Lecture 11 — Optimal Control

- The Maximum Principle Revisited
- Examples
- Numerical methods/Optimica
- Examples, Lab 3

Material

- Lecture slides,
  including material by J. kesson, Automatic Control LTH
- Glad & Ljung, part of Chapter 18

Goal

To be able to

- solve simple problems using the maximum principle
- formulate advanced problems for numerical solution

Outline

- **The Maximum Principle Revisited**
- Examples
- Numerical methods/Optimica
- Example — Double integrator
- Example — Alfa Laval Plate Reactor

Problem Formulation (1)

Standard form (1):

$$\text{Minimize} \int_0^{t_f} \overbrace{L(x(t), u(t))}^{\text{Trajectory cost}} \, dt + \overbrace{\phi(x(t_f))}^{\text{Final cost}}$$
$$\dot{x}(t) = f(x(t), u(t))$$
$$u(t) \in U, \quad 0 \le t \le t_f, \qquad t_f \text{ given}$$
$$x(0) = x_0$$

$x(t) \in R^n$, $u(t) \in R^m$
U control constraints

Here we have a fixed end-time t_f.

The Maximum Principle (18.2)

Introduce the **Hamiltonian**

$$H(x, u, \lambda) = L(x, u) + \lambda^T(t) f(x, u).$$

If problem (1) has a solution $\{u^*(t), x^*(t)\}$, then

$$\min_{u \in U} H(x^*(t), u, \lambda(t)) = H(x^*(t), u^*(t), \lambda(t)), \quad 0 \leq t \leq t_f,$$

where $\lambda(t)$ solves the **adjoint equation**

$$d\lambda(t)/dt = -H_x^T(x^*(t), u^*(t), \lambda(t)), \quad \text{with} \quad \lambda(t_f) = \phi_x^T(x^*(t_f))$$

Remarks

The Maximum Principle gives **necessary** conditions
A pair $(u^*(\cdot), x^*(\cdot))$ is called **extremal** if the conditions of the Maximum Principle are satisfied. Many extremals can exist.
The maximum principle gives all possible candidates.
However, there might not exist a minimum!

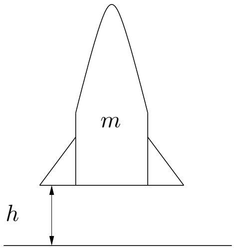
Example

Minimize $x(1)$ when $\dot{x}(t) = u(t)$, $x(0) = 0$ and $u(t)$ is free

Goddard's Rocket Problem revisited

How to send a rocket as high up in the air as possible?

$$\frac{d}{dt}\begin{pmatrix} v \\ h \\ m \end{pmatrix} = \begin{pmatrix} \frac{u-D}{m} - g \\ v \\ -\gamma u \end{pmatrix}$$



$(v(0), h(0), m(0)) = (0, 0, m_0)$, $g, \gamma > 0$
u motor force, $D = D(v, h)$ air resistance
Constraints: $0 \leq u \leq u_{max}$ and $m(t_f) = m_1$ (empty)
Optimization criterion: $\max_u h(t_f)$

Problem Formulation (2)

$$\min_{\substack{t_f \geq 0 \\ u:[0,t_f]\to U}} \int_0^{t_f} L(x(t), u(t))\, dt + \phi(x(t_f))$$

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0$$

$$\psi(x(t_f)) = 0$$

Note the differences compared to standard form:

- End constraints $\psi(x(t_f)) = 0$
- t_f free variable (i.e., not specified *a priori*)

The Maximum Principle–General Case (18.4)

