Approaches for Nonlinear Control
Recap: The Linear Quadratic Regulator (LQR)

\[
\min_{\pi_0, \ldots, \pi_{H-1}} \mathbb{E} \left[ x_H^T Q x_H + \sum_{h=0}^{H-1} (x_h^T Q x_h + u_h^T R u_h) \right] \]

such that

\[
\begin{align*}
    x_{h+1} &= A x_h + B u_h + w_h, \\
    u_h &= \pi_h(x_h) \\
    x_0 &\sim \mu_0, \\
    w_h &\sim N(0, \sigma^2 I)
\end{align*}
\]

Here, \( x_h \in \mathbb{R}^d, u_h \in \mathbb{R}^k \), the disturbance \( w_t \in \mathbb{R}^d \) is multi-variate normal, with covariance \( \sigma^2 I \);

\( A \in \mathbb{R}^{d \times d} \) and \( B \in \mathbb{R}^{d \times k} \) are referred to as system (or transition) matrices;

\( Q \in \mathbb{R}^{d \times d} \) and \( R \in \mathbb{R}^{k \times k} \) are psd matrices that parameterize the quadratic costs.
Recap: Optimal Control on LQR:

\[ V_H^*(x) = x^T Q x, \text{ define } P_H = Q, p_H = 0, \]

We have shown that \( V_h^*(x) = x^T P_h x + p_h \), where:

\[ P_h = Q + A^T P_{h+1} A - A^T P_{h+1} B (R + B^T P_{h+1} B)^{-1} B^T P_{h+1} A, \]

\[ p_h = \text{tr} \left( \sigma^2 P_{h+1} \right) + p_{h+1} \]

Along the way, we also have shown that \( \pi_h^*(x) = -K_h^* x \) where:

\[ \pi_h^*(x) = - (R + B^T P_{h+1} B)^{-1} B^T P_{h+1} A x \]

\[ := K_h^* \]

Optimal control has nothing to do with initial distribution, and the noise!
Today’s Question:
What about nonlinear and non-quadratic control?
Outline for today:

1. Local Linearization Approach
   (We will implement in HW1 for CartPole simulation)

2. Iterative LQR
Setting for Local Linearization Approach:

\[
\minimize \quad c(x_h, u_h) = d(u, u^*) + d(x_h, x^*) \\
\text{such that} \quad x_{h+1} = f(x_h, u_h), \quad u_h = \pi(x_h), \quad x_0 \sim \mu_0
\]

Goal: stabilizing around the goal \((x = x^*, u = u^*)\)

Assumptions:
Setting for Local Linearization Approach:

\[ \minimize c(x_h, u_h) = d(u, u^*) + d(x_h, x^*) \]

such that \( x_{h+1} = f(x_h, u_h), \ u_h = \pi(x_h) \), \( x_0 \sim \mu_0 \)

Assumptions:

1. We have black-box access to \( f \) & \( c \):
Setting for Local Linearization Approach:

\[
\min \quad \mathbb{E}_\pi \left[ \sum_{h=0}^{H-1} c(x_h, u_h) \right]
\]

such that \( x_{h+1} = f(x_h, u_h), u_h = \pi(x_h), x_0 \sim \mu_0 \)

Goal: stabilizing around the goal \((x = x^*, u = u^*)\)

\[ c(x_h, u_h) = d(u, u^*) + d(x_h, x^*) \]

Assumptions:

1. We have black-box access to \( f & c \): i.e., unknown analytical form, but can reset to any \((x, u)\), the black boxes outputs \(x', c\), where \(x' = f(x, u), c = c(x, u)\)
Setting for Local Linearization Approach:

\[ \minimize_{u_h} \; c(x_h, u_h) = d(u, u^*) + d(x_h, x^*) \]

such that \( x_{h+1} = f(x_h, u_h) \), \( u_h = \pi(x_h), x_0 \sim \mu_0 \)

Goal: stabilizing around the goal \( (x = x^*, u = u^*) \)

Assumptions:

1. We have black-box access to \( f \& c \):
   i.e., unknown analytical form, but can reset to any \((x, u)\), the black boxes outputs \( x', c \), where \( x' = f(x, u), c = c(x, u) \)

2. \( f \) is differentiable and \( c \) is double differentiable
Setting for Local Linearization Approach:

