Lecture 8: Open-Loop Dynamic Optimization

Sondre Pedersen

January 31, 2025

1 Optimization in Control

This is the start of the control part of the course.

Optimization problems we study	**Use of optimization in control**
• Class: Linear programming • Method: Simplex	• (Some use in MPC, but not typical) • (Sometimes used for computing economic setpoints for PID control)
• Class: Quadratic programming • Method: Active set	• Dynamic optimizatin using linear models – Open loop dynamic optimization – Closed loop dynamic optimization / MPC – Linear Quadratic Control
• Class: Nonlinear programming • Method: Sequential Quadratic Programming	• Dynamic optimization using nonlinear models • Nonlinear Model Predictive Control

Static vs dynamic optimization - Does time matter?

When solving practical problems (that is, we optimize some process) we have two cases:

- Time independent process $\implies$ static optimization
  - Common in finance, economic optimization
  - Farming example from previous lectures
- Time dependent process $\implies$ dynamic optimization
  - More common in cybernetics/control engineering
  - Mechanical systems (boat, drone, robot, ...), chemical process (chemical reactor, process plant, ...), electrical/energy system

It is possible to go for a third option: quazi-dynamic optimization. If the process changes slowly over time, it is possible to assume that it is not changing at all and using static optimization. To compensate, re-solve regularly.

Quazi-dynamic example - Oil production

Given a floating production ship with n wells. These wells will produce a stream of oil, gass and water. The ship has a separator to put these in different tanks. The ouput of the separator (g: gass, o: oil, w: water) from well i:

$$q_g = \sum q_{gi}$$
$$q_o = \sum q_{oi}$$
$$q_w = \sum q_{wi}$$

From day to day the ouputs do not change very much, so we can make a static model.

$$(*) \begin{cases} q_{qi} = \alpha_{gi} \times q_i, \ i = 1, \dots, n \\ q_{wi} = \alpha_{wi} \times q_i, \ i = 1, \dots, n \\ q_{oi} = (1 - \alpha_{gi} - \alpha_{wi}) \times q_i, \ i = 1, \dots, n \\ \sum_{i=1}^{n} q_{gi} \leq q_{g,max} \\ \sum_{i=1}^{n} q_{wi} \leq q_{w,max} \end{cases}$$

Optimize

$$\min_z \quad -\sum_{i=1}^{n} q_{oi} \qquad \text{s.t.} \qquad (*)$$
$$\text{where } z = (q_1, g_{g_1}, q_{wq}, q_{o_1}, q_1, \dots)^\top$$

Here α is a gass-oil ratio. We assume the parameters are constant for one day, but they might change the next. When they do, re-optimize.

2 Dynamic models

$$\begin{aligned} x_{t+1} &= g(x_t, u_t) && \text{(general nonlinear dynamic model)} \\ x_{t+1} &= A_t x_t + B_t u_t && \text{(LTV)} \\ x_{t+1} &= A x_t + B u_t && \text{(LTI)} \end{aligned}$$

In the world, models are not linear. But they can still be a good approximation. To get this approximation, we linearize $x_{t+1} = g(x_t, u_t)$. Linearize about either $\begin{cases} \text{trajectory } \overline{x}_{t+1} = g(\overline{x}_t, \overline{u}_t) \\ \text{stationary point } \overline{x} = g(\overline{x}, \overline{u}) \end{cases}$

Define perturbation $x_t = \overline{x}_t + \delta x_t, \ u_t = \overline{u}_t + \delta u_t$. Now

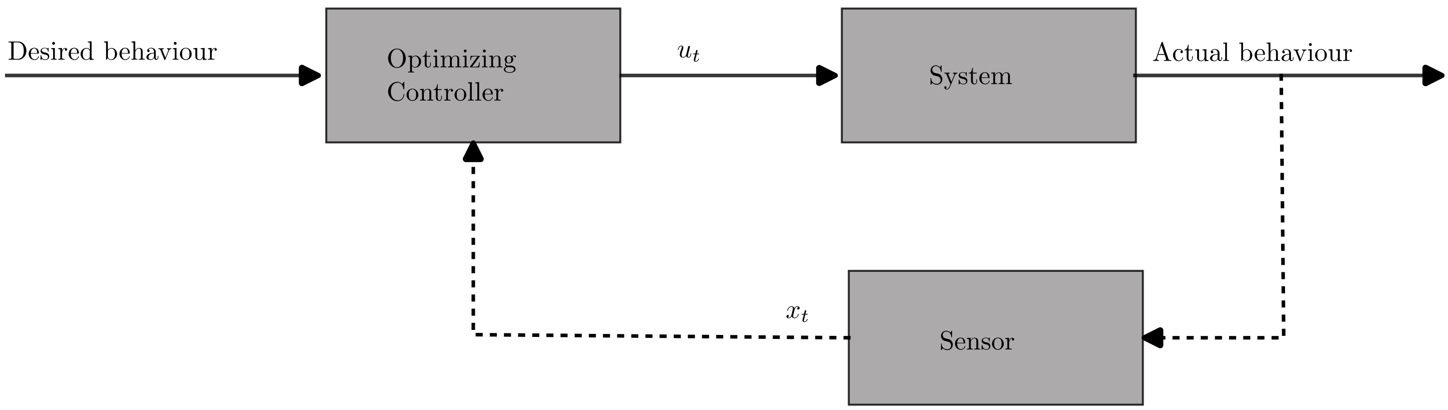
$$\begin{aligned} x_{t+1} &= g(\overline{x}_t + \delta x_t, \overline{u}_t + \delta u_t) \\ &\approx g(\overline{x}_t, \overline{u}_t) + \left.\frac{\partial g}{\partial x}\right|_{\overline{x}_t, \overline{u}_t} \times \delta x_t + \left.\frac{\partial g}{\partial u}\right|_{\overline{x}_t, \overline{u}_t} \times \delta u_t \\ \delta x_{t+1} &= A_t \delta x_t + B_t \delta u_t \qquad \text{(LTV)} \\ \delta x_{t+1} &= A \delta x_t + B \delta u_t \qquad \text{(LTI)} \end{aligned}$$

