AVT | SVT | RWTH Aachen | 52074 Aachen

Systemverfahrenstechnik

Prof. Alexander Mitsos, Ph.D.

Bonus Point Programming Exercise 3

Deadline: February 7, 2021, 24:00 h

Applied Numerical Optimization

Wintersemester 2020/2021

Rules for bonus point exercises

- Please work on the bonus point exercise in groups of 2, 3 or 4 students. If you cannot find a group, use the forum or send an email to optimierung.svt@avt.rwth-aachen.de
- **One member per group** should submit the solution (typically, one or more MATLAB '.m' files) on Moodle before the deadline. The names and enrollment-numbers (or TIM-number, in case no enrollment-number is available) of the group members should be written as comments at the top of the '.m' file.
- Please take care that your code is well-documented (through comments within the source code) and executes out of the box. The results of the third bonus points exercise will be published by February 12, 2021, on RWTHmoodle.

Dynamic Optimization

Introduction

In this programming exercise, you will solve a dynamic optimization problem, more precisely, an optimal control problem. For this exercise, we introduce the following class of optimal control problems:

$$\min_{x(\cdot),u(\cdot)} \Phi(x(\cdot)) = \phi(x(t_f)) \tag{1a}$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), u(t)), \quad x(t_0) = x_0, \quad t \in [t_0, t_f] \tag{1b}$$

$$x_{min} \le x(t) \le x_{max}, \quad \forall t \in [t_0, t_f] \tag{1c}$$

$$u_{\min} \le u(t) \le u_{\max}, \quad \forall t \in [t_0, t_f]. \tag{1d}$$

The state variables $x(t) \in \mathbb{R}^{n_x}$ and control variables $u(t) \in \mathbb{R}^{n_u}$ are time-dependent. The so-called Mayer-type objective functional Φ is defined by the function $\phi : \mathbb{R}^{n_x} \to \mathbb{R}$, that only depends on the state $x(t)$ at the final time t_f. The constraints are ordinary differential equations (1b) and state path constraints (1c). We consider control path constraints in form of simple lower and upper bounds $u_{\min} \in \mathbb{R}^{n_u}$ and $u_{\max} \in \mathbb{R}^{n_u}$ on $u(t)$, respectively. The right hand side of the ordinary differential equation (ODE) is given by the function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. The dimension of the optimization problem is infinite, since for every $t \in [t_0, t_f]$, $u(t)$ and $x(t)$ are optimization variables. For fixed $u(\cdot)$, the state variables $x(t)$, $t \in [t_0, t_f]$ are uniquely determined by the solution of the initial value problem (1b). Thus, the control vector function $u : [t_0, t_f] \to \mathbb{R}^{n_u}$ is the actual (infinite-dimensional) degree of freedom.

Full discretization approach

The so-called full discretization approach discretizes state and control variables, as well as the differential equations. Thus, the original optimal control problem is transformed in a nonlinear program (NLP). We will use the implicit Euler method to discretize the ordinary differential equation (1b) into a set of nonlinear equations. The procedure to obtain a nonlinear program is now described in detail.

The first step is to divide the time horizon $[t_0, t_f]$ into M intervals $[t_{k-1}, t_k]$, $k = 1, \dots, M$ of length h with

$$t_M = t_f, \qquad t_k - t_{k-1} = h,\ k = 1, \dots, M, \qquad h = \frac{t_f - t_0}{M}.$$

The implicit Euler discretization is then

$$x_{k+1} = x_k + h \cdot f(x_{k+1}, u_{k+1}), \quad k = 0, 1, \dots, M-1, \tag{2}$$

where $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$, $k = 1, 2, \dots, M$ are finite dimensional decision variables that approximate the states $x(t)$ and controls $u(t)$, respectively, at the discretization points $t_1, t_2, \dots, t_M$. The optimization variable vector of the full discretization NLP is

$$y = \begin{pmatrix} x_1 \\ u_1 \\ x_2 \\ u_2 \\ \vdots \\ x_M \\ u_M \end{pmatrix} \in \mathbb{R}^{n_y}, \quad \text{where } n_y = M \cdot (n_x + n_u).$$

The full discretization NLP is

$$\min_{y \in \mathbb{R}^{n_y}} \phi(x_M) \tag{3}$$

$$\text{s.t.} \quad c_k(y) = 0, \quad k = 0, 1, \ldots, M-1 \tag{4}$$

$$u_{\min} \leq u_k \leq u_{\max}, \quad k = 1, \ldots, M, \tag{5}$$

where the constraint functions $c_k : \mathbb{R}^{n_y} \to \mathbb{R}^{n_x}$, $k = 0, 1, \ldots, M-1$ are defined by means of (2): $c_k(y) := x_{k+1} - x_k - h \cdot f(x_{k+1}, u_{k+1})$. The state path constraint (1c) can be easily set as bound.

The Van der Pol oscillator

In this exercise, a fixed final time problem of the van der Pol oscillator is considered.

$$\min_{x(\cdot), u(\cdot)} \; x_3(t_f) \tag{6a}$$

$$\text{s.t.} \quad \dot{x}_1(t) = (1 - x_2(t)^2)x_1(t) - x_2(t) + u(t) \tag{6b}$$

$$\dot{x}_2 = x_1(t) \tag{6c}$$

$$\dot{x}_3 = x_1(t)^2 + x_2(t)^2 + u(t)^2 \tag{6d}$$

$$x(0) = [0, 1, 0] \tag{6e}$$

$$-0.4 \leq x_1(t) \quad \forall t \in [t_0, t_f] \tag{6f}$$

$$-0.3 \leq u(t) \leq 1.0 \quad \forall t \in [t_0, t_f] \tag{6g}$$

with $t_f = 5$.

Your task

1. Set up the nonlinear program resulting from applying the full discretization based on the implicit Euler method with different and variable $M \in \{10, 50, 100\}$ for (6).

2. Solve the nonlinear optimization problem by applying the built-in function "fmincon" in MATLAB. The code should be flexible with respect to M. Use the value 0.0 as initial guess for u_k, and the initial conditions as an initial guess for x_k. List the optimal objective function values of the problem for different values of M.

3. In a single shooting (late discretization) approach, would the objective value in general increase or decrease with a finer discretization of the control variables?

Please answer shortly the last question within your code!

Hint

The computed optimal objective function values for $M = 10$, $M = 50$ and $M = 100$ are 2.0611, 2.7558, 2.8535, respectively.