Advanced and Multivariable Control

Optimal Control

Riccardo Scattolini

Optimal control

Basic idea: the control problem is transformed into an ***optimization*** one, where the goal is to compute the control variable by ***minimising a suitable performance index*** (or cost function) under ***constraints*** on the input, state, and output variables

Extremely flexible approach, which allows to consider nonlinear systems and to formulate different objectives and constraints

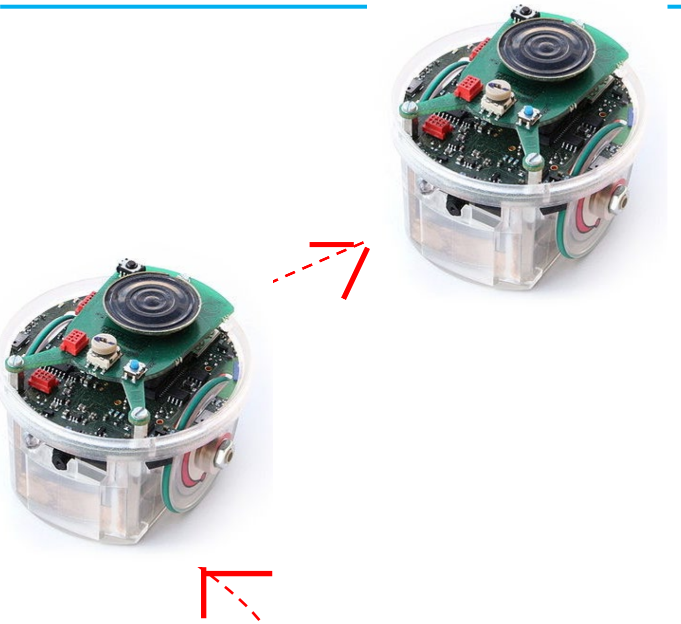
Widely used in many fields, such as all the ***engineering*** problems, and in particular in ***aerospace, mechanical, chemical*** fields, but also in ***economics***, ***finance***, ***biological*** systems,…

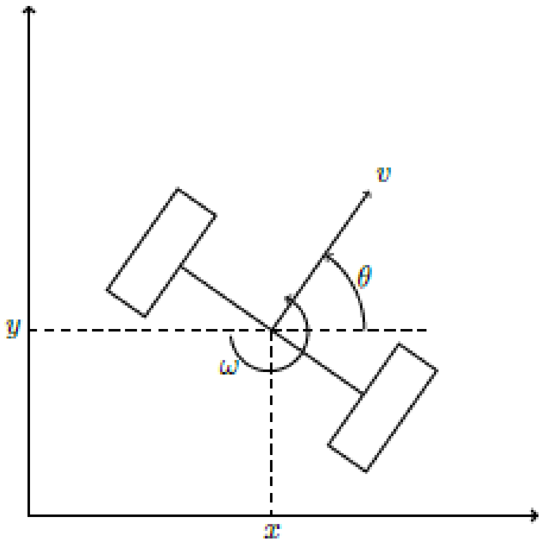
Different approaches to its solution, based on sufficient conditions formuated by means of ***dynamic programming***, or necessary conditions (Pontryagin's ***Maximum Principle***)

It may be very difficult to find a solution, many ***numerical methods*** are available. Close connections with ***Reinforcement Learning***

It is the precursor ***of Model Predictive Control***, the most popular method for advanced process control (last Chapter of our course)

We'll give a hint of the dynamic programming approach, and then we'll specialize to the simplest case of application to ***linear systems*** with simple, ***quadratic cost functions***





Mobile robot - path tracking problem

Model - unicycle

$$
\begin{aligned}
\dot{x} &= \cos\theta\, v \\
\dot{y} &= \sin\theta\, v \\
\dot{\theta} &= \omega.
\end{aligned}
$$

control variables

$$v,\ \omega$$

$$\text{reference trajectory: } x^o(t),\ y^o(t),\ t \in [0, T]$$

Optimization problem

weights

performance index, or cost function

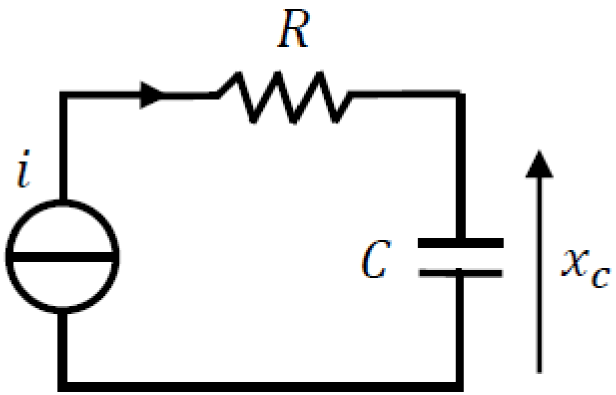
$$\min_{v,\omega} \int_0^T ((x(\tau) - x^o(\tau))^2 + (y(\tau) - y^o(\tau))^2) + r_1 v^2(\tau) + r_2 \omega^2(\tau))d\tau +$$

