

# Collocation Methods in JModelica.org

Lund University  
Faculty of Engineering  
Department of Automatic Control

Fredrik Magnusson

January 31, 2012

Optimization of large-scale dynamic systems, ranging from power plants to vehicle systems is becoming a standard industrial technology. JModelica.org is a tool partly aimed at large-scale optimization, which is done by formulating a *dynamic optimization problem* and then solving it using numerical methods. Numerical methods are iterative methods finding approximate solutions to mathematical problems and are well-suited for being implemented in computers.

In JModelica.org, dynamic optimization problems are described using the Modelica modeling language. Modelica is an equation-based language designed for graphical and textual modeling of complex physical systems. In Modelica, physical systems are modelled using differential algebraic equation systems, i.e. a mixture of differential and algebraic equations. Modelica is used in a wide variety of applications, such as chemistry, mechanics and electronics. Its industrial usage is increasing and it is today used extensively by e.g. automotive companies, such as BMW, Ford and Toyota, and power plant providers, such as ABB and Siemens.

The applications of optimization are many. One is the solution of optimal control problems, where the goal is to find a strategy for controlling a physical system (e.g. a vehicle or power plant) so that it performs a certain action while minimizing the consumed resources (e.g. time or fuel). Another application is the solution of parameter estimation problems, where the goal is to find the values of unknown physical parameters of a system. This problem is formulated as finding the parameter values that allows the model of the physical system to most closely mimic the behavior of the real physical system.

The aim of my thesis has been to implement a new optimization algorithm in the JModelica.org platform, for the solution of dynamic optimization problems described by Modelica code. There are many approaches to solving dynamic optimization problems, which have evolved rapidly during the last 50 years. The approach taken in my thesis is to use *direct collocation methods*.

The main idea of collocation methods is to approximate the solution to an equation system by polynomials. By interpolation, a polynomial of degree  $n$  is uniquely determined if its values in  $n + 1$  distinct points are known. This allows for the

representation of a function using just  $n + 1$  discrete values. This discrete representation makes the solution well-suited to be optimized using numerical optimization methods.

I have also used my implemented algorithm to solve various optimization problems, of both academic and industrial nature, with the purpose of verifying the algorithm's performance. One of the optimization problems I studied is the optimal startup of a combined cycle power plant, where the goal is to minimize the startup duration. A combined cycle power plant is driven by a steam turbine, so the goal is to find the optimal input signal  $u$  to this turbine. The equation system modeling the power plant consists of 138 time-varying variables and equations. The number of discrete optimization variables for the problem constructed by using direct collocation has a number of optimization variables on the order of  $10^5$ .

In order to reduce the wear and tear on the steam turbine, which is one of the most expensive parts of the power plant, the thermal stress in the turbine  $\sigma$  [Pa] may not exceed 260 MPa. The startup process is considered finished when the full load of the system has been reached, which corresponds to  $u = 1$ , and one of the system variables, namely the evaporator pressure  $p$  [Pa], has reached its desired value of  $p_{\text{ref}} \approx 8.35$  MPa. The time profiles of 3 of the 138 system variables found during the optimization are shown in Figure 1.

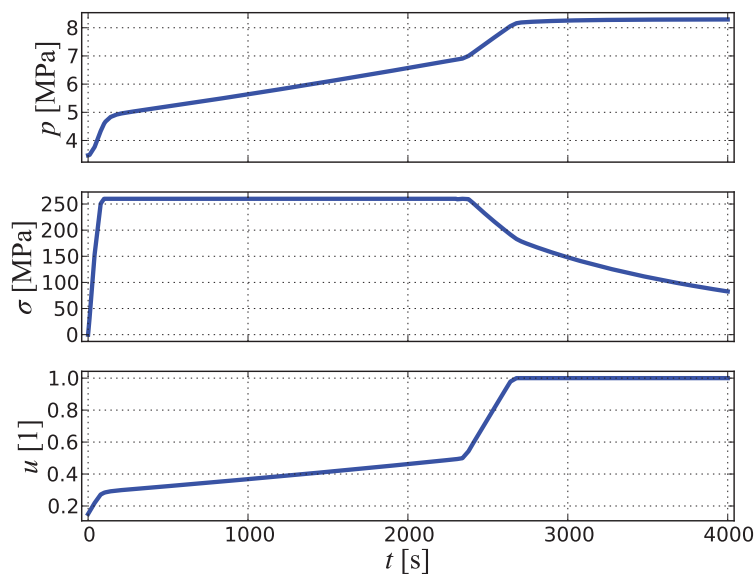


Figure 1: Optimal startup of a combined cycle power plant

I have during my work verified that the implemented algorithm not only produces results consistent with already existing and well-tested algorithms, but also compares favorably to these algorithms in terms of execution times.