Goal: stabilizing around the goal \( (x = x^*, u = u^*) \)

\[
c(x_h, u_h) = d(u, u^*) + d(x_h, x^*)
\]

minimize \( \mathbb{E}_\pi \left[ \sum_{h=0}^{H-1} c(x_h, u_h) \right] \)

such that \( x_{h+1} = f(x_h, u_h), u_h = \pi(x_h), x_0 \sim \mu_0 \)

Assumptions:

1. We have black-box access to \( f \& c \):
   
i.e., unknown analytical form,
   but can reset to any \( (x, u) \),
   the black boxes outputs \( x', c \),
   where \( x' = f(x, u), c = c(x, u) \)

2. \( f \) is differentiable
   and \( c \) is double differentiable

\[
\nabla_x f(x, u), \nabla_u f(x, u), \nabla_x c(x, u), \nabla_u c(x, u),
\n\nabla_x^2 c(x, u), \nabla_u^2 c(x, u), \nabla_{x,u}^2 c(x, u)
\n\]
Local Linearization Approach

Assume that all possible initial states $x_0$ are close to $(x^*, u^*)$

$$x_0 \sim N(x^*, \Sigma)$$
Local Linearization Approach

Assume that all possible initial states $x_0$ are close to $(x^*, u^*)$

We can approximate $f(x, u)$ locally with First-order Taylor Expansion:

$$f(x, u) \approx f(x^*, u^*) + \nabla_x f(x^*, u^*)(x - x^*) + \nabla_u f(x^*, u^*)(u - u^*)$$
Local Linearization Approach

Assume that all possible initial states $x_0$ are close to $(x^*, u^*)$

We can approximate $f(x, u)$ locally with First-order Taylor Expansion:

$$f(x, u) \approx f(x^*, u^*) + \nabla_x f(x^*, u^*)(x - x^*) + \nabla_u f(x^*, u^*)(u - u^*)$$

where:

$$\nabla_x f(x, u) \in \mathbb{R}^{dx \times d}, \quad \nabla_x f(x, u)[i, j] = \frac{\partial f[i]}{\partial x[j]}(x, u)$$

$$\nabla_u f(x, u) \in \mathbb{R}^{dx \times k}, \quad \nabla_u f(x, u)[i, j] = \frac{\partial f[i]}{\partial u[j]}(x, u)$$
Local Linearization Approach

We can approximate $c(x, u)$ locally at $(x^*, u^*)$ with second-order Taylor Expansion:
Local Linearization Approach

We can approximate \( c(x, u) \) locally at \((x^*, u^*)\) with second-order Taylor Expansion:

\[
c(x, u) \approx c(x^*, u^*) + \nabla_x c(x^*, u^*)^T (x - x^*) + \nabla_u c(x^*, u^*)^T (u - u^*) \\
+ \frac{1}{2} (x - x^*)^T \nabla_x^2 c(x^*, u^*) (x - x^*) \\
+ \frac{1}{2} (u - u^*)^T \nabla_u^2 c(x^*, u^*) (u - u^*) \\
+ (u - u^*)^T \nabla_{u,x} c(x, u) (x - x^*)
\]
Local Linearization Approach

We can approximate $c(x, u)$ locally at $(x^*, u^*)$ with second-order Taylor Expansion:

$$c(x, u) \approx c(x^*, u^*) + \nabla_x c(x^*, u^*)^T (x - x^*) + \nabla_u c(x^*, u^*)^T (u - u^*)$$

$$+ \frac{1}{2} (x - x^*)^T \nabla_x^2 c(x^*, u^*) (x - x^*) + \frac{1}{2} (u - u^*)^T \nabla_u^2 c(x^*, u^*) (u - u^*) + (u - u^*)^T \nabla_{u,x}^2 c(x, u) (x - x^*)$$
Local Linearization Approach

\[ c(x, u) \approx c(x^*, u^*) + \nabla_x c(x^*, u^*)^T (x - x^*) + \nabla_u c(x^*, u^*)^T (u - u^*) \]
\[ + \frac{1}{2} (x - x^*)^T \nabla_x^2 c(x^*, u^*) (x - x^*) + \frac{1}{2} (u - u^*)^T \nabla_u^2 c(x^*, u^*) (u - u^*) + (u - u^*)^T \nabla_{u,x} c(x, u) (x - x^*) \]

