Previously on MLRG
Closed loop control

action $u$ \hspace{2cm} \text{state } x

System

Controller
$x' = Ax + Bu$

Physics

State  Action

LQR
LQR

\[ x = \begin{bmatrix} \theta \\ \dot{\theta} \\ \text{pos} \\ \text{speed} \end{bmatrix} \]

\[ x' = Ax + Bu \]

Physics

State  Action

\[ u = Kx \]
Solving for \( K \)

\[
x' = Ax + Bu
\]

\[
u = Kx
\]
Solving for $K$

$u = Kx$

$x' = Ax + Bu$

$J = \int (x - x^*)^\top Q (x - x^*) + u^\top R u \ dt$

Go to $x^*$ min control
Solving for $K$

\[
J = \int (x - x^*)^\top Q (x - x^*) + u^\top R u \ dt
\]

Go to $x^*$ \hspace{1cm} \text{min control}
Learning $A, B$

$x' = Ax + Bu$
Learning $A, B$

$x' = Ax + Bu$

$$\min_{A,B} \|Ax + Bu - x'\|$$
Learning $A, B$

$$x' = Ax + Bu$$

$$\min_{A,B} \|Ax + Bu - x'\|$$
Today

What if?
Today

What if?

- Nonlinear?
- Wrong model?
Today

What if?

• Nonlinear?
• Wrong model?
Today

What if?

- Nonlinear?
- Wrong model?
Today

What if?

- Nonlinear?
- Wrong model?
- Learning?
- Safety?
Today

What if?

- Nonlinear?
- Wrong model?
- Learning?
- Safety?
Today

What if?

- Nonlinear?
- Wrong model?
- Learning?
- Safety?
Today

LQR
  ↓
Model Predictive Control
    ↓ Nonlinear
      ↓ Learning
        ↓ Explore/Exploit
Constraints ↓
  safe RL
LQR vs MPC

\[ u = Kx \]
\[ x' = Ax + Bu \]

Find \( K \) a priori
LQR vs MPC

\[ u = Kx \]
\[ x' = Ax + Bu \]

Find \( K \) a priori

MPC System

Model/Simulator

Find \( u \) at every step
\[ x = \begin{bmatrix} \theta \\ \dot{\theta} \\ \text{pos} \\ \text{speed} \end{bmatrix} \]
\[ x = \begin{bmatrix} \theta \\ \dot{\theta} \\ \text{pos} \\ \text{speed} \end{bmatrix} \]
\[ x = \begin{bmatrix} \theta \\ \dot{\theta} \\ \text{pos} \\ \text{speed} \end{bmatrix} \]
\[
x = \begin{bmatrix}
\theta \\
\dot{\theta} \\
pos \\
speed
\end{bmatrix}
\]

\[
x_0 \quad u \quad t
\]
\[
x = \begin{bmatrix}
\theta \\
\dot{\theta} \\
pos \\
speed
\end{bmatrix}
\]

\[x_0 \]

\[u \]

\[t \]
\[
x = \begin{bmatrix}
\theta \\
\dot{\theta} \\
pos \\
speed
\end{bmatrix}
\]
\[ x = \begin{bmatrix} \theta \\ \dot{\theta} \\ \text{pos} \\ \text{speed} \end{bmatrix} \]
$x = \begin{bmatrix} \theta & \dot{\theta} & \text{pos} & \text{speed} \end{bmatrix}$
<table>
<thead>
<tr>
<th>Off/Online</th>
<th>Full/Local model</th>
<th>Non-linear</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LQR vs MPC**
LQR vs MPC

Off/Online  Full/Local model  Non-linear
LQR vs MPC

Off/Online  Full/Local model  Non-linear
Example

\[ x_{i+1} = A(x_t)x_i + B(x_t)u_i \]
Example

\[ x_{i+1} = A(x_t)x_i + B(x_t)u_i \]

\[
\begin{align*}
\min_{u_{t+1}, \ldots, u_{t+H}} & \quad \sum_{i=1}^{H} \ldots \\
\text{s.t.} & \quad x_{i+1} = A(x_t)x_i + B(x_t)u_i
\end{align*}
\]
Example

$$x_{i+1} = A(x_t)x_i + B(x_t)u_i$$

$$\min_{u_{t+1}, \ldots, u_{t+H}} \sum_{i=1}^{H} x_{t+i}^\top Q x_{t+i} + u_{t+i}^\top R u_{t+i}$$

s.t. $$x_{i+1} = A(x_t)x_i + B(x_t)u_i$$
Example

\[ x_{i+1} = A(x_t)x_i + B(x_t)u_i \]

\[
\begin{align*}
\min_{u_{t+1}, \ldots, u_{t+H}} & \quad \sum_{i=1}^{H} x_{t+i}^\top Q x_{t+i} + u_{t+i}^\top R u_{t+i} \\
\text{s.t.} & \quad x_{i+1} = A(x_t)x_i + B(x_t)u_i
\end{align*}
\]

or \[ u_t = Kx_t \]

or \[ u_t, \ldots, u_{t+H} = NN(x_t, \theta) \quad \text{more flexible and hungry} \]
MPC

Replace full planning by local optimization
Learning

“Online LQR”
Learning

“Online LQR”

True dynamics: \[ x' = Ax + Bu \]

Model of the system: \[ x' = A_t x + B_t u \]
Learning

“Online LQR”

True dynamics: \( x' = Ax + Bu \)

Model of the system: \( x' = A_t x + B_t u \)

Start with \((A_0, B_0)\), update as you go
Learning safely?

