Tropidance : Trajectory Optimization and Guidance

Sparsh Yadav

September 16, 2020

Contents

1 Introduction

Tropidance is an optimal control toolbox that enables the user to solve challenging trajectory optimization and guidance problems with ease, using direct optimal control techniques. With its ease of use, engineering teams can now get the results to complex optimal control problems rapidly. The potential for the software is limitless and can be used across multiple industries, including aerospace, robotics, energy, mobility and many more.

Historically, the techniques for solving optimal control problems are broadly classified into direct and indirect methods. Indirect methods rely on the first-order necessary conditions and Pontryagin's minimum principle. These methods lead to a two-point boundary value problem and try to find the optimal trajectories for the state and the control. Although indirect methods enjoy fewer dimensions in the search space, they suffer from extreme sensitivities to the search parameter. The extraordinary speed of modern computers has opened new opportunities for indirect methods for real-time applications. Direct methods transform the optimal control problem into a non-linear constrained optimization problem by discretization of the control input and the system dynamics at finite grid points. Model Predictive Static Programming (MPSP) is one such direct optimal technique.

Optimal control is particularly relevant in Aerospace applications where the economic cost of the mission can be significantly reduced by using optimal control algorithms. The importance of optimal control can be gauged by the fact that, in 2007, using pseudo-spectral optimal control technique NASA saved 1 million dollars to manoeuvre the International Space Station. It should be noted, the control trajectory was computed offline and uploaded to the International Space station. Real-time optimal control is still in its nascent stages. To fill the lacuna between offline and real-time applications of optimal control, we need efficient algorithms and faster processors. MPSP has been shown to be computationally efficient for several missile and spacecraft guidance problems.

2 Tropidance

Tropidance builds upon a decade of research into optimal control algorithms for real-time applications. The motivation of the package is to provide researchers, labs, and industry alike, an easy to use, optimal control and guid-

ance software. Anyone with a basic understanding of optimal control should be able to use it without any knowledge of underlying algorithms. The package includes computationally efficient algorithms with mathematically proven convergence properties fit for real-time applications. The software is written in Python, one of the world's most popular programming language, currently implements several state-of-the-art computationally efficient MPSP algorithms developed at ASL, Aerospace Engineering, Indian Institute of Science. The fact it is written in Python makes it platform agnostic, as long as the python interpreter is available, you can get started in less than no time. Tropidance is accompanied with detailed user and developer documentation. The software follows a modular approach and can be extended to implement any number of algorithms.

In summary, Tropidance has the following advantages to the software currently available. Firstly, It implements state-of-the-art MPSP (Model Predictive Static Programming) techniques, which are computationally efficient. Moreover, with several variants of MPSP now available it can address a broad class of optimal control problems. Secondly, It is implemented in Python, an open-source language. Users can quickly get started with the examples provided with the software package with little effort. Thirdly, Tropidance uses open-source libraries and doesn't have a dependency on expensive software like MATLAB, making it more affordable. Finally, the software is built using a modular structure and flexibility in mind; it can be very easily extended to include new algorithms as and when required. The developer documents describe how that can be done.

Numerically, MPSP has the following advantages for solving optimal control problems:

1. A dynamic programming problem is converted to a static programming problem, and thus it requires only a static costate vector for the control update

2. The symbolic costate vector has a closed-form solution for unconstrained problems, which significantly reduces the computational load.

3. The sensitivity matrices that are necessary to compute the static costate vector are calculated recursively.

3 Class of problems

Building on a decade of research, Tropidance implements several MPSP algorithms that have stood the test of time. These algorithms have been extended for several use cases over the years.

3.1 Fixed Final time t_f with functions of state specified at final time.

It includes problems in which the functions of the state variables $\mathbf{Y} = \mathbf{H}(\mathbf{X})$ are prescribed at a fixed terminal time t_f. We have system dynmaics given by $\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}, \mathbf{U})$ and we want that as $t \to t_f$, $\mathbf{Y} \to \mathbf{Y}^*$. Several algorithms that can be used to solve such problems include:

MPSP

MPSP combines the philosophy of non-linear model predictive control and approximate dynamic programming. The basic version of MPSP is applicable for finite-horizon non-linear problems with terminal constraints was presented in [1]. It brings the concept of trajectory optimization into guidance laws which are optimal in regards to the control effort required.

MPSP technique is computationally efficient and can be implemented online, providing for a wide range of real-time applications. The effectiveness of the method is demonstrated for the ascent phase guidance of a ballistic missile using solid motors in [1].

GMPSP

A generalized version of MPSP was presented in [2] that extends upon the MPSP philosophy for solving a class of finite-horizon non-linear optimal control problems with a hard terminal constraint. Two key features for its high computational efficiency include one-time backward integration of a small-dimensional weighting matrix dynamics, followed by a static optimization formulation that requires only a static Lagrange multiplier to update the control history. Moreover, it turns out that under Euler integration and rectangular approximation of finite integrals it is equivalent to model predictive static programming technique [1].