Introduce the Hamiltonian

$$H(x, u, \lambda, n_0) = n_0 L(x, u) + \lambda^T(t) f(x, u)$$

If problem (2) has a solution $u^*(t), x^*(t)$, then there is a vector function $\lambda(t)$, a number $n_0 \geq 0$, and a vector $\mu \in R^r$ such that $[n_0 \;\; \mu^T] \neq 0$ and

$$\min_{u \in U} H(x^*(t), u, \lambda(t), n_0) = H(x^*(t), u^*(t), \lambda(t), n_0), \quad 0 \leq t \leq t_f,$$

where

$$\dot{\lambda}(t) = -H_x^T(x^*(t), u^*(t), \lambda(t), n_0)$$
$$\lambda(t_f) = n_0 \phi_x^T(x^*(t_f)) + \Psi_x^T(x^*(t_f))\mu$$

If the end time t_f is free, then $H(x^*(t_f), u^*(t_f), \lambda(t_f), n_0) = 0$.

Normal/abnormal cases

Can scale $n_0, \mu, \lambda(t)$ by the same constant
Can reduce to two cases

- $n_0 = 1$ (normal)
- $n_0 = 0$ (abnormal, since L and ϕ don't matter)

As we saw before (18.2): fixed time t_f and no end constraints $\Rightarrow$ normal case

Hamilton function is constant

H is constant along extremals (x^*, u^*)
Proof (in the case when $u^*(t) \in \mathrm{Int}(U)$):

$$\frac{d}{dt}H = H_x\dot{x} + H_\lambda\dot{\lambda} + H_u\dot{u} = H_x f - f^T H_x^T + 0 = 0$$

Feedback or feed-forward?

Example:

$$\frac{dx}{dt} = u, \qquad x(0) = 1$$
$$\text{minimize } J = \int_0^\infty \left(x^2 + u^2\right) dt$$

$J_{min} = 1$ is achieved for

$$u(t) = -e^{-t} \qquad \text{open loop} \tag{1}$$

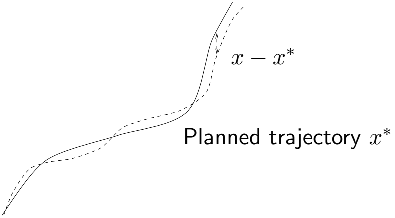
or

$$u(t) = -x(t) \qquad \text{closed loop} \tag{2}$$

(**??**) $\Longrightarrow$ stable system
(**??**) $\Longrightarrow$ asympt. stable system
Sensitivity for noise and disturbances differ!!

Reference generation using optimal control

Note that the optimization problem makes no distinction between open loop control $u^*(t)$ and closed loop control $u^*(t, x)$. Feedback is needed to take care of disturbances and model errors.
Idea: Use the optimal open loop solution $u^*(t), x^*(t)$ as reference values to a linear regulator that keeps the system close to the desired trajectory
Efficient for large setpoint changes.



Recall Linear Quadratic Control

$$\text{minimize } x^T(t_f)Q_N x(t_f) + \int_0^{t_f} \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} Q_{11} \\ Q_{12} \\ Q_{12}^T \\ Q_{22} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

where

$$\dot{x} = Ax + Bu, \quad y = Cx$$

Optimal solution if $t_f = \infty$, $Q_N = 0$, all matrices constant, and x measurable:

$$u = -Lx$$

where $L = Q_{22}^{-1}(Q_{12} + B^T S)$ and $S = S^T > 0$ solves

$$SA + A^T S + Q_{11} - (Q_{12} + SB)Q_{22}^{-1}(Q_{12} + B^T S) = 0$$

Second Variations

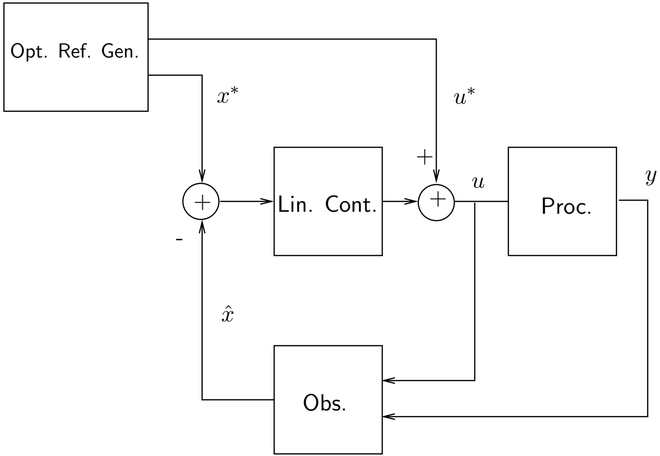
Approximating $J(x,u)$ around (x^*,u^*) to second order

