**MPC Implementation**

**Simulink and Code Generation**

There are several options for Simulink and code generation.

***An official from Matlab***

* Model Predictive Control Toolbox™ software provides code generation functionality for controllers designed in Simulink and MATLAB. More information can be found in the MATLAB website with the reference link below.

<https://nl.mathworks.com/help/mpc/code-generation.html>

* There are also a third party add on which can be downloaded from MATLAB exchange file from NXP (NXP Support Package MPC57xx). An instruction on how to download and install can be seen in the reference link.

<https://nl.mathworks.com/matlabcentral/fileexchange/70327-nxp-support-package-mpc57xx>

**Python**

# ***do-mpc***

* The do-mpc software is Python based and works therefore on any OS with a Python 2.7 distribution.
* do-mpc is a set of Python scripts for the easy, efficient and modular implementation of Nonlinear Model Predictive Control (NMPC) and in particular of Robust NMPC methods. do-mpc divides the implementation of an NMPC controller into four modules (*model, optimizer, observer and simulator*). With this classification the implementation of a new NMPC requires only the redefinition of the corresponding module. In addition we provide automatic design of a multi-stage NMPC algorithm to consider explicitly uncertainty in the model.
* do-mpc strongly relies on third-party software. Especially the whole code is based on the use of [CasADi](https://github.com/casadi/casadi/wiki) for the formulation of the problem and on [Ipopt](https://projects.coin-or.org/Ipopt) for the solution of the resulting nonlinear programming problems.

<https://github.com/do-mpc/do-mpc/wiki>

***Casadi***

* CasADi is an open-source tool for nonlinear optimization and algorithmic differentiation. Casadi is available for Python, Matlab and Octave

<https://web.casadi.org/>

# ***ACADO Toolkit***

* ACADO Toolkit is a software environment and algorithm collection for automatic control and dynamic optimization. It provides a general framework for using a great variety of algorithms for direct optimal control, including model predictive control, state and parameter estimation and robust optimization. ACADO Toolkit is implemented as self-contained C++ code and comes along with user-friendly MATLAB interface.
* The tool allows to export optimized, highly efficient C-code to solve *nonlinear model predictive control (NMPC)* .

<http://acado.github.io/index.html>

**Others:**

**MPCpy** : A group of classes to run Model Predictive Control (MPC) simulations using python.

<https://pypi.org/project/mpcpy/>

**Operator Splitting Quadratic Program**

The OSQP (Operator Splitting Quadratic Program) solver is a numerical optimization package for solving convex quadratic programs

<https://osqp.org/docs/index.html>

# **PyAdvancedControl**

Python Codes for Advanced Control

<https://github.com/AtsushiSakai/PyAdvancedControl>

**References:**

**Simulink Code generation:**

**Ofiicial Matlab**

<https://nl.mathworks.com/help/mpc/code-generation.html>

**NXP model base design**

<https://nl.mathworks.com/matlabcentral/fileexchange/70327-nxp-support-package-mpc57xx>

# **do-mpc**

<https://github.com/do-mpc/do-mpc/wiki>

**Casadi**

<https://web.casadi.org/>

# **ACADO Toolkit**

<http://acado.github.io/index.html>

Example:

## Real Time Embedded Target Application of MPC

https://nl.mathworks.com/matlabcentral/fileexchange/67401-real-time-embedded-target-application-of-mpc

<http://cvxr.com/cvx/>

<https://projects.coin-or.org/qpOASES>

<http://pages.cs.wisc.edu/~swright/ooqp/>

<https://www.uiam.sk/pc11/data/workshops/mpc/MPC_PC11_Lecture4.pdf>

<https://www.uiam.sk/pc11/data/workshops/mpc/MPC_PC11_Lecture4.pdf>