\[ f(x, u) \approx f(x^*, u^*) + \nabla_x f(x^*, u^*) (x - x^*) + \nabla_u f(x^*, u^*) (u - u^*) \]
Local Linearization Approach

\[
\begin{align*}
  c(x, u) &\approx c(x^*, u^*) + \nabla_x c(x^*, u^*)^T (x - x^*) + \nabla_u c(x^*, u^*)^T (u - u^*) \\
  &\quad + \frac{1}{2} (x - x^*)^T \nabla^2_x c(x^*, u^*) (x - x^*) + \frac{1}{2} (u - u^*)^T \nabla^2_u c(x^*, u^*) (u - u^*) + (u - u^*)^T \nabla_{ux} c(x, u)(x - x^*) \\
  f(x, u) &\approx f(x^*, u^*) + \nabla_x f(x^*, u^*) (x - x^*) + \nabla_u f(x^*, u^*) (u - u^*) \\
  \therefore &:= A \\
  \therefore &:= B \\
  Q_z &= \frac{1}{2} \nabla^2_x c(x^*, u^*) \\
  R_z &= \frac{1}{2} \nabla^2_u c(x^*, u^*) \\
  \min_{\pi_0, \ldots, \pi_{H-1}} \mathbb{E} \left[ \sum_{h=0}^{H-1} (x_h^T Q x_h + u_h^T R u_h + u_h^T M x_h + x_h^T q + u_h^T r + c) \right] \\
  \text{such that} \quad x_{h+1} = A x_h + B u_h + v, \quad u_h = \pi_h(x_h) \quad x_0 \sim \mu_0,
\end{align*}
\]

(HW1 problem)
Summary So far:

For tasks such as balancing on goal state \((x^*, u^*)\):
we can perform **first order Taylor expansion on** \(f(x, u)\),
and **second order Taylor expansion on** \(c(x, u)\) around the balancing point \((x^*, u^*)\)
Summary So far:

For tasks such as balancing on goal state \((x^*, u^*)\):
we can perform **first order Taylor expansion on** \(f(x, u)\),
and **second order Taylor expansion on** \(c(x, u)\) around the balancing point \((x^*, u^*)\)

\[
\min_{\pi_0, \ldots, \pi_{H-1}} \mathbb{E} \left[ \sum_{h=0}^{H-1} (x_h^T Q x_h + u_h^T R u_h + u_h^T M x_h + x_h^T q + u_h^T r + c) \right]
\]
such that \(x_{h+1} = Ax_h + Bu_h + v, \quad u_h = \pi_h(x_h) \quad x_0 \sim \mu_0\)
Summary So far:

For tasks such as balancing on goal state \((x^*, u^*)\): we can perform **first order Taylor expansion** on \(f(x, u)\), and **second order Taylor expansion** on \(c(x, u)\) around the balancing point \((x^*, u^*)\).

\[
\min_{\pi_0, \ldots, \pi_{H-1}} \mathbb{E} \left[ \sum_{h=0}^{H-1} (x_h^T Q x_h + u_h^T R u_h + u_h^T M x_h + x_h^T q + u_h^T r + c) \right]
\]

such that \(x_{h+1} = A x_h + B u_h + v, \ u_h = \pi_h(x_h) \ x_0 \sim \mu_0\)

Last step: compute the optimal policy of the above problem, and test on the real system!
Some practical concerns in Local Linearization Approach

Note that \( c(x, u) \) might not even be convex;

So, \( \nabla_x^2 c(x, u) \) & \( \nabla_u^2 c(x, u) \) may not be positive definite
Some practical concerns in Local Linearization Approach

Note that $c(x, u)$ might not even be convex;

So, $\nabla_x^2 c(x, u) \& \nabla_u^2 c(x, u)$ may not be positive definite

In practice, we force them to be Positive definite:
Some practical concerns in Local Linearization Approach

Note that $c(x, u)$ might not even be convex;

So, $\nabla_x^2 c(x, u) \& \nabla_u^2 c(x, u)$ may not be positive definite

In practice, we force them to be Positive definite:

Given a symmetric matrix $H \in \mathbb{R}^{d \times d}$:

we compute the eigen-decomposition $H = \sum_{i=1}^{d} \sigma_i u_i u_i^T$,

and we approximate it as $H \approx \sum_{i=1}^{d} 1(\sigma_i > 0)\sigma_i u_i u_i^T + \lambda I$,

where $\lambda \in \mathbb{R}^+$
Some practical concerns in Local Linearization Approach