$$\delta^2 J = \frac{1}{2}\delta_x^T \phi_{xx}\,\delta_x + \frac{1}{2}\int_{t_0}^{t_f} \begin{bmatrix} \delta_x \\ \delta_u \end{bmatrix}^T \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_u \end{bmatrix} dt$$

$$\delta\dot{x} = f_x\delta_x + f_u\delta_u$$

where $J = J^* + \delta^2 J + \ldots$ is a Taylor expansion of the criterion and $\delta_x = x - x^*$ and $\delta_u = u - u^*$.
Treat this as a new optimization problem. Linear time-varying system and quadratic criterion. Gives optimal controller

$$u - u^* = L(t)(x - x^*)$$



Take care of deviations with linear controller

Outline

Example: Optimal heating

$$\text{Minimize} \quad \int_0^{t_f=1} P(t)\,dt$$

$$\begin{aligned} \text{when} \quad & \dot{T} = P - T \\ & 0 \le P \le P_{max} \\ & T(0) = 0, \quad T(1) = 1 \end{aligned}$$

T temperature
P heat effect

Solution

Hamiltonian

$$H = n_0 P + \lambda P - \lambda T$$

Adjoint equation

$$\dot{\lambda}^T = -H_T = -\frac{\partial H}{\partial T} = \lambda \qquad \lambda(1) = \mu$$

$$\Rightarrow \quad \lambda(t) = \mu e^{t-1}$$

$$\Rightarrow \quad H = \underbrace{\left(n_0 + \mu e^{t-1}\right)}_{\sigma(t)} P - \lambda T$$

At optimality

$$P^*(t) = \begin{cases} 0, & \sigma(t) > 0 \\ P_{max}, & \sigma(t) < 0 \end{cases}$$

Solution

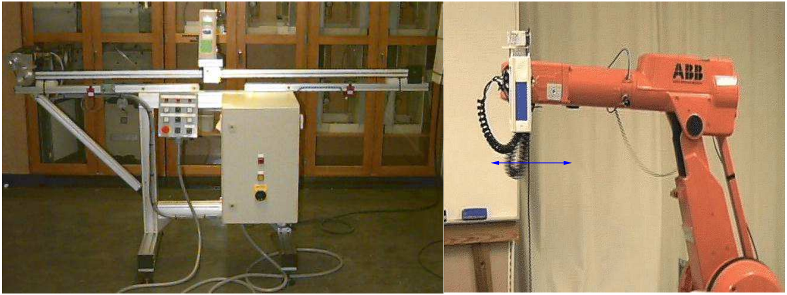
$\mu = 0 \Rightarrow (n_0, \mu) = (0, 0) \Rightarrow$ Not allowed!
$\mu \neq 0 \Rightarrow$ Constant P or just one switch!
$T(t)$ approaches one from below, so $P \neq 0$ near $t = 1$. Hence

$$P^*(t) = \begin{cases} 0, & 0 \leq t \leq t_1 \\ P_{\text{max}}, & t_1 < t \leq 1 \end{cases}$$

$$T(t) = \begin{cases} 0, & 0 \leq t \leq t_1 \\ \int_{t_1}^{1} e^{-(t-\tau)} P_{\text{max}} \, d\tau = \left(e^{-(t-1)} - e^{-(t-t_1)}\right) P_{\text{max}}, & t_1 < t \leq 1 \end{cases}$$

