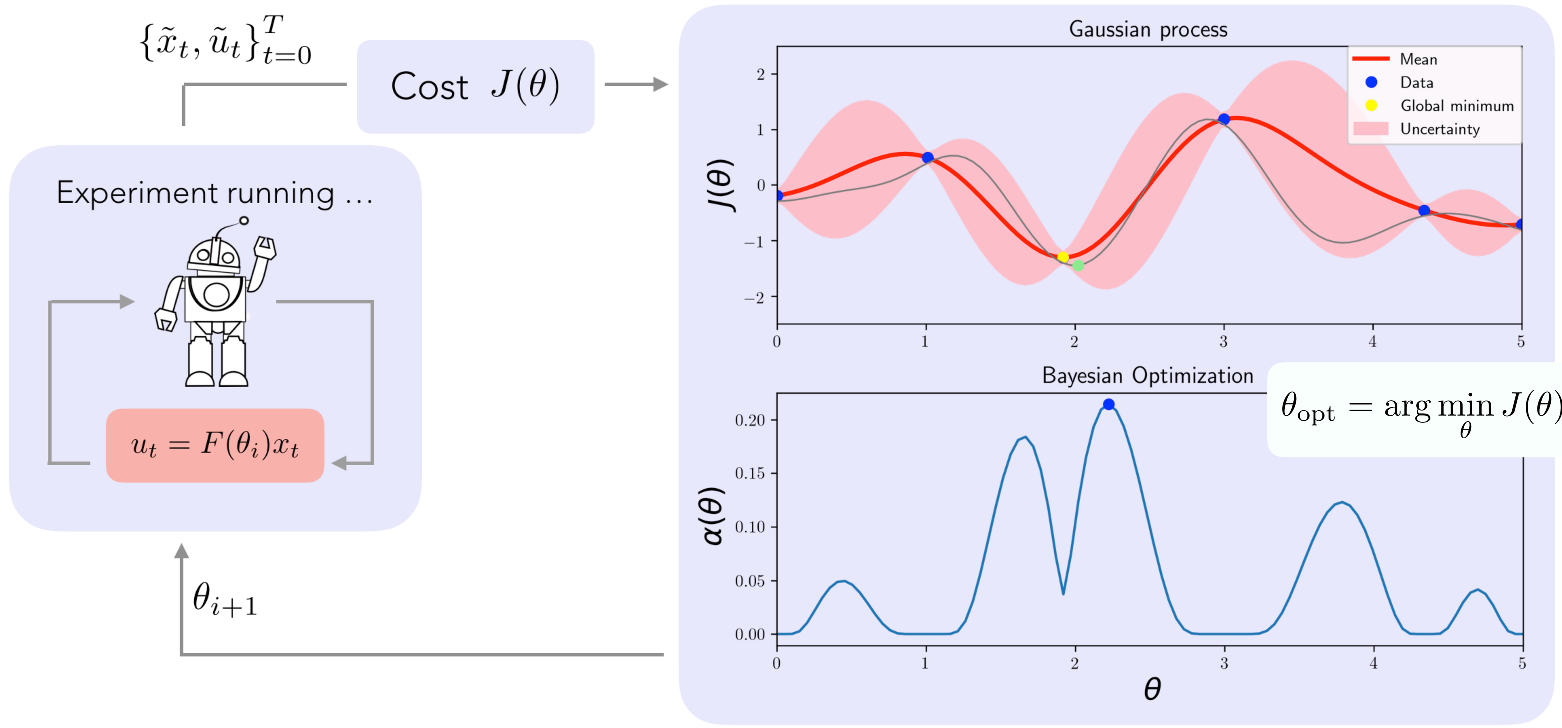
Example

True system $x_{t+1} = 0.9x_t + u_t + v_t, \; v_t \sim \mathcal{N}(0, 1)$

Available model $(a = 0.9, b = 1)$ **Available model !** $(a = 0.8, b = 0.9)$



Goal: Controller learning [3]



