INSA Toulouse

Course "Inverse problems, control, data assimilation" by J. Monnier.

Practical: optimal control of a 1D vehicle trajectory

1 Introduction

Optimal control of a dynamical model enables to plan an optimized vehicle trajectory, given a target trajectory. This is dynamic optimization (called model predictive control too).

The common strategy to solve dynamic control problems is to numerically integrate the dynamic model in discrete time intervals. Next, the numerical model solution is compared to the desired trajectory (the target) and the difference is minimized by calibrating / tuning the control variable.

One way to proceed is to apply this strategy time step by time step: this is the direct method proposed here.

Objective: To compute a control that automatically regulates a vehicle velocity.
Method: Direct dynamic method based on the Sequential Quadratic Programming (SQP) algorithm.

2 Problem set up

2.1 The vehicle trajectory model

Let us denote by: $x(t)$ the vehicle position at time t (ms^{-1}), m the vehicle mass (kg), u the pedal position (in %), K a friction coefficient (Ns/m), and G a gain (in $ms^{-1}(\%\text{pedal})^{-1}$).

The considered dynamic trajectory model is 1D and basic. It reads:

$$m\, x"(t) = -K\ x'(t) + G\ K\ u(t) \text{ in } (0,T) \tag{2.1}$$

with the initial condition $(x', x)(0) = (0, 0)$.
The state of the system is $x(t)$; the control variable is $u(t)$. (m, G, K) are constant parameters of the model.

Since there is no 0th order term in the differential equation above, the following change of variable is natural: $y(t) = x'(t)$. This reduces the order of the differential equation.
Then, by introducing the constant $\tau = m/K$, the state equation simplifies as:

$$y'(t) = -\tau(y(t) - G\ u(t)) \text{ in } (0,T) \tag{2.2}$$

with the I.C. $y(0) = 0$.
The original position variable x can be next recovered by: $x(t) = \int_0^t y(s)ds + c$ with c s.t. $x(0) = 0$.

2.2 The optimal control problem

The goal is to set the gas pedal in an optimal way to reach a given velocity y_{target} (ms^{-1}).

The objectives could be for examples:

- minimize travel time
- remain within speed limits
- improve vehicle fuel efficiency
- discourage excessive gas pedal adjustments
- or do not accelerate excessively.

These multiple objectives can be implemented by defining objective function(s) and/or defining constraints on the control variable $u(t)$.

Let us discuss the following simple examples.

1. If the only objective is to minimize the travel time, the solution is to use full gas pedal position ($u = 100\%$) for the complete duration of the simulation T.
2. If the objective is to minimize travel time but remain within speed limits, the solution is to reach v_{fin} as fast as possible without overshoot.

Note that each *additional* objective would require to adapt the control to achieve an optimal tradeoff.

The considered objective function In the present practical, given the target function $y_{target}(t)$, we seek to minimize the following objective function:

$$J_\alpha(u; y) = \int (y(t) - y_{target}(t))^2 \, dt + \alpha \int \left(\frac{d^p u}{dt^p}(t) \right)^2 dt \tag{2.3}$$

with $p \in \{0, 1, 2\}$ (value to be experimented).
The cost function is defined as: $j_\alpha(u) = J_\alpha(u; y^u)$ with y^u the state given u.
We seek to solve: $arg\min_u j_\alpha(u)$. This an optimal control problem.

2.3 The numerical strategy

We employ here a basic direct method.

A direct is here well adapted since the direct model is 1D therefore extremely fast to solve.
The total time intervall $[0, T]$ is digitalized with a constant time step: $h = T/N$, $t_i = ih$, $i = 0, \ldots, N$.
The ODE is solved by a standard numerical scheme (eg RK4).

Given the current time index i, the ODE is solved on the sub-intervall centered on the time index i, of length $2P$: $I_P = [i - P, i + P]$.

The control is performed on sub-intervalls centered on i too, of potentially shorter horizon M: on the intervall $I_M = [i - M, i + M]$ with $M \leq P$.