Time t_1 is given by $T(1) = \left(1 - e^{-(1-t_1)}\right) P_{\text{max}} = 1$
Has solution $0 \leq t_1 \leq 1$ if $P_{\text{max}} \geq \dfrac{1}{1 - e^{-1}}$

Example – The Milk Race



Move milk in minimum time without spilling!
[M. Grundelius – Methods for Control of Liquid Slosh]

[movie]

Minimal Time Problem

NOTE! Common trick to rewrite criterion into “standard form”!!

$$\text{minimize } \ t_f = \text{minimize } \ \int_0^{t_f} 1\, dt$$

Control constraints

$$|u(t)| \leq u_i^{max}$$

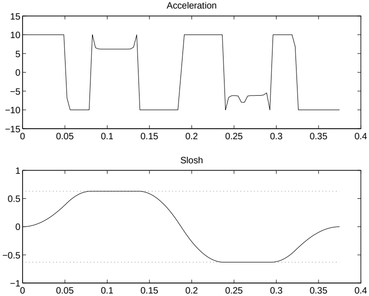
No spilling

$$|Cx(t)| \leq h$$

Optimal controller has been found for the milk race
Minimal time problem for linear system $\dot{x} = Ax + Bu$, $y = Cx$ with control constraints $|u_i(t)| \leq u_i^{max}$. Often bang-bang control as solution

Results- milk race

Maximum slosh $\phi_{max} = 0.63$
Maximum acceleration $=$ 10 m/s^2
Time optimal acceleration profile



Optimal time $=$ 375 ms, industrial $=$ 540ms

Outline

- ◦ The Maximum Principle Revisited
- ◦ Examples
- • **Numerical methods/Optimica**
- ◦ Example — Double integrator
- ◦ Example — Alfa Laval Plate Reactor

Numerical Methods for Dynamic Optimization

- Many algorithms
  - Applicability highly model-dependent
    (ODE, DAE, PDE, hybrid?)
    - Calculus of variations
    - Single/Multiple Shooting
    - Simultaneous methods
    - Simulation-based methods
  - Analogy with different simulation algorithms
    (but larger diversity)
- Heavy programming burden to use numerical algorithms
  - Fortran
  - C
- Engineering need for high-level descriptions

Modelica — A Modeling Language

- Modelica is increasingly used in industry
  - Expert knowledge
  - Capital investments
- Usage so far
  - Simulation (mainly)
- Other usages emerge
  - Sensitivity analysis
  - Optimization
  - Model reduction
  - System identification
  - Control design

Optimica and JModelica — A Research Project

- Shift focus:
  - from *encoding*
  - to *problem formulation*
- Enable dynamic optimization of Modelica models
  - State of the art numerical algorithms
- Develop a high level description for optimization problems
  - Extension of the Modelica language
- Develop prototype tools
  - JModelica and The Optimica Compiler
  - Code generation

Outline

Optimica—An Example

$$\min_{u(t)} \int_0^{t_f} 1\, dt$$

subject to the dynamic constraint

$$\begin{aligned} \dot{x}(t) &= v(t), \quad x(0) = 0 \\ \dot{v}(t) &= u(t), \quad v(0) = 0 \end{aligned}$$

and

$$\begin{aligned} x(t_f) &= 1 \\ v(t_f) &= 0 \\ v(t) &\leq 0.5 \\ -1 \leq u(t) &\leq 1 \end{aligned}$$

A Modelica Model for a Double Integrator

A double integrator model

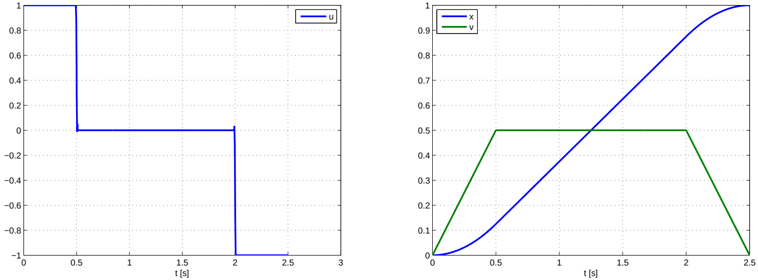
```
model DoubleIntegrator
  Real x(start=0);
  Real v(start=0);
  input Real u;
equation
  der(x)=v;
  der(v)=u;
end DoubleIntegrator;
```

