Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-7 (2016) 290–295

Numerical Optimal Control Mixing Collocation with Single Shooting: A Case Study

Anders Albert * **Lars Imsland** * **Joakim Haugen** **

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway (e-mail: {anders.albert, lars.imsland}@ntnu.no).*
** *SINTEF Fisheries and Aquaculture, Trondheim, Norway (e-mail: joakim.haugen@sintef.no)*

Abstract: This paper looks into implementation of numerical optimal control problems of systems with a cascade structure, in which only one part of the dynamic equality constraints has path constraints. We consider two different direct strategies for numerical implementation using direct methods: 1. Collocation for both parts of the cascade. 2. Direct collocation for one part and single shooting for the other. To compare the methods we study the case of iceberg monitoring using a single unmanned aerial vehicle. The study reveals that the second method, under some conditions can be more computationally efficient than the first method.

Keywords: Optimal control, Implementation, UAV, Iceberg Monitoring, Nonlinear programming

1. INTRODUCTION

In this paper, we study implementation of different numerical methods for continuous time optimal control problems (OCPs) formulated as autonomous cascaded nonlinear systems:

$$\min_{u(\cdot)} \int_{t_0}^{t_F} L(p,z,u)dt + E[p(t_F), z(t_F)] \tag{1a}$$

s.t.

$$\dot{p} = f_1(p,z,u), \quad p(t_0) = p_0 \tag{1b}$$

$$\dot{z} = f_2(z,u), \quad z(t_0) = z_0 \tag{1c}$$

$$z_{\min} \le z \le z_{\max} \tag{1d}$$

where $p \in \mathbb{R}^{n_p}$ and $z \in \mathbb{R}^{n_z}$ are state variables. The $u(t) : [t_0, t_F] \to \mathbb{R}^{n_u}$ is the control input. The objective function consist of the *Lagrange term*, $L(p,z,u) : \mathbb{R}^{n_p} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and the *Mayer term*, $E[p(t_F), z(t_F)] : \mathbb{R}^{n_p} \times \mathbb{R}^{n_z} \to \mathbb{R}$. It is solved over a time interval from $[t_0, t_F]$. In addition, we have dynamic equality constraints for both state variables: $f_1(p,z,u) : \mathbb{R}^{n_p} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_p}$ and $f_2(z,u) : \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_z}$. Finally, $z_{\min}$ and $z_{\max}$ are lower and upper limits for the z-state variable. Notice that we only have inequality constraints for one of the state variables.

The dynamic systems we study has a cascaded structure, see e.g. Loria and Panteley (2005) for examples. We choose to call the "outer state" p the system state, and the "inner state" z the actuator state. This naming is for convenience, and need not be consistent with all problems of this form.

Our problem belongs to the field of optimal control theory. Mathematicians like Bellman and Pontryagin developed this field of mathematics during the 1950s (Pesch et al., 2009). A breakthrough in the research of optimal control theory came with the *Pontryagin's maximum principle* (Pontryagin, 1957). This principle states necessary conditions for optimal control problems in continuous time. We can use these conditions to eliminate the controls, u, from the problem and get a boundary value problem, which we can solve numerically. This is referred to as an indirect approach to optimal control. However, an indirect approach suffers from drawbacks like difficulty in initializing the problem (Betts, 2010; Binder et al., 2001). Another approach for solving optimal control problems, which we focus on in this paper, is the direct approach.

In a direct approach, the optimal control problem is first discretized, before the discretized problem is solved. This enables us to transform the optimal control problem to a nonlinear programming problem (NLP). NLPs have well developed solvers, which are efficient even for large problems, at least when they have structure.

1.1 Contribution

In this paper we investigate whether merging the objective function and the system state (the state without path constraints) into a new objective function, can increase computational efficiency. This is based on the premise that reverse algorithmic differentiation is efficient for scalar functions of many variables (Griewank and Walther, 2008). This enables us to exploit the structure and compare different numerical implementation strategies for the new objective and the actuator state.

1.2 Previous Work

The general field of numerical optimal control is a large field, in which single shooting and collocation are standard

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.307

methods. We recommend Betts (2010) and Biegler (2010) as a starting points.

The case we study in this paper is path planning using a mobile sensor for monitoring some objects or targets, where we directly build on the approach used in Haugen and Imsland (2013). Another interesting paper studying the same problem is Walton et al. (2014). Both these papers use collocation in implementing a nonlinear problem for path planning formulated similar as OCP (1). A difference between the two is the solver used; where Haugen and Imsland (2013) uses an interior-point solver as in this paper, while Walton et al. (2014) uses a SQP algorithm.