Control loop

x_t :	states
u_t :	control input
F :	feedback controller
θ :	parameters

Gaussian process

$$J(\theta) \sim \mathcal{GP}(m(\theta), k(\theta, \theta'))$$

$m(\theta)$:	mean
$k(\theta, \theta')$:	kernel

Bayesian Optimization

- Guides exploration towards informative areas to learn faster the global optimum

Bounded unsafety

Motivation

- Safe Bayesian optimization [6] allows no failures at all: conservative
- Bayesian optimization with constraints (BOC) [5] avoids unsafe regions, but can fail arbitrarily many times
- **Goal**: adaptive strategy under a limited budget of failures with bounded regret

Approach

- At each iteration t, approximate a batch of representative locations that satisfies the budget constraint and maximizes an improvement

$$\max_{x_{t+1},\cdots,x_N} R(x_{t+1}, \cdots, x_N)$$

$$\text{s.t. } \Pr\left[\sum_{j=t+1}^{N}(1 - \xi_j) \le \Delta M_t\right] \ge 1 - \delta_t$$

- Use DPPs with $\hat{N} >> N$ for resolution

$$\Pr\left[\sum_{j=t+1}^{N}(1 - \xi_j) \le \Delta M_t\right] = \Pr\left[\sum_{j=t+1}^{\hat{N}}(1 - \xi_j) \le \frac{\hat{N} - t}{N - t}\Delta M_t\right]$$

Problem formulation

$$f_{\mathrm{cm}} = \min_{x\in\mathcal{X}} f(x) \\ \text{s.t. } g(x) \le 0 \\ \text{under trials } \sum_{t=0}^{N}(1 - \xi_t) \le M$$

M : budget of failures
N : max nr. evaluations

$\xi_t = 1_{\{g(x_t)\le 0\}}$ → evaluation

$\Delta M_t = M - \sum_{j=1}^{t}(1 - \hat{\xi}_j)$: remaining budget

Expected improvement over the batch

$$R(x_1, \cdots, x_N) = \\ = \mathbb{E}\left[\max\{\eta_t - (f(x_{t+1}), \cdots, f(x_N)), 0\}\prod_{j=t+1}^{N} 1_{\{g(x_j)\le 0\}}\right]$$

$$F_t = \sum_{j=t+1}^{N}(1 - \xi_j) \sim \text{Poisson-Binomial, non i.i.d} \\ \mathbb{E}[\xi_j] = \mu_j, \;\; \mathrm{Cov}[\xi_i, \xi_j] = \Sigma_{ij}$$

$$\mathbb{E}[F_t] = \sum_{j}^{N}\mu_j, \;\; \mathrm{Var}[F_t] = \sum_{i,j}^{N}\Sigma_{ij}$$

References

[1] Marco, Alonso, Philipp Hennig, Stefan Schaal, and Sebastian Trimpe. "On the design of LQR kernels for efficient controller learning." *Annual Conference on Decision and Control (CDC)*, pp. 5193-5200, 2017

[2] Cunningham, John P., Philipp Hennig, and Simon Lacoste-Julien. "Gaussian probabilities and expectation propagation." *arXiv: 1111.6832*, 2011

[3] Marco, Alonso, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. "Automatic LQR tuning based on Gaussian process global optimization." *International conference on robotics and automation (ICRA)*, pp. 270-277, 2016

[4] Wang, Zi, and Stefanie Jegelka. "Max-value entropy search for efficient Bayesian optimization." *International Conference on Machine Learning, vol. 70*, pp. 3627-3635, 2017

[5] Gelbart, Michael A., Jasper Snoek, and Ryan P. Adams. "Bayesian optimization with unknown constraints." *arXiv:1403.5607*, 2014)

[6] Sui, Yanan, Alkis Gotovos, Joel Burdick, and Andreas Krause. "Safe exploration for optimization with Gaussian processes." In *International Conference on Machine Learning*, pp. 997-1005, 2015

Gaussian process for classified regression (GPCR)