The Optimica Description

Minimum time optimization problem

```
optimization DIMinTime (objective=cost(finalTime),
                        startTime=0,
                        finalTime(free=true,initialGu
  Real cost;
  DoubleIntegrator di(u(free=true,initialGuess=0.0));
equation
  der(cost) = 1;
constraint
  finalTime>=0.5;
  finalTime<=10;
  di.x(finalTime)=1;
  di.v(finalTime)=0;
  di.v<=0.5;
  di.u>=-1; di.u<=1;
end DIMinTime;
```

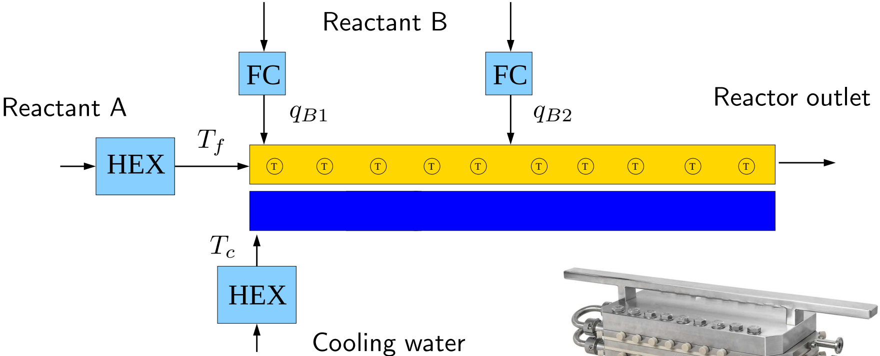
Optimal Double Integrator Profiles

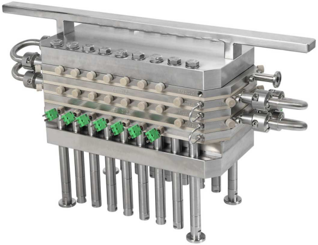


Outline

- ◦ The Maximum Principle Revisited
- ◦ Examples
- ◦ Numerical methods/Optimica
- ◦ Example — Double integrator
- **Example — Alfa Laval Plate Reactor**

Optimal Start-up of a Plate Reactor





- Achieve safe start-up $T_r \leq T_{max}$
- Maximize the conversion
- Minimize the start-up time

The Optimization Problem

Reduce sensitivity of the nominal start-up trajectory by:

- Introducing a constraint on the accumulated concentration of reactant B
- Introducing high frequency penalties on the control inputs

$$\min_u \int_0^{t_f} \alpha_A c_{A,out}^2 + \alpha_B c_{B,out}^2 + \alpha_{B1} q_{B1,f}^2 + \alpha_{B2} q_{B2,f}^2 + \alpha_{T_1} \dot{T}_f^2 + \alpha_{T_2} \dot{T}_c^2 \; dt$$

$$\text{subject to} \quad \dot{x} = f(x,u)$$

$$T_{r,i} \leq 155, \quad i = 1..N \quad c_{B,1} \leq 600, \quad c_{B,2} \leq 1200$$