Recall our assumption: we only have black-box access to $f$ & $c$:

i.e., unknown analytical form, but given any $(x, u)$, the black boxes outputs $x', c$, where

$$x' = f(x, u), c = c(x, u)$$
Some practical concerns in Local Linearization Approach

Recall our assumption: we only have black-box access to $f$ & $c$:

i.e., unknown analytical form, but given any $(x, u)$, the black boxes outputs $x', c$, where

$$x' = f(x, u), \quad c = c(x, u)$$

Compute gradient using Finite differencing:

$$g : \mathbb{R} \rightarrow \mathbb{R}$$

$$g'(v_0) = \lim_{\delta \to 0} \frac{g(v_0 + \delta) - g(v_0 - \delta)}{2\delta}$$

$$\approx \frac{g(v_0 + \delta) - g(v_0 - \delta)}{2\delta}$$
Some practical concerns in Local Linearization Approach

Recall our assumption: we only have black-box access to $f$ & $c$: i.e., unknown analytical form, but given any $(x, u)$, the black boxes outputs $x', c$, where

$$x' = f(x, u), c = c(x, u)$$

Compute gradient using Finite differencing:

$$\frac{\partial f[i]}{\partial x[j]}(x, u) \approx \frac{f(x + \delta_j, u)[i] - f(x - \delta_j, u)[i]}{2\delta}, \text{where } \delta_j = [0, \ldots, 0, \delta_j, 0, \ldots, 0]^T$$
Some practical concerns in Local Linearization Approach

Recall our assumption: we only have black-box access to $f$ & $c$:

i.e., unknown analytical form, but given any $(x, u)$, the black boxes outputs $x', c$, where

$$x' = f(x, u), c = c(x, u)$$

Compute gradient using Finite differencing:

$$\frac{\partial f[i]}{\partial x[j]}(x, u) \approx \frac{f(x + \delta_j, u)[i] - f(x - \delta_j, u)[i]}{2\delta}, \text{where } \delta_j = [0, \ldots, 0, \delta_j, 0, \ldots, 0]^{\top} \text{ } j^{th} \text{ entry}$$

To compute second derivative, i.e.,

$$\frac{\partial^2 c}{\partial u[i] \partial x[j]}(x, u)$$
Some practical concerns in Local Linearization Approach

Recall our assumption: we only have black-box access to $f$ & $c$:

i.e., unknown analytical form, but given any $(x, u)$, the black boxes outputs $x', c$, where

$$x' = f(x, u), c = c(x, u)$$

Compute gradient using Finite differencing:

$$\frac{\partial f[i]}{\partial x[j]}(x, u) \approx \frac{f(x + \delta_j, u)[i] - f(x - \delta_j, u)[i]}{2\delta}, \text{where } \delta_j = [0, \ldots, 0, \delta_j, 0, \ldots, 0]^T$$

To compute second derivative, i.e.,

$$\frac{\partial^2 c}{\partial u[i]\partial x[j]}(x, u)$$

First implement FD procedure for $\frac{\partial c}{\partial u[i]}$, and then perform another FD wrt $x[j]$ on top of the FD procedure for $\frac{\partial c}{\partial u[i]}$
Summary for local linearization approach

1. we perform first order Taylor expansion on $f(x, u)$, and second order Taylor expansion on $c(x, u)$ around the balancing point $(x^*, u^*)$

2. We force Hessians $\nabla^2_x c(x, u) \& \nabla^2_u c(x, u)$ to be Positive Definite

3. Leverage Finite difference to approximate Gradients and Hessians

4. The approximation is an LQR from which we compute the optimal policy
Outline for today:

1. Local Linearization Approach  
   (We will implement it in HW1 for CartPole simulation)

2. Iterative LQR
Iterative LQR

Local Linearization approach could work if $x_0$ is very close to $(x^*)$

But when $x_0$ is far away from $x^*$, first/second-order Taylor expansion is not accurate anymore
Iterative LQR

Recall $x_0 \sim \mu_0$; denote $\mathbb{E}_{x_0 \sim \mu_0}[x_0] = \bar{x}_0$

Initialize $\bar{u}_0^0, \ldots, \bar{u}_{H-1}^0$.