This paper starts with a formulation of the problem and implementation strategies in Section 2. In Section 3 we go into details for the implementation for the different approaches we use. We present the case we are using for simulation in Section 4. In Section 5 we explain the setup for the simulation. We run and discuss the results in Section 6 before we come with concluding remarks in the final Section 7.

2. PROBLEM FORMULATION AND IMPLEMENTATION STRATEGIES

We want to explore discretization strategies in direct approaches for solving problems on the form of OCP (1) in a computationally efficient manner. There are broadly three discretization approaches: Single shooting, multiple shooting and collocation (Binder et al., 2001). Each of the approaches have their own advantages and disadvantages.

We apply two different strategies for implementation. First, we use collocation for both the system and actuator state. This is the same strategy as used by Haugen and Imsland (2013) and Walton et al. (2014). However, we use a different number of integration steps and degree of the collocation polynomial for the two states. We call this the pure collocation approach. Second, we want to exploit the structure of our problem. We can merge the objective function with the system state into a scalar function using single shooting, for which evaluating the gradient has approximately the same complexity as evaluating the function itself using reverse algorithmic differentiation (Griewank and Walther, 2008). For the actuator state, which contains both inequality and dynamic equality constraints, we apply collocation for easy handling of the inequality constraints. We term this the combined approach with exact Hessian. In addition, we extend the second approach into two additional approaches. Third, we use limited-memory BFGS-update for the Hessian, we term this approach BFGS. This will generally lead to more iterations, but avoid calculating the computationally expensive Hessian. Fourth, we use the Hessian and increase the convergence tolerance for the NLP-solver. We term this approach BFGS−. With this approach we avoid more iterations, but we might get suboptimal solutions.

3. IMPLEMENTATION

For implementation we use Python with CasADi (Andersson, 2013), which is “a symbolic framework for algorithmic differentiation and numeric optimization”. CasADi is open-source and implemented in C++ with Python wrappers. We exploit the CasADi framework to use the NLP-solver IPOPT (Wächter and Biegler, 2006). IPOPT is a primal-dual interior-point NLP-solver. We compile it with the linear algebra sparse direct solver MA57 (HSL, 2015). We chose a interior-point solver over a SQP -solver. The single-shooting approach may fit a SQP-solver better, however we will exploit collocation for all our approaches that leads to huge problems with a sparse structure, for which an interior-point solver in general is a good match. Therefore, we do not include a SQP-solver in our simulations.

We use different strategies to approximate the integral in equation (1a) from problem (1) depending on our chosen implementation.

3.1 Collocation Approach

For the pure collocation approach we approximate the integral (1a) as a sum of states. When using only collocation we have all the states of the state variables available.

$$\min_{u(\cdot),p(\cdot),z(\cdot)} \sum_{n=1}^{N} \Delta t L(p_n, z_n, z_n) + E[p(t_F), z(t_F)] \tag{2a}$$

s.t.

$$\dot{p} = f_1(p, z, u), \quad p(t_0) = p_0 \tag{2b}$$

$$\dot{z} = f_2(z, u), \quad z(t_0) = z_0 \tag{2c}$$

$$z_{\min} \le z \le z_{\max} \tag{2d}$$

Here N is the number of integration steps in approximating the integral. We use collocation for both the state and actuator state.

3.2 Combined Approach

In the combined approaches we embed the system dynamics (the p-dynamics) into the objective, by solving it by means of single shooting. For this type of cascade systems this will always be feasible. To illustrate this, we formulate the optimization problem in the following manner

$$\min_{u(\cdot),z(\cdot)} \; c(z(\cdot), u(\cdot); p_0, t_f) \tag{3a}$$

s. t.

$$\dot{z} = f_2(z, u); \quad z(t_0) = z_0 \tag{3b}$$

$$z_{\min} \le z \le z_{\max} \tag{3c}$$

where the only dynamic constraint is the actuator dynamics (z-dynamics). The function c is a a scalar function obtained by solving the system dynamics (e.g. by single shooting),

$$p(t; z(\cdot), u(\cdot), p_0) = p_0 + \int_{t_0}^{t_f} f_1(p, z, u) dt,$$

and inserting this solution into the objective:

$$c(z(\cdot), u(\cdot); p_0, t_f) = \int_{t_0}^{t_f} L(p, z, u) dt + E[p(t_f), z(t_f)].$$

The actuator dynamics is still discretized using collocation (Biegler, 2010). In our implementation, we use a simple Euler method for the single-shooting discretization of p, and correspondingly a rectangle method for approximating the integral.