Motivation

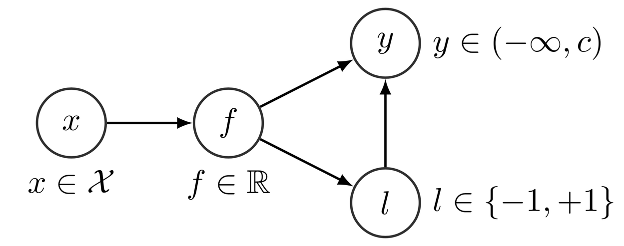
- System **failure** -> Press the button!
- Collected data is **scarce** due to premature experiment detention
- Resulting cost orders of magnitude higher
- **Any penalty number is arbitrary**

Goal

- Bayesian model for the cost captures exactly what we know about unstable controllers: *Nothing*
- It models a large **unknown** number

Observation model

	Observation y_i	Label l_i
Stable	$f_i(x_i) + \varepsilon$	$+1$
Failure	$\emptyset$	-1



- Hybrid data set $\mathcal{D} = \underbrace{\{y_i, l_i\}_{i=1}^{N_s}}_{\text{Stable eval.}} \bigcup \underbrace{\{l_i\}_{i=1}^{N_u}}_{\text{Unstable eval.}}$
- Observations predicted below stability threshold $y(x) < c$

$$p(y_i, l_i | f_i, x_i) = H(l_i(c - f_i)) \times \\ \left(1_{\{l_i=+1\}}\mathcal{N}\left(y_i; f_i, \sigma^2\right) + 1_{\{l_i=-1\}}\right)$$

c can be estimated from data!

Problem formulation

$$f_{\mathrm{cm}} = \min_{x\in\mathcal{X}} f(x) \\ \text{s.t. } g(x) \le c_g$$

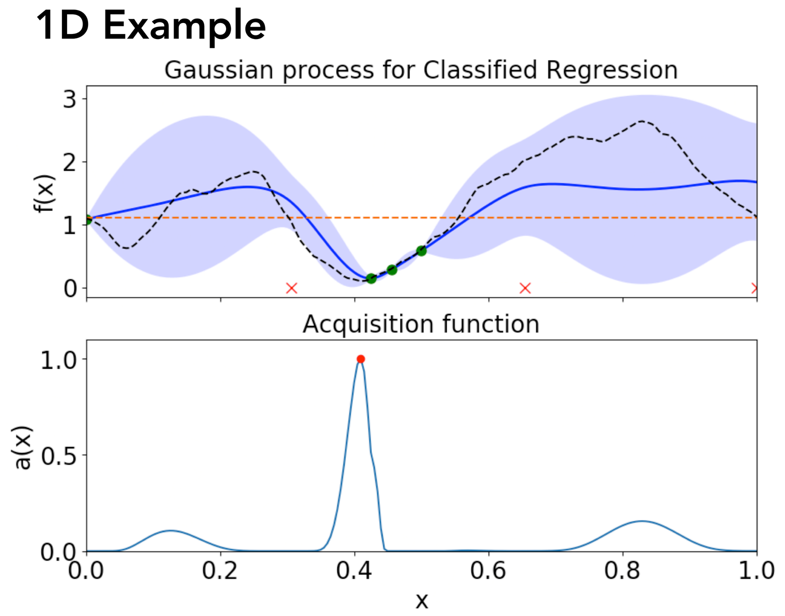
Unknown constraint *absorbed* by the GPCR model

Posterior: Unnormalized Gaussian with support over unbounded hyper-rectangle

$$p(\boldsymbol{f}|\mathcal{D}, X) \propto \prod_{i=0}^{N_u} H(f_i - c)\prod_{i=0}^{N_s} H(c - f_i)\mathcal{N}(\boldsymbol{y}_s|\boldsymbol{f}_s, \sigma^2\boldsymbol{I})\mathcal{N}\left(\begin{bmatrix}\boldsymbol{f}_s \\ \boldsymbol{f}_u\end{bmatrix} \middle| \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{0}\end{bmatrix}, \begin{bmatrix}K_{ss} & K_{su} \\ K_{us} & K_{uu}\end{bmatrix}\right)$$