Generate nominal trajectory: $\bar{x}_0, \bar{u}_h, \bar{x}_{h+1} = f(\bar{x}_h, \bar{u}_h), \ldots, \bar{x}_{H-1}, \bar{u}_{H-1}$
Iterative LQR

Recall $x_0 \sim \mu_0$; denote $\mathbb{E}_{x_0 \sim \mu_0}[x_0] = \bar{x}_0$

Initialize $u_0^0, \ldots, u_{H-1}^0$,

Generate nominal trajectory: $\bar{x}_0, \bar{u}_h, \bar{x}_{h+1} = f(\bar{x}_h, \bar{u}_h), \ldots, \bar{x}_{H-1}, \bar{u}_{H-1}$

For $t = 0$, \ldots,
Iterative LQR

Recall $x_0 \sim \mu_0$; denote $\mathbb{E}_{x_0 \sim \mu_0}[x_0] = \bar{x}_0$

Initialize $u_0^0, \ldots, u_{H-1}^0$.

Generate nominal trajectory: $\bar{x}_0, \bar{u}_h, \bar{x}_{h+1} = f(\bar{x}_h, \bar{u}_h), \ldots, \bar{x}_{H-1}, \bar{u}_{H-1}$

For $t = 0$, …,

Linearize $f(x, u)$ at $(\bar{x}_h, \bar{u}_h), \forall h$: $f(x, u) \approx f(\bar{x}_h, \bar{u}_h) + \nabla_x f(\bar{x}_h, \bar{u}_h)(x - \bar{x}_h) + \nabla_u f(\bar{x}_h, \bar{u}_h)(u - \bar{x}_h)$
Iterative LQR

Recall $x_0 \sim \mu_0$; denote $\mathbb{E}_{x_0 \sim \mu_0}[x_0] = \bar{x}_0$

Initialize $u_0^0, \ldots, u_{H-1}^0$.

Generate nominal trajectory: $\bar{x}_0, \bar{u}_h, \bar{x}_{h+1} = f(\bar{x}_h, \bar{u}_h), \ldots, \bar{x}_{H-1}, \bar{u}_{H-1}$

For $t = 0, \ldots$,

Linearize $f(x, u)$ at $(\bar{x}_h, \bar{u}_h), \forall h$: $f(x, u) \approx f(\bar{x}_h, \bar{u}_h) + \nabla_x f(\bar{x}_h, \bar{u}_h)(x - \bar{x}_h) + \nabla_u f(\bar{x}_h, \bar{u}_h)(u - \bar{x}_h)$

Quadratize $c(x, u)$ at $(\bar{x}_h, \bar{u}_h), \forall h$:

$$c(x, u) \approx \frac{1}{2} \begin{bmatrix} x - \bar{x}_h \end{bmatrix}^T \begin{bmatrix} \nabla_x^2 c(\bar{x}_h, \bar{u}_h), & \nabla_{x,u}^2 c(\bar{x}_h, \bar{u}_h) \\ \nabla_{u,x}^2 c(\bar{x}_h, \bar{u}_h), & \nabla_u^2 c(\bar{x}_h, \bar{u}_h) \end{bmatrix} \begin{bmatrix} x - \bar{x}_h \\ u - \bar{u}_h \end{bmatrix} + \begin{bmatrix} x - \bar{x}_h \\ u - \bar{u}_h \end{bmatrix}^T \begin{bmatrix} \nabla_x c(\bar{x}_h, \bar{u}_h) \\ \nabla_u c(\bar{x}_h, \bar{u}_h) \end{bmatrix} + c(\bar{x}_h, \bar{u}_h)$$
Iterative LQR

Recall $x_0 \sim \mu_0$; denote $\mathbb{E}_{x_0 \sim \mu_0}[x_0] = \bar{x}_0$

Initialize $u_0^0, \ldots, u_{H-1}^0$,

Generate nominal trajectory: $\bar{x}_0, \bar{u}_h, \bar{x}_{h+1} = f(\bar{x}_h, \bar{u}_h) \ldots, \bar{x}_{H-1}, \bar{u}_{H-1}$

For $t = 0 \ldots$,

Linearize $f(x, u)$ at $(\bar{x}_h, \bar{u}_h)$, $\forall h$: $f(x, u) \approx f(\bar{x}_h, \bar{u}_h) + \nabla_x f(\bar{x}_h, \bar{u}_h)(x - \bar{x}_h) + \nabla_u f(\bar{x}_h, \bar{u}_h)(u - \bar{x}_h)$

