Last Time:

- Stochastic Optimal Control
- LQG

Today:

- Robust Control
- Minimax DDP

---

* Context:

- Stochastic methods assume random noise inputs. LQG assumes noise is zero-mean. This is a poor model for parameter uncertainties or unmodeled dynamics.

- ILC fits under the general category of adaptive control: Try to learn/adapt to the unknown model parameters online to achieve optimal performance.

- Robust control methods take a more conservative approach: Try to design a controller ahead of time that will still work under a range of different possible models. Generally, sacrifice some performance for safety/robustness.

- Historically: Early work on LQR showed strong natural robustness. Later it was discovered that LQG (LQR + KF) could be arbitrarily fragile: An infinitessimal perturbation can make the system unstable.

* Robust Control Problem:

- Assume a system model of the form:

$$x_{n+1} = f(x_n, u_n, w_n)$$

"disturbance"

- Disturbance input can be anything: constant parameter offsets, time-varying or non-smooth forces, bounds on prediction error of your model.

- We typically assume some bounds on w, e.g. $\|w\| < \varepsilon$

- We solve the following optimization problem:

$$\min_{\substack{x_{1:N} \\ u_{1:N-1}}} \max_{w_{1:N-1}} J = \sum_{n=1}^{N-1} l_n(x_n, u_n, w_n) + l_N(x_N)$$

$$\text{s.t.} \quad x_{n+1} = f(x_n, u_n, w_n)$$

$$x_n \in \mathcal{X}$$
$$u_n \in \mathcal{U}$$
$$w_n \in \mathcal{W}$$

- We're looking for a saddle point where the cost is minimized w.r.t. x, u and maximized w.r.t. w

- Problems like this are called "minimax" optimization problems

- Can also interpret this as 2-player zero-sum game where the u-player and w-player both get to choose inputs to the system and have competing objectives.

- In general, these problems are very hard

- For linear systems, there is a general theory developed in the 1980s-90s called H_∞ control that generalizes LQG. There is a MATLAB toolbox.

- For nonlinear systems, we can find local solutions with approximations.

---

* Minimax DDP:

  - Use a local linear/quadratic Taylor expansion to iteratively find a locally-optimal trajectory and feedback policy.

  - Dynamics Expansion:

$$f(x+\Delta x, u+\Delta u, w+\Delta w) \approx f(x,u,w) + A\Delta x + B\Delta u + D\Delta w$$

- Action-Value Function Expansion:

$$S(x+\Delta x, u+\Delta u, w+\Delta w) \approx S(x,u,w) + \begin{bmatrix} g_x \\ g_u \\ g_w \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta w \end{bmatrix}$$

$$+ \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta w \end{bmatrix}^T \underbrace{\begin{bmatrix} G_{xx} & G_{xu} & G_{xw} \\ G_{ux} & G_{uu} & G_{uw} \\ G_{wx} & G_{wu} & G_{ww} \end{bmatrix}}_{G} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta w \end{bmatrix}$$

- Bellman Equation:

$$V_{k-1}(x+\delta x) = \min_{\delta u} \max_{\delta w} \Big[S(x,u,w) + g_x^T \delta x + g_u^T \delta u + g_w^T \delta w$$

$$+ \tfrac{1}{2}\delta x^T G_{xx} \delta x + \tfrac{1}{2}\delta u^T G_{uu} \delta u + \tfrac{1}{2}\delta w^T G_{ww} \delta w + \delta x^T G_{xu} \delta u$$

$$+ \delta x^T G_{xw} \delta w + \delta u^T G_{uw} \delta w \Big]$$

$$\frac{\partial [\;]}{\partial u} = g_u + G_{uu}\delta u + G_{ux}\delta x + G_{uw}\delta w = 0$$

$$\frac{\partial [\;]}{\partial w} = g_w + G_{ww}\delta w + G_{wx}\delta x + G_{wu}\delta u = 0$$

note there's no distinction between min/max here. Need $G_{ww} < 0$

$$\Rightarrow \delta u = -G_{uu}^{-1}\left[g_u + G_{ux}\delta x + G_{uw}\delta w\right]$$

$$\delta w = -G_{ww}^{-1}\left[g_w + G_{wx}\delta x + G_{wu}\delta u\right]$$

- Plug into each other to get rid of cross terms:

$$\Rightarrow g_u + G_{uu}\delta u + G_{ux}\delta x - G_{uw}G_{ww}^{-1}\left[g_w + G_{wx}\delta x + G_{wu}\delta u\right] = 0$$

$$\Rightarrow \delta u = -d - K\delta x, \quad d = (G_{uu} - G_{uw}G_{ww}^{-1}G_{wu})^{-1}(g_u - G_{uw}G_{ww}^{-1}g_w)$$

$$K = (G_{uu} - G_{uw}G_{ww}^{-1}G_{wu})^{-1}(G_{ux} - G_{uw}G_{ww}^{-1}G_{wx})$$

* From LQR   * New robust terms

$$\Delta w = -e - L\Delta x, \quad e = (G_{ww} - G_{wu} G_{uu}^{-1} G_{uw})^{-1} (g_w - G_{wu} G_{uu}^{-1} g_u)$$

$$L = (G_{ww} - G_{wu} G_{uu}^{-1} G_{uw})^{-1} (G_{wx} - G_{wu} G_{uu}^{-1} G_{ux})$$

- Plug Δu and Δw back into S_{n-1} to get $V_{n-1}(x+\Delta x)$

$$V_{n-1}(x+\Delta x) \approx V_{n-1}(x) + p_{n-1}^T \Delta x + \frac{1}{2} \Delta x^T P_{n-1} \Delta x$$

$$p_{n-1} = g_x - G_{xu} G_{uu}^{-1} g_u - G_{xw} G_{ww}^{-1} g_w$$

$$P_{n-1} = G_{xx} - G_{xu} G_{uu}^{-1} G_{ux} - G_{xw} G_{ww}^{-1} G_{wx}$$

- Action-Value expansions are exactly the same as standard DDP

$$\frac{\partial S}{\partial x} = \frac{\partial l}{\partial x} + \frac{\partial V}{\partial f} \frac{\partial f}{\partial x} \Rightarrow \boxed{\begin{aligned} g_x &= \nabla_x l(x,u,w) + A_n^T p_{n+1} \\ g_u &= \nabla_u l(x,u,w) + B_n^T p_{n+1} \\ g_w &= \nabla_w l(x,u,w) + D_n^T p_{n+1} \end{aligned}}$$

- Biggest difference vs. standard DDP is that action-value Hessian G is now quasi-definite. It should have:

  $\dim(x) + \dim(u)$ positive eigenvalues

  $\dim(w)$ negative eigenvalues

- When writing down a quadratic cost function, Hessian w.r.t. w should negative definite:

$$J = \sum_{k=1}^{N-1} \frac{1}{2} x_n^T Q x_n + \frac{1}{2} u_n^T R u_n + \frac{1}{2} w_n^T S w_n + \frac{1}{2} x_N^T Q_N x_N$$

$$Q, Q_N, R \succ 0 \quad , \quad S \prec 0$$

- When regularizing G, do $G + \begin{bmatrix} \alpha I & 0 \\ 0 & -\alpha I \end{bmatrix}$

- Small $\|S\|$ allows large $\|w\| \Rightarrow$ more robust
- Large $\|S\|$ penalizes $w \Rightarrow$ small $\|w\| \Rightarrow$ less robust
- In the limit $S \rightarrow -\infty$, we recover standard DDP.