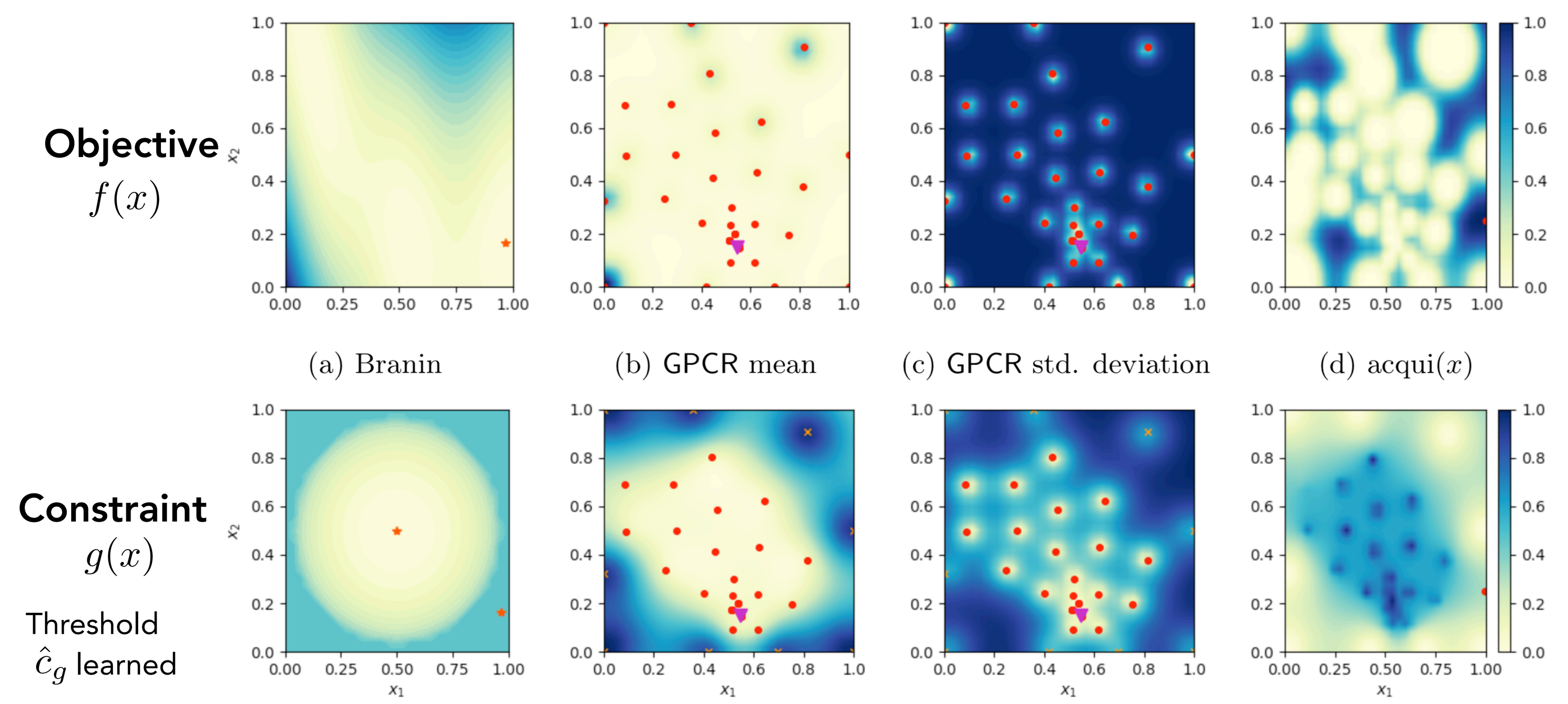
Predictive: Gaussian approx. [2] $\quad p(f_*|\mathcal{D}, X, x_*) \simeq q(f_*) = \mathcal{N}\left(f_*; \mu(x_*|\mathcal{D}), \sigma^2(x_*|\mathcal{D})\right)$

1D Example



Results

- Extended Max-Value Entropy Search [4] for constraints: *mESCO*
- Learn a 5D controller parametrization on the robot Apollo without specifying the penalty
- The same model can be used to model a constraint with **unknown threshold**!



(a) Branin (b) GPCR mean (c) GPCR std. deviation (d) acqui(x)

(e) Circle (f) GPCR mean (g) GPCR std. deviation (h) $\Pr g(x) \le \hat{c}$