Safety  Correct models  Explore/Exploit  Uncertainty
Safety
Safety
Safety

Ensure we don’t crash

avoid known bad outcomes / stay within known safe regime
A safe starting point

Uncertainty about the dynamics
A safe starting point

Uncertainty about the dynamics

There is a safe controller that can take over if $x \in \mathcal{X}_{\text{safe}}$
A safe starting point

Uncertainty about the dynamics

There is a safe controller that can take over if $x \in \mathcal{X}_{\text{safe}}$
A safe starting point

How do we get to $x^*$?
Safety objectives
Safety objectives

\[ \text{Loss} = J \]
Safety objectives

Loss = J
Safety objectives

$$\text{Loss} = \mathbb{E}[J]$$
Safety objectives

\[ \text{Loss} = \mathbb{E}[J] \]
Safety objectives

\[
\text{Loss} = \mathbb{E}[J] + \mathbb{E}[(J - \mathbb{E}[J])^2]
\]
Safety objectives

Loss = \mathbb{E}[J], but with safety constraints
Safety constraints

Each step is optimization $\rightarrow$ constrained opt.
Safety constraints

Each step is optimization → constrained opt.
Safety constraints

Each step is optimization $\rightarrow$ constrained opt.

$g(x) \geq 0$
Safety constraints

Each step is optimization \rightarrow \text{constrained opt.}

\[ g(x) \geq 0 \quad \theta \rightarrow u \rightarrow x \quad g(\theta) \geq 0 \]
Safety constraints

How do we get to $x^*$?
Uncertain transition dynamics

\[ x' = f(x, u) \]
\[ f(x, u) = h(x, u) + e(x, u) \]

\( h \) is known
\( e \) is not, but we have a probabilistic model
Uncertain transition dynamics

\[ x' = f(x, u) \]
\[ f(x, u) = h(x, u) + e(x, u) \]

\( h \) is known
\( e \) is not, but we have a probabilistic model

For \((\mu, \sigma)\) and \(\beta\), with high probability

\[ |\mu(x, u) - e(x, u)| \leq \beta \sigma(x, u) \]
Optimization with Uncertainty

Gaussian Processes Crash Course

Kernel $k(x, y)$
Optimization with Uncertainty

Gaussian Processes Crash Course

Kernel $k(x, y)$

Squared exponential

$k[x, x] = \alpha^2 \cdot \exp \left( -\frac{d^2}{2\lambda} \right)$

Kernel response
Optimization with Uncertainty

Gaussian Processes Crash Course

Kernel $k(x, y)$

Periodic

$$k(x, x') = \alpha^2 \cdot \exp\left[-\frac{2\sin(\pi d/\tau)^2}{\lambda^2}\right]$$
Optimization with Uncertainty

Gaussian Processes Crash Course

Kernel $k(x, y)$

Matérn 0.5

$k[x, x'] = \alpha^2 \cdot \exp \left[ -\frac{d}{\lambda^2} \right]$
Optimization with Uncertainty

Gaussian Processes Crash Course

Kernel $k(x, y)$

Linear regression: $\|Xw - y\|^2$

Kernel regression: $\|Kw - y\|^2$ where $K_{ij} = k(x_i, x_j)$

+ Covariances
Exploration/Exploitation
Exploration/Exploitation
Exploration/Exploitation
Exploration/Exploitation
Exploration/Exploitation
Safe Explore/Exploit
Safe Explore/Exploit

Performance $J(\theta)$

Safety $g(\theta)$

Parameters $\theta$
Safe Explore/Exploit
Safe Explore/Exploit
Safe Explore/Exploit

Performance $J(\theta)$

Safety $g(\theta)$

Parameters $\theta$
Safe Explore/Exploit

Performance $J(\theta)$

Safety $g(\theta)$

Parameters $\theta$
Safe Explore/Exploit
Safe Explore/Exploit
Ideal result

$\mathcal{X}^*$

$\mathcal{X}_{\text{safe}}$

$\mathcal{X}$
Slight issue: propagating uncertainty

$g(x)$ might be linear/convex but $g(\theta)$ is not
Slight issue: propagating uncertainty

\[ g(x) \] might be linear/convex but \( g(\theta) \) is not

\[ g(f(x, u = \text{Model}(x, \theta))) \]
Slight issue: propagating uncertainty

$g(x)$ might be linear/convex but $g(\theta)$ is not

$|\mu(x, u) - e(x, u)| \leq \beta \sigma(x, u)$
Slight issue: propagating uncertainty

$g(x)$ might be linear/convex but $g(\theta)$ is not

$$|\mu(x, u) - e(x, u)| \leq \beta \sigma(x, u)$$
Linearized uncertainty propagation
One last thing...
One last thing...
Simultaneous planning

- A safe default controller
- A definition of the boundaries
- A well specified Gaussian Process
- The Lipschitz constant of the model error
- Bayesian Optimization Model Predictive Control
Simultaneous planning

- A safe default controller
- A definition of the boundaries
- A well specified Gaussian Process
- The Lipschitz constant of the model error
- Bayesian Optimization Model Predictive Control