$$+ q((x(T) - x^o(T))^2 + (y(T) - y^o(T)^2))$$

constraints

subject to the system’s dynamics and

$$|v(t)| \leq \bar{v},\ |\omega(t)| \leq \bar{\omega},\ t \in [0, T]$$



Model

$$C\dot{x}_c(t) = i(t)$$

Goals

1. minimize the energy stored in the capacitor at time ***t=T*** given a known initial condition
2. minimize the power dissipated in the resistance

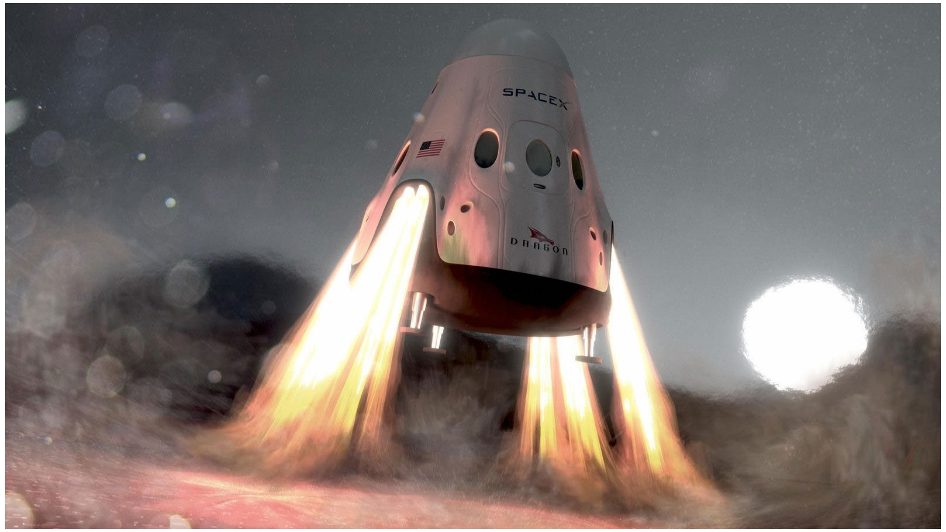
weight *terminal cost*

$$min_i J = s x_c^2(T) + \int_{t_0}^{T} R i^2(\tau) d\tau, \quad s \geq 0$$

performance index, or cost function

subject to the system's dynamics
and to possible constraints on ***i*** and $\boldsymbol{x_c}$

constraints



Spacecraft landing

Model

total mass

Jet thrusters

$$M\ddot{h} = -gM + u \qquad \dot{M} = -ku$$

$$M(0) = M_0,\, h(0) = h_0,\, \dot{h}(0) = \dot{h}_0, \text{ and } k > 0.$$

Optimal control problem: optimally manage the thrusters ***u*** in order to minimize the final time ***T*** under constraints

$$\min_{\boldsymbol{u}} \boldsymbol{T}$$

performance index, or cost function

$$M(t) \geq m \qquad h(t) \geq 0$$

$$h(T) = \dot{h}(T) = 0$$

constraints

Generic stabilization problem

$Q \geq 0$ weights the deviation of the state from zero

$R > 0$ weights the input

$S \geq 0$ weights the deviation of the final state from zero

design parameters

$$\min_u J = \int_{t_0}^{T} (x'(\tau)Qx(\tau) + u'(\tau)Ru(\tau))d\tau + x'(T)Sx(T)$$

Example: second order system

$$Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}, R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}, S = 0 \xrightarrow{\textit{the integrand is}} q_1x_1^2 + q_2x_2^2 + r_1u_1^2 + r_2u_2^2$$

($\boldsymbol{q_1}$,$\boldsymbol{q_2}$) >> ($\boldsymbol{r_1}$,$\boldsymbol{r_2}$) → I am mainly interested to quickly bring the state to zero (leads to a fast closed-loop system)

($\boldsymbol{r_1}$,$\boldsymbol{r_2}$) >> ($\boldsymbol{q_1}$,$\boldsymbol{q_2}$) → I don’t want to use too much the control variables (typically, I’ll obtain a slow closed-loop system)