QS-MPSP

Quasi-spectral MPSP was proposed in [3] where the control variable is expressed as a weighted sum of basis functions. The coefficients of the basis functions are then determined using optimization techniques, in contrast, to control input at every grid point. This reduces the complexity of the problem and also ensures the smoothness of the control variable. Such a procedure also provides the spread of control over the available time to go.

QS-MPSP guidance was demonstrated in [3] by formulating and solving an angle-constrained missile guidance problem to successfully engage an incoming high-speed ballistic missile target in three dimensions. The proposed QS-MPSP guidance showed considerable improvement in the lateral acceleration demand over the BPN guidance scheme.

3.2 Tracking Problems

Trajectory tracking problems can also be solved using the recently developed TMPSP technique.

T-MPSP

A non-linear optimal command tracking technique was presented in [4] called the Tracking-oriented MPSP. In this technique, like MPC a model-based prediction-correction approach is adopted.

In [4] TMPSP was used for trajectory tracking problem of a two-wheel differential drive mobile robot both in simulations and on real hardware. Successful results in both simulation and hardware implementation study demonstrated that the proposed, computationally efficient, T-MPSP algorithm can be implemented online and is promising for controlling dynamic systems under the paradigm of fast non-linear MPC.

3.3 Problems with uncertainties in the system parameters

There are several scenarios where the exact current state of the system can not be deteremined or there are several uncertainities in the system model. In such scenarios recently developed, U-MPSP technique can be used.

U-MPSP

Unscented MPSP was presented in [5], which applies to a class of problems where there are uncertainties in the time-invariant system parameters and/or initial conditions. This technique is a fusion of two recent ideas, namely MPSP and Riemann–Stieltjes optimal control problems. In UMPSP, first an unscented transform is utilized to construct a low-dimensional finite number of deterministic problems. The philosophy of MPSP is utilized next so that the solution can be obtained in a computationally efficient manner. The control solution not only ensures that the terminal constraint is met accurately with respect to the mean value, but it also ensures that the associated covariance matrix (i.e., the error ball) is minimized.

Significance of U-MPSP has been demonstrated in [5] by successfully solving two benchmark problems, namely the Zermelo problem and inverted pendulum problem, which contain parametric and initial condition uncertainties.

4 Applications

Numerous complex engineering problems can be formulated as optimal control or guidance problems, be it guiding a rocket to its desired orbit or optimal guidance for a swarm of drones. Over a decade, the algorithm's feasibility and efficiency have been demonstrated on several benchmarks and engineering problems ranging from biomedical to aerospace applications.

- Aerospace : Ascent guidance for rockets, interceptors, and missiles [1, 6, 3, 7, 8]; Landing guidance for terrestrial (RLVs) and extra-terrestrial missions [9, 10, 11]; flexible final time guidance [12].
- Robotics: Trajectory tracking for robotics [4] ; Inverted Pendulum [5]; Unmanned Aerial Systems; Energy sector.
- Process Control: Biomedical; Industrial Processes etc.

5 Software Architecture

The aim of the software is usability and extensibility; these principles have been the guiding lights for the development. The software follows a modular

approach, with APIs provided to set up and solve the problem. Not only is it user-friendly, but also care has been taken that the software is developer-friendly, and new functionality can be added to the code rapidly.

Tropidance follows a layered architectural pattern with three main layers, namely:

- Client Layer: Contains several client interfaces.
- Interface Layer: Provides an abstraction over the core layer so different clients can communicate to the core functionality
- Core Layer: Implementation of Algorithms (MPSPS, GMPSP, etc.) and utilities like logging, plotting.

The overall block diagram of the package is shown in Figure. 1.

