

# Trajectory Optimization Framework

Johnathan Van Why

September 10, 2013

## 1 Introduction

This framework is designed for the optimization of trajectories for robotic systems. It can handle underactuated and hybrid systems. In the future, it will likely have support for multiple "phases", or time periods in which specific dynamics are enforced.

The framework makes use of the MATLAB Symbolic Toolbox to aid in problem formulation and manipulation. The actual optimization, however, is almost always numerical. By keeping the problem in a symbolic form, we allow for the implementation of a mode where the optimization algorithm and problem are formulated together, leading to a final solver optimized for the specific problem's structure.

This document gives a basic description of how to use the framework, which will eventually be supplemented by code samples (there will soon be a sample program for the reworked interface in the repository). It also covers the architectural design of the framework and details on some of the most important algorithms contained within the framework.

## 2 Usage

At the moment, the interface to the optimizer is undergoing a re-design, so it is likely to change in the near future. This section will hopefully be kept up to date with changes.

One thing that currently appears to be invariant is the usage of a central structure, called the "scenario" structure. The scenario structure contains all information necessary to run an optimization – it is incrementally filled out by the optimizer as it completes the various stages of problem setup and optimization. The final scenario structure contains the results of the optimization, and should contain all information that may need to be saved for future use.

The overall workflow for using the optimizer is as follows:

- Do basic problem specification in the scenario
- Run the scenario through `traj_setup()`, which automatically does most setup.
- Call `traj_run_opt()` to do the optimization.