General dynamic optimization problem -

$$\begin{aligned} \min_z \quad & \sum_{t=0}^{N-1} f_t(x_{t+1}, u_t) \\ \text{s.t.} \quad & x_{t+1} = g(x_t, u_t), \ t = 0, \dots, N-1 \\ & h_t(x_t, u_t) \leq 0, \ t = 0, \dots, N \end{aligned}$$

x_0 is given. $z = \left[u_0^\top, x_1^\top, u_1^\top, \dots, u_{N-1}^\top, x_N^\top\right]^\top$

- N: prediction horizon
- f_t: stage cost
- u_t: system input
- g_t: dynamic model
- h_t: state and input constraints



Typical objectives (stage costs) in dynamic optimization: penalize deviations from a constant reference/setpoint (regulation), or deviations from a reference trajectory (tracking). Other typical objectives include: maximize profit, maximize production, minimize costs, limit tear, reach endpoints fast.

General quadratic stage cost:

$$f_t(x_{t+1}, u_t) = \frac{1}{2} x_{t+1}^\top Q_t K_{t+1} + d_{x,t}^\top x_t + \frac{1}{2} u_t^\top R_t u_t + d_{u,t}^\top u_t.$$

This general form reduces nicely to special cases.

A quadratic objective is good for two reasons: (i) it is convenient mathematically. The gradient is linear, and smoothness is good for calculations. (ii) It has a natural response to deviations. A small deviation does not matter too much, but big deviations will cause a stronger response.

Standard linear dynamic optimization problem

$$\begin{aligned}
\min_{z} \quad & \frac{1}{2} \sum_{t=0}^{N-1} x_{t+1}^\top Q x_{t+1} + \frac{1}{2} u_t^\top R u_t \\
\text{s.t.} \quad & x_{t+1} = A x_t + B u_t, \ t = 0, \dots, N-1 \\
& x^{low} \leq x_t \leq x^{high}, \ t = 1, \dots, N \\
& u^{low} \leq u_t \leq u^{high}, \ t = 0, \dots, N-1 \\
& \\
& x_0 : \ \text{given} \\
& z = \left[u_0^\top, x_1^\top, u_1^\top, x_2^\top, \dots, u_{N-1}^\top, x_N^\top \right]^\top
\end{aligned}$$

Can be solved using 'Batch approach v1, "Full space"'. Great name. With this approach we can write the problem as

$$\begin{aligned}
\min_{z} \quad & \frac{1}{2} z^\top \begin{bmatrix} R & & & & \\ & Q & & & \\ & & R & & \\ & & & Q & \\ & & & & \ddots \end{bmatrix} z \\
\text{s.t.} \quad & \begin{bmatrix} -B & I & 0 & 0 & 0 & \dots \\ 0 & -A & -B & I & 0 & \dots \\ 0 & 0 & -A & -B & I & \dots \\ \vdots & & & \ddots & & \end{bmatrix} z = \begin{bmatrix} Ax_0 \\ 0 \\ \\ \vdots \end{bmatrix} \\
& \begin{bmatrix} I & 0 & 0 & 0 & \dots \\ -I & 0 & 0 & 0 & \dots \\ 0 & I & 0 & 0 & \dots \\ 0 & -I & 0 & 0 & \dots \\ & & \vdots & & \end{bmatrix} z \geq \begin{bmatrix} u^{low} \\ -u^{high} \\ x^{low} \\ x^{high} \\ \vdots \end{bmatrix}
\end{aligned}$$

More common in the industry is 'Batch approach v2, "Reduced space"/"condensing"'

$$
\begin{aligned}
x_1 &= Ax_0 + Bu_0 \\
x_2 &= Ax_1 + Bu_1 = A(Ax_0 + Bu_0) + Bu_1 = A^2 x_0 + ABu_0 + Bu_1 \\
x_3 &= Ax_2 + Bu_2 = A(\dots) + Bu_2 = A^3 x_0 + A^2 Bu_0 + ABu_1 + Bu_2
\end{aligned}
$$

Can be written on matrix form

$$
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \end{bmatrix} x_0 + \begin{bmatrix} B & 0 & 0 & \dots \\ AB & B & 0 & \dots \\ A^2B & AB & B & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix}
$$

$$
\rightarrow x = S^x x_0 + S^u u
$$

Use the above formulation to write the dynamic optimization problem as

$$
\begin{aligned}
\min \quad & \frac{1}{2}(S^x x_0 + S^u u)^\top \begin{bmatrix} Q & & & \\ & Q & & \\ & & \ddots & \\ & & & Q \end{bmatrix} (S^x x_0 + S^u u) + u^\top \begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix} u \\
\text{s.t.} \quad & \begin{bmatrix} x^{low} \\ \vdots \\ x^{low} \end{bmatrix} \leq S^x x_0 + S^u u \leq \begin{bmatrix} x^{high} \\ \vdots \\ x^{high} \end{bmatrix} \\
& \begin{bmatrix} u^{low} \\ \vdots \\ u^{low} \end{bmatrix} \leq u \leq \begin{bmatrix} u^{high} \\ \vdots \\ u^{high} \end{bmatrix}
\end{aligned}
$$