$\boldsymbol{q_1}$ >> $\boldsymbol{q_2}$ → I want that the first state $\boldsymbol{x_1}$ goes much faster to zero than the second state $\boldsymbol{x_2}$

and so on…

Formal problem statement

$$\dot{x}(t) = f(x(t), u(t)), \quad x \in R^n, \quad u \in R^m$$

f continuously differentiable with respect to its arguments, x measurable

Goal: compute an "optimal control" $u^o(t)$, $t \in [t_0, T]$ minimizing

$$J(x(t_0), u(\cdot), t_0) = \int_{t_0}^{T} l(x(\tau), u(\tau))d\tau + m(x(T))$$

l, m continuously differentiable

subject to the system's dynamics and state and input constraints:

$$x(t) \in X, \quad u(t) \in U$$

$X \subseteq R^n$, $U \subseteq R^m$ are compact sets containing the origin

Denote by $u_{[a,b]}$ the control functions $u(\cdot)$ in the interval $[a,b]$ and define

$$J^0(x(t),t) = \min_{u[t,T]} J(x(t),u(\cdot),t) = \int_t^T l(x(\tau),u(\tau))d\tau + m(x(T)) \qquad t \in [t_0,T]$$

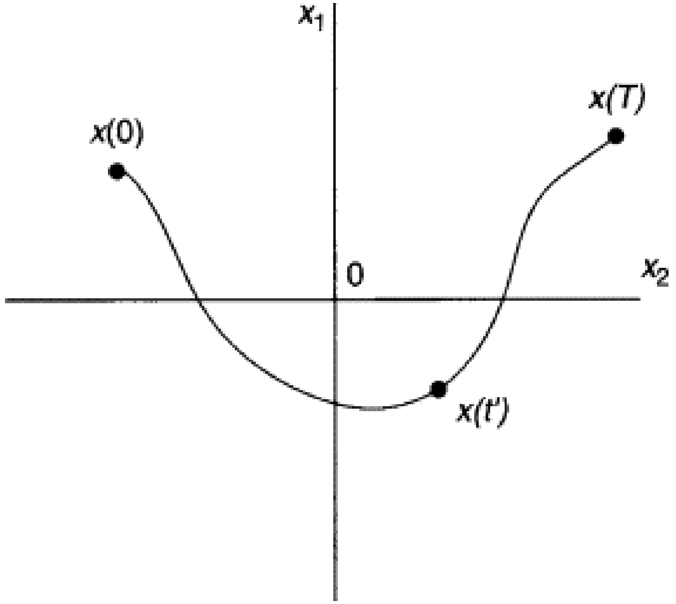
Note that J^0 and J depend on $x(t)$, i.e. on the current value of the state, while they do not depend on the state evolution up to time t

To proceed, we need the Bellman's principle of optimality

Bellman's principle of optimality

From any point of an optimal trajectory, the remaining trajectory is optimal for the corresponding problem over the remaining number of stages, or time interval, initiated at that point

Richard Bellman, the father of Dynamic Programming



If some trajectory in the phase space connects the initial ***x(0)*** and terminal ***x(T)*** points and is optimal in the sense of some cost functional, then the sub-trajectory, connecting any intermediate point ***x(t')*** of the same trajectory with the same terminal point ***x(T)***, should also be optimal.

Alexander S. Poznyak, in Advanced Mathematical Tools for Automatic Control Engineers: Deterministic Techniques, Volume 1, 2008

From

$$J^0(x(t),t) = \min_{u[t,T]} J(x(t),u(\cdot),t) = \int_t^T l(x(\tau),u(\tau))d\tau + m(x(T))$$

$$J^0(x(t),t)$$
$$= \min_{u[t,t_1]} \left\{ \min_{u[t_1,T]} \left[\int_t^{t_1} l(x(\tau),u(\tau))d\tau + \int_{t_1}^{T} l(x(\tau),u(\tau))d\tau + m(x(T)) \right] \right\}$$

$$\min_{u[t_1,T]} \left[\underbrace{\int_t^{t_1} l(x(\tau),u(\tau))d\tau}_{\text{does not depend on } u[t_1,T]} + \int_{t_1}^{T} l(x(\tau),u(\tau))d\tau + m(x(T)) \right]$$

$$J^0(x(t),t)$$
$$= \min_{u[t,t_1]} \left\{ \int_t^{t_1} l(x(\tau),u(\tau))d\tau + \min_{u[t_1,T]} \left[\int_{t_1}^{T} l(x(\tau),u(\tau))d\tau + m(x(T)) \right] \right\}$$

$$
\begin{aligned}
& J^0(x(t),t) \\
= \quad & \min_{u[t,t_1]} \left\{ \int_t^{t_1} l(x(\tau),u(\tau))d\tau + \min_{u[t_1,T]} \left[\int_{t_1}^{T} l(x(\tau),u(\tau))d\tau + m(x(T)) \right] \right\}
\end{aligned}
$$

recall Bellman!