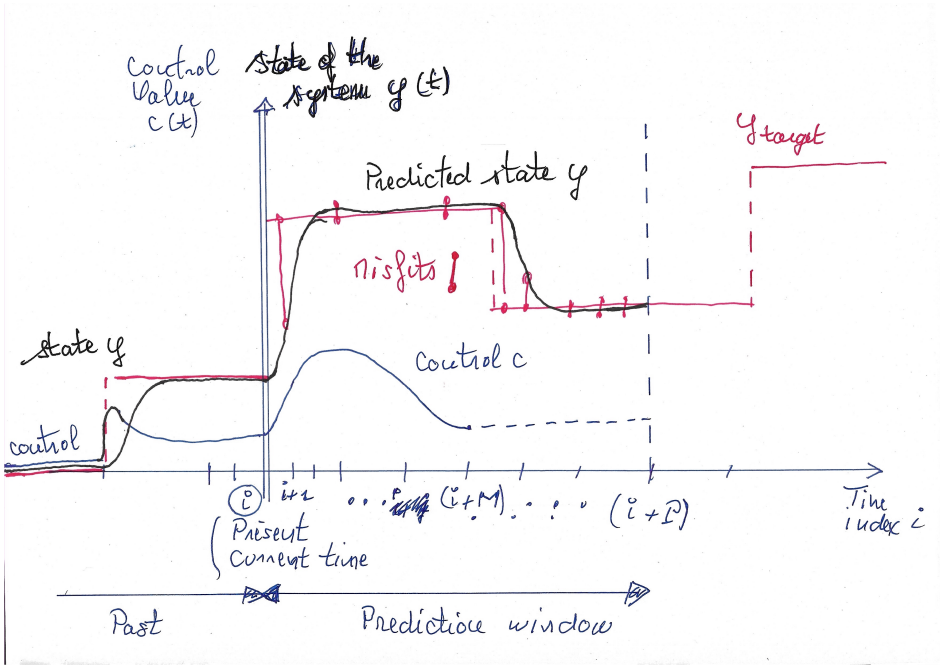

Figure 2.1: The basic direct method to solve the control problem, providing a dynamic control policy.

3 The employed algorithm

The algorithm implemented in the Python code is as follows.

- Initialization. The state $y(0)$ is given (I.C.), the "first guess" of control $u^{(0)}(t)$ is given ($u^{(0)}(t) = 0$ for all t).
- For each time instant i, $i = 1, \ldots, N$,
  - ▷ Setup the current working sub-interval I_P, $I_P = [max(0, i - P), i + P]$, P the prediction horizon.
  - ▷ Given the previous values of control $u^{(i-1)}$,
    - solve the ODE in the time intervall I_P.
      This provides the state $y_l \approx y(t_l)$ at time instants l, $l = i, \ldots, (i + P)$.
    - Evaluate the cost function in the prediction window: $j_\alpha(u_i, \ldots, u_{i+P})$. This provides the first value of j_α (the value before optimization).
  - ▷ Solve the optimization problem:

$$\min_{(u_i,\ldots,u_{i+M})} j_\alpha(u_i, \ldots, u_{i+P}) \tag{3.1}$$

$$\text{under the constraint that the ODE is satisfied} \tag{3.2}$$

  Note. For the indices in $\{i+M+1, \ldots, i+P\}$, the control values are here set to the value at index $(i+M)$ i.e. $u^{(i)}(i + M)$ (that is a constant extraplation of the contol).
  This step provides the control value $u^{(i)}$ in the current working sub-intervall $[t_i, t_{i+P}]$.

  The optimization problem is here simply solved at each time step by using a SQP type method implemented in the Python routine named *minimize.*

- Iterate

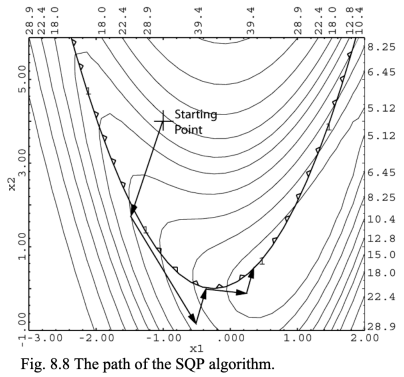

Fig. 8.8 The path of the SQP algorithm.

Figure 3.1: SQP algorithm basic principle. Image extracted from: X

4 Tasks to address

See the complementary document.