A multipurpose, easy-to-use Model Predictive Control design and simulation code

Marco Vaccari* Gabriele Pannocchia†

We present a multipurpose, easy-to-use code for Model Predictive Control (MPC) design, analysis and simulation. The major goal of this code is to provide the user with a general, versatile MPC framework that can be adapted to problems in different areas, e.g. process control, robotics, aerospace. Many different MPC simulation problems can be easily studied with this code, given the following features:

- The MPC model can be linear or non-linear, discrete and continuous time.
- The same possibilities apply to the controlled process, which can also be selected equal to the MPC model for nominal simulation cases.
- The finite-horizon cost function can be quadratic as in standard (tracking) MPC problems or any (linear or nonlinear) function, e.g. to consider economic MPC problems. In addition, it can be defined as the sum of discrete-time stage cost terms or the integral of a continuous-time running cost.
- A target calculation module is also available, with simple quadratic or generic cost function.
- State feedback or output feedback can be considered, and in the latter case several choice of linear and nonlinear observers are made available.
- In all cases the user can activate the "offset-free" design option, to implement a linear or non-linear disturbance model, and an associated augmented observer.

The code has been implemented in Python 2.7, using CasADi 3.1, and its architecture is rather simple: the basic idea is to have a main file containing all the MPC core functions. Everything else, such as model or cost functions, steady-state and dynamic optimization problems, state estimation, is made in separate Python scritpts. In this way the architecture is somehow modular and easier to develop and to further customize.

For each case to simulate, the user has to fill in a single Python script with all the required information such as matrices or functions describing the state dynamics and cost terms, MPC horizon, simulation horizon, time step, etc. Several boolean variables are provided to help the user with the problem definition, e.g. state-feedback, nominal case, continuous cost function. Finally, there is also the possibility to specify which offset-free disturbance model applies by simply changing a name tag.

During this talk, the main features and problems that can be solved with this code will be thoroughly illustrated, discussing the problem setup and methodologies, the coding choices. In addition, the results obtained on several test cases taken from different areas will be presented. Finally, we will illustrate the future research and development directions, which may include: implementing a moving horizon estimation, implementing a modifiers adaptation based technique for the economic MPC case, an object oriented programming of the code.

*University of Pisa, Department of Civil and Industrial Engineering, Largo Lucio Lazzarino 2, 56126 Pisa (Italy), marco.vaccari@ing.unipi.it

†University of Pisa, Department of Civil and Industrial Engineering, Largo Lucio Lazzarino 2, 56126 Pisa (Italy), gabriele.pannocchia@unipi.it