Chapter 11
Differential Algebraic Equations

Abstract This chapter documents how to formulate and solve optimization problems with differential and algebraic equations (DAEs). The `pyomo.dae` package allows users to easily incorporate detailed dynamic models within an optimization framework and is flexible enough to represent a wide variety of differential equations. We also demonstrate several automated solution techniques included in `pyomo.dae` that apply a simultaneous discretization approach to solve dynamic optimization problems.

11.1 Introduction

In order to develop a better understanding of real-world phenomena, scientists and engineers often develop dynamic, or differential equation based, models. High fidelity simulation of these models can often be difficult and computationally expensive and is still an active research area in many fields. But after a model suitable for simulation has been developed, the next goal is often to optimize a particular aspect of the dynamic system. (e.g., model parameter estimates given dynamic data, or control of the dynamic system to a desired set point). For example, consider the small optimal control problem from [49]:

\[
\begin{align*}
\text{min} & \quad x_3(t_f) \\
\text{s.t.} & \quad \dot{x}_1 = x_2 \\
& \quad \dot{x}_2 = -x_2 + u \\
& \quad \dot{x}_3 = x_1^2 + x_2^2 + 0.005 \cdot u^2 \\
& \quad x_2 - 8 \cdot (t - 0.5)^2 + 0.5 \leq 0 \\
& \quad x_1(0) = 0, \quad x_2(0) = -1, \quad x_3(0) = 0, \quad t_f = 1
\end{align*}
\] (11.1)

© Springer International Publishing AG 2017
W.E. Hart et al., Pyomo — Optimization Modeling in Python, Springer Optimization and Its Applications 67, DOI 10.1007/978-3-319-58821-6_11
where the objective is to minimize the value of $x_3$ at the final time point by finding the optimal values for the input variable $u$. This problem includes three differential equations as constraints and also includes an inequality constraint restricting the profile of $x_2$, also known as a path constraint.

While it is easy to write down optimization problems including dynamic models, solving them is hard. Off-the-shelf optimization solvers cannot handle differential equations directly. Therefore, optimization problems including differential equations as constraints, or dynamic optimization problems, must be reformulated in order to be solved. Common solution approaches include single or multiple shooting methods or a full discretization approach. Regardless of the solution strategy, the implementation of the technique is often entwined with the particular model or problem being solved which makes it time-consuming to apply these solution techniques to new dynamic optimization problems or experiment with different solution strategies on the same model.

The pyomo.dae package addresses several of these challenges. It provides users the ability to separate the dynamic optimization formulation from the solution strategy used to solve it. This is done by introducing modeling components for representing continuous domains and derivative terms directly. pyomo.dae also includes implementations of the simultaneous discretization solution technique which can be applied automatically to a Pyomo model with differential equations.

The remainder of this chapter provides a brief overview of how to use the pyomo.dae package. We refer the reader to Nicholson et al. [64] for a more detailed description and information about the design and novelty of this package. This package is still under active development and expansion. Please refer to the online Pyomo documentation for the most up-to-date documentation on new features.

### 11.2 Pyomo DAE Modeling Components

In order to represent DAE models in Pyomo, the pyomo.dae package defines two new components:

- the **ContinuousSet** represents continuous domains over which a derivative can be taken, and
- the **DerivativeVar** represents the derivative of a **Var** with respect to a **ContinuousSet**.

The package is explicitly imported to access these modeling components:

```python
from pyomo.environ import *
from pyomo.dae import *
```

The **ContinuousSet** component functions similarly to the regular Pyomo **Set**. It can be used to index other Pyomo components such as **Var**, **Constraint**, or **Expression**. A **ContinuousSet** can be thought of as a bounded virtual set. In order to construct a **ContinuousSet** you must supply numeric values repre-