Quadratize $c(x, u)$ at $(\bar{x}_h, \bar{u}_h)$, $\forall h$: $c(x, u) \approx \frac{1}{2} \begin{bmatrix} x - \bar{x}_h \end{bmatrix}^T \begin{bmatrix} \nabla_x^2 c(\bar{x}_h, \bar{u}_h) & \nabla_{x,u}^2 c(\bar{x}_h, \bar{u}_h) \\ \nabla_{u,x}^2 c(\bar{x}_h, \bar{u}_h) & \nabla_u^2 c(\bar{x}_h, \bar{u}_h) \end{bmatrix} \begin{bmatrix} x - \bar{x}_h \end{bmatrix} + \begin{bmatrix} x - \bar{x}_h \end{bmatrix}^T \begin{bmatrix} \nabla_x c(\bar{x}_h, \bar{u}_h) \\ \nabla_u c(\bar{x}_h, \bar{u}_h) \end{bmatrix} + c(\bar{x}_h, \bar{u}_h)$

Formulate **time-dependent** LQR and compute its optimal control $\pi_0, \ldots, \pi_{H-1}$
Iterative LQR

Recall $x_0 \sim \mu_0$; denote $\mathbb{E}_{x_0 \sim \mu_0}[x_0] = \bar{x}_0$

Initialize $u_0^0, \ldots, u_{H-1}^0$.

Generate nominal trajectory: $\bar{x}_0, \bar{u}_h, \bar{x}_{h+1} = f(\bar{x}_h, \bar{u}_h), \ldots, \bar{x}_{H-1}, \bar{u}_{H-1}$

For $t = 0\ldots$,

Linearize $f(x, u)$ at $(\bar{x}_h, \bar{u}_h)$, $\forall h$: $f(x, u) \approx f(\bar{x}_h, \bar{u}_h) + \nabla_x f(\bar{x}_h, \bar{u}_h)(x - \bar{x}_h) + \nabla_u f(\bar{x}_h, \bar{u}_h)(u - \bar{x}_h)$

Quadratize $c(x, u)$ at $(\bar{x}_h, \bar{u}_h)$, $\forall h$:

\[
c(x, u) \approx \frac{1}{2} \begin{bmatrix} x - \bar{x}_h \\ u - \bar{u}_h \end{bmatrix}^T \begin{bmatrix} \nabla_x^2 c(\bar{x}_h, \bar{u}_h) & \nabla_{x,u}^2 c(\bar{x}_h, \bar{u}_h) \\ \nabla_{u,x}^2 c(\bar{x}_h, \bar{u}_h) & \nabla_u^2 c(\bar{x}_h, \bar{u}_h) \end{bmatrix} \begin{bmatrix} x - \bar{x}_h \\ u - \bar{u}_h \end{bmatrix} + \begin{bmatrix} x - \bar{x}_h \\ u - \bar{u}_h \end{bmatrix}^T \begin{bmatrix} \nabla_x c(\bar{x}_h, \bar{u}_h) \\ \nabla_u c(\bar{x}_h, \bar{u}_h) \end{bmatrix} + c(\bar{x}_h, \bar{u}_h)
\]

Formulate time-dependent LQR and compute its optimal control $\pi_0, \ldots, \pi_{H-1}$

Set new nominal trajectory: $\bar{u}_0 = \pi_0(\bar{x}_0), \bar{u}_h = \pi_h(\bar{x}_h)$, where $\bar{x}_{h+1} = f(\bar{x}_h, \bar{u}_h)$.
Iterative LQR

Consider iteration $t$, we have computed $\pi^t_h, \forall h$:

\[
\begin{align*}
\overline{u}_t &= \pi^t_0 (\overline{x}_0) \\
\overline{x}_{H-1} &= A_h x_t + B_h u_t + v_u \\
A_h &= \nabla_x f(\overline{x}_h, \overline{u}_h) \\
B_h &= \nabla_u f(\overline{x}_h, \overline{u}_h)
\end{align*}
\]
Iterative LQR

Consider iteration $t$, we have computed $\pi_h^t$, $\forall h$:

After linearization and qudartization around $H$ waypoints $(\bar{x}_h, \bar{u}_h), \forall h$, re-arrange terms, we get:

$$\min_{\pi_0, \ldots, \pi_{H-1}} \mathbb{E} \left[ \sum_{h=0}^{H-1} (x_h^T Q_h x_h + u_h^T R_h u_h + u_h^T M_h x_h + x_h^T q_h + u_h^T r_h + c_h) \right]$$

such that $x_{h+1} = A_h x_h + B_h u_h + v_h$ $u_h = \pi_h(x_h)$ $x_0 \sim \mu_0$;
Some practical considerations of Iterative LQR:

1. We still want to use the eigen-decomposition trick to ensure positive definite Hessians;
Some practical considerations of Iterative LQR:

1. We still want to use the eigen-decomposition trick to ensure positive definite Hessians;

2. We want to use linear-search to get monotonic improvement:
Some practical considerations of Iterative LQR:

1. We still want to use the eigen-decomposition trick to ensure positive definite Hessians;

2. We want to use linear-search to get monotonic improvement:

Given the previous nominal control $\tilde{u}_0^t, \ldots, \tilde{u}_{t-1}^t$, and the latest computed controls $\tilde{u}_0, \ldots, \tilde{u}_{H-1}$
Some practical considerations of Iterative LQR:

1. We still want to use the eigen-decomposition trick to ensure positive definite Hessians;

2. We want to use linear-search to get monotonic improvement:

Given the previous nominal control $\tilde{u}_0^t, \ldots, \tilde{u}_{H-1}^t$, and the latest computed controls $\tilde{u}_0, \ldots, \tilde{u}_{H-1}$

We want to find $\alpha \in [0,1]$ such that $\tilde{u}_h^{t+1} := \alpha \tilde{u}_h^t + (1 - \alpha) \tilde{u}_h$ has the smallest cost,
Some practical considerations of Iterative LQR:

1. We still want to use the eigen-decomposition trick to ensure positive definite Hessians;

2. We want to use linear-search to get monotonic improvement:

Given the previous nominal control $\tilde{u}_0^t, \ldots, \tilde{u}_{H-1}^t$, and the latest computed controls $\tilde{u}_0^t, \ldots, \tilde{u}_{H-1}^t$

We want to find $\alpha \in [0, 1]$ such that $\tilde{u}_h^{t+1} := \alpha \tilde{u}_h^t + (1 - \alpha) \tilde{u}_h$ has the smallest cost,

$$\min_{\alpha \in [0, 1]} \sum_{h=0}^{H-1} c(x_h, \tilde{u}_h^{t+1}),$$

s.t., $x_{h+1} = f(x_h, \tilde{u}_h^{t+1}), \tilde{u}_h^{t+1} = \alpha \tilde{u}_h^t + (1 - \alpha) \tilde{u}_h, x_0 = \bar{x}_0$
Example:

2-d car navigation

Cost function is designed such that it gets to the goal without colliding w/ obstacles (red)
Summary:

**Local Linearization:**
Approximate an LQR at the balance (goal) position \((x^*, u^*)\);
and then solve the approximated LQR;

**Iterative LQR**
Iterate between (1) forming an LQR around the current nominal trajectory,
(2) compute a new nominal trajectory using the optimal policy of the LQR;
Starting from next week:

We will move on to data-driven approach for computing approximately optimal policy

1. Model-based RL: certainty equivalence

2. Model-free RL: Fitted Value Iteration
\[ g : \mathbb{R} \to \mathbb{R} \]

\[ g''(x) \]

\[ g'(x) = \frac{g(x+h) - g(x-h)}{2h} \]

\[ FD(g, x) = \frac{g(x+h) - g(x-h)}{2h} \]

\[ g''(x) = \frac{FD(g, x+h) - FD(g, x-h)}{2h} \]

\[ FD_2(g, x) \sim \frac{g'(x+h) - g'(x-h)}{2h} \]
\[ f : \mathbb{R}^2 \to \mathbb{R} \]

\[ f(a, b) = \begin{cases} 
  f_a(a, b) \\
  f_b(a, b) 
\end{cases} \]

\[ \nabla f(x)[i,j] = \frac{\partial f}{\partial x[i,j]}(x) \]

\[ f : \mathbb{R}^3 \to \mathbb{R}^2 \]

\[ \nabla \times f \rightarrow \frac{\partial f[i,j]}{\partial x[i,j]} \]