$$0 \leq q_{B1} \leq 0.7, \quad 0 \leq q_{B2} \leq 0.7$$

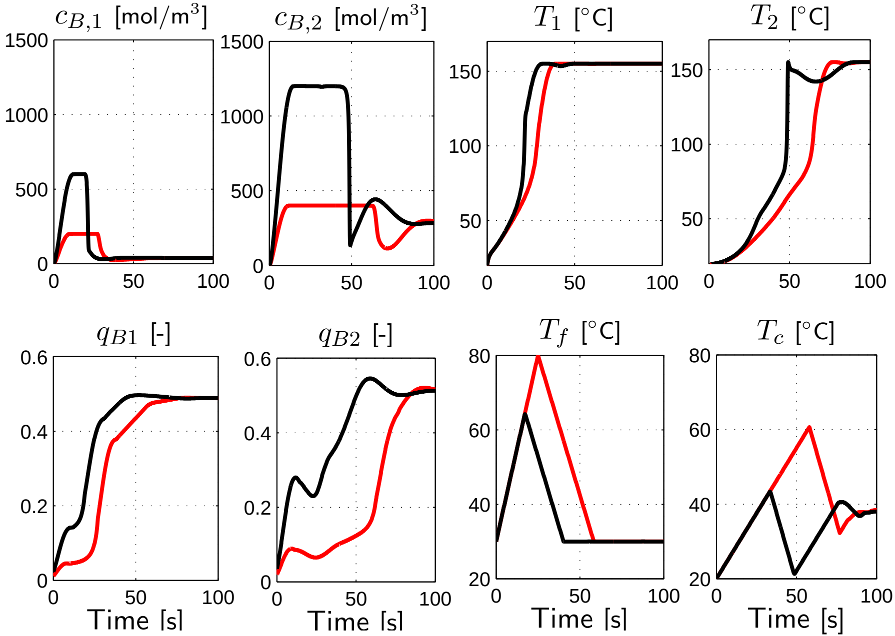
$$-1.5 \leq \dot{T}_f \leq 2, \quad -1.5 \leq \dot{T}_c \leq 0.7$$

$$30 \leq T_f \leq 80, \quad 20 \leq T_c \leq 80$$

The Optimization Problem—Optimica

Robust optimization formulation

```
optimization PlateReactorOptimization (objective=cost(finalTime),
                                       startTime=0,
                                       finalTime=150)
  PlateReactor pr(u_T_cool_setpoint(free=true), u_TfeedA_setpoint(free=true),
                  u_B1_setpoint(free=true), u_B2_setpoint(free=true));
  parameter Real sc_u = 670/50 "Scaling factor";
  parameter Real sc_c = 2392/50 "Scaling factor";
  Real cost(start=0);
equation
  der(cost) = 0.1*pr.cA[30]^2*sc_c^2 + 0.025*pr.cB[30]^2*sc_c^2 + 1*pr.u_B1_setpoint_f^2
              1*pr.u_B2_setpoint_f^2 + 1*der(pr.u_T_cool_setpoint)^2*sc_u^2 +
              1*der(pr.u_TfeedA_setpoint)^2*sc_u^2;

constraint
  pr.Tr/u_sc<=(155+273)*ones(30);

  pr.cB[1]<=200/sc_c; pr.cB[16]<=400/sc_c;

  pr.u_B1_setpoint>=0; pr.u_B1_setpoint<=0.7;
  pr.u_B2_setpoint>=0; pr.u_B2_setpoint<=0.7;

  pr.u_T_cool_setpoint>=(15+273)/sc_u; pr.u_T_cool_setpoint<=(80+273)/sc_u;
  pr.u_TfeedA_setpoint>=(30+273)/sc_u; pr.u_TfeedA_setpoint<=(80+273)/sc_u;

  der(pr.u_T_cool_setpoint)>=-1.5/sc_u; der(pr.u_T_cool_setpoint)<=0.7/sc_u;
  der(pr.u_TfeedA_setpoint)>=-1.5/sc_u; der(pr.u_TfeedA_setpoint)<=2/sc_u;
end PlateReactorOptimization;
```



Almost as fast, but more robust with lower c_B-constraints

Summary

- The Maximum Principle Revisited
- Examples
- Numerical methods/Optimica
- Example — Double integrator
- Example — Alfa Laval Plate Reactor