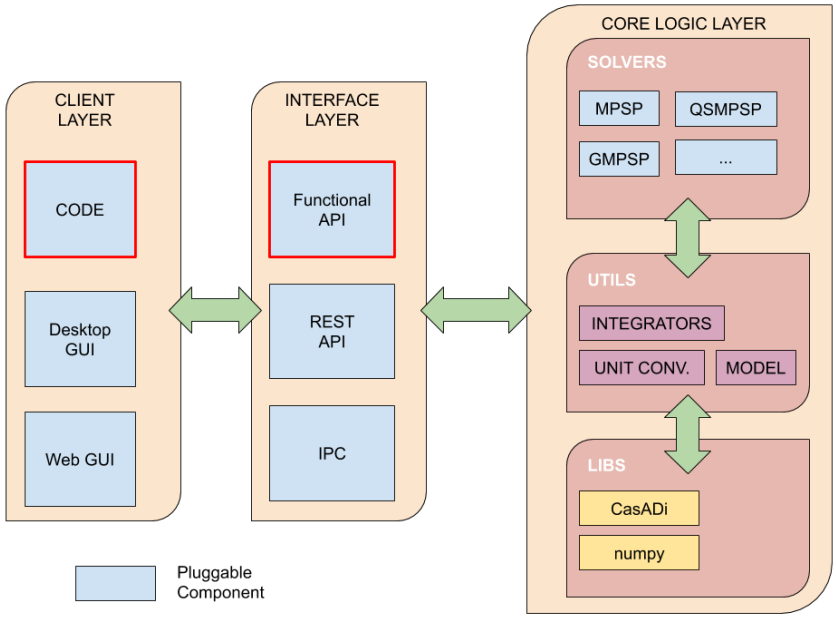

Figure 1: Block diagram of the software package.

5.1 Python Package

Tropidance is implemented in Python because of its widespread adoption and ease of use. Moreover, this language is extremely popular with academia

and the industry as it a high-level language that enables rapid testing and deployment. The language is also open-source and has a rich collection of scientific libraries.

5.2 API

The package provides an easy to use front-end API, which can be used to describe the system dynamics, configure the algorithm and start the solver.

5.3 Salient features of Tropidance

The software package has several advantages over existing packages which are enumerated below:

- Configuring the system model is straight-forward.
- It is easy to configure a particular algorithm and its parameters.
- A simple front-end API for the user, that can be used for coding or to create a GUI.
- Comes with a logging module that provides verbose and descriptive messages to guide the user and help with debugging.
- Uses the principles of OOP (Object-oriented programming).
- Requires no knowledge of underlying algorithms.
- The aim is to write extensive tests for regression testing.
- Thorough documentation autogenerated with pydoc.
- Source and version control using git.

References

[1] Radhakant Padhi and Mangal Kothari. Model predictive static programming: a computationally efficient technique for suboptimal control design. *International journal of innovative computing, information and control*, 5(2):399–411, 2009.

[2] Arnab Maity, Harshal B Oza, and Radhakant Padhi. Generalized model predictive static programming and angle-constrained guidance of air-to-ground missiles. *Journal of Guidance, Control, and Dynamics*, 37(6):1897–1913, 2014.

[3] Sabyasachi Mondal and Radhakant Padhi. Angle-constrained terminal guidance using quasi-spectral model predictive static programming. *Journal of Guidance, Control, and Dynamics*, 41(3):783–791, 2018.

[4] Prem Kumar, B Bhavya Anoohya, and Radhakant Padhi. Model predictive static programming for optimal command tracking: A fast model predictive control paradigm. *Journal of Dynamic Systems, Measurement, and Control*, 141(2), 2019.

[5] S Mathavaraj and Radhakant Padhi. Unscented mpsp for optimal control of a class of uncertain nonlinear dynamic systems. *Journal of Dynamic Systems, Measurement, and Control*, 141(6), 2019.

[6] Prasiddha Nath Dwivedi, Abhijit Bhattacharya, and Radhakant Padhi. Suboptimal midcourse guidance of interceptors for high-speed targets with alignment angle constraint. *Journal of Guidance, Control, and Dynamics*, 34(3):860–877, 2011.

[7] Harshal B Oza and Radhakant Padhi. Impact-angle-constrained suboptimal model predictive static programming guidance of air-to-ground missiles. *Journal of Guidance, Control, and Dynamics*, 35(1):153–164, 2012.

[8] Mangal Kothari and Radhakant Padhi. A nonlinear suboptimal robust guidance scheme for long range flight vehicles with solid motors. *Automatic Control in Aerospace*, 3(1), 2010.

[9] Charu Chawla, Pranjit Sarmah, and Radhakant Padhi. Suboptimal reentry guidance of a reusable launch vehicle using pitch plane maneuver. *Aerospace Science and Technology*, 14(6):377–386, 2010.

[10] Kapil Sachan and Radhakant Padhi. Waypoint constrained multi-phase optimal guidance of spacecraft for soft lunar landing. *Unmanned Systems*, 07(02):83–104, April 2019.

[11] Omkar Halbe, Ramsingh G Raja, and Radhakant Padhi. Robust reentry guidance of a reusable launch vehicle using model predictive static programming. *Journal of Guidance, Control, and Dynamics*, 37(1):134–148, 2014.

[12] Arnab Maity, Radhakant Padhi, Sanjeev Mallaram, G Mallikarjuna Rao, and M Manickavasagam. A robust and high precision optimal explicit guidance scheme for solid motor propelled launch vehicles with thrust and drag uncertainty. *International Journal of Systems Science*, 47(13):3078–3097, 2016.