$$
J^0(x(t),t) = \min_{u[t,t_1]} \left\{ \int_t^{t_1} l(x(\tau),u(\tau))d\tau + J^o(x(t_1),t_1) \right\}
$$

if the optimal control has been applied in the interval $[t_1,T)$, the optimal cost of the state trajectory starting at t is obtained by minimizing the sum of the cost incurred from t to t_1 plus the optimal cost from t_1 to T

$$J^0(x(t),t) = \min_{u[t,t_1]} \left\{ \int_t^{t_1} l(x(\tau),u(\tau))d\tau + J^o(x(t_1),t_1) \right\}$$

mean value theorem with $\alpha \in [0,1]$

$$t_1 = t + dt$$

$$J^0(x(t),t) = \min_{u[t,t+dt]} \{ l(x(t+\alpha dt), u(t+\alpha dt))dt + J^o(x(t+dt), t+dt) \}$$

expand $J^o(x(t+dt),t+dt)$

$$J^o(x(t+dt),t+dt) = J^o(x(t),t) + \frac{\partial J^o(x(t),t)}{\partial x}\frac{dx(t)}{dt}dt + \frac{\partial J^o(x(t),t)}{\partial t}dt + O(dt)^2$$

$$\cancel{J^0(x(t),t)}$$

$$= \min_{u[t,t+dt]} \left\{ l(x(t+\alpha dt), u(t+\alpha dt))dt + \cancel{J^o(x(t),t)} + \frac{\partial J^o(x(t),t)}{\partial x}\frac{dx(t)}{dt}dt \right.$$

$$\left. + \frac{\partial J^o(x(t),t)}{\partial t}dt + O(dt)^2 \right\}, \quad \alpha \in [0,1]$$

$\downarrow$ divide by dt, and let $dt \to 0$

does not depend on ***u***

$$0 = \min_{u[t]} \left\{ l(x(t),u(t)) + \frac{\partial J^o(x(t),t)}{\partial x} f(x(t),u(t)) + \frac{\partial J^o(x(t),t)}{\partial t} \right\}$$

$\downarrow$

at a fixed time t, x and u must be considered as vectors, instead of functions of time

$\downarrow$

$$\boxed{\frac{\partial J^o(x,t)}{\partial t} = -\min_u \left\{ l(x,u) + \frac{\partial J^o(x,t)}{\partial x} f(x,u) \right\}}$$

Hamilton Jacobi Bellman equation

$$\downarrow \quad J(x,u(\cdot),T) = m(x)$$

*does not depend on **u***

$$\boxed{J^o(x,T) = m(x)}$$

How to use the HJB equation?

$$\frac{\partial J^o(x,t)}{\partial t} = -\min_u \left\{ l(x,u) + \frac{\partial J^o(x,t)}{\partial x} f(x,u) \right\}$$

$$J^o(x,T) = m(x)$$

$\boxed{\text{Step 1}}$ compute the value u^o minimizing

$$\left\{ l(x,u) + \frac{\partial J^o(x,t)}{\partial x} f(x,u) \right\} \longrightarrow u^o = \kappa\left(x, \frac{\partial J^o(x,t)}{\partial x}\right)$$

$\boxed{\text{Step 2}}$ compute the function $J^o(x,t)$ satisfying the HJB equation

$$\frac{\partial J^o(x,t)}{\partial t} = -l\left(x, \kappa\left(x, \frac{\partial J^o(x,t)}{\partial x}\right)\right) - \frac{\partial J^o(x,t)}{\partial x} f\left(x, \kappa\left(x, \frac{\partial J^o(x,t)}{\partial x}\right)\right) \quad , \quad J^o(x,T) = m(x)$$

$\boxed{\text{Step 3}}$ use $\frac{\partial J^o(x,t)}{\partial x}$ in the control law $u^o = \kappa\left(x, \frac{\partial J^o(x,t)}{\partial x}\right) \longrightarrow u^o = \kappa(x,t)$

Comments

From a computational point of view, this is a very tough probem, many different approaches have been proposed

The computations must proceed backwards in time. You start from the final value $J^o(x,T) = m(x)$
and move on with reverse time

The resulting control law will be of the form ***u=κ(x,t)***, i.e. a state feedback control law even though an open-loop optimization problem has been formulated