1. Simulation Tool

The simulations of the previous study were mainly designed for an specific choice of modes and a specific trajectory to follow. In this study, we wanted to study general properties of the FOM problem, to generalize the results to more complex pusher-slider systems and trajectories, and to compare our different FOM strategies to solve the MPC problem with other conventional methods as MIQP. In order to satisfy this needs, we decided to implement a general simulation tool. The code is added in appendix **??** but the main structure is explained here.

An abstract state interface is implemented to provide the dynamics and cone constraints of the state as well as their linearized versions at a certain objective state space and control input. To define a certain expimental problem, the dynamics function and constraint equations are derived depending on a general state space variable **x** and controller action **u** and the necessary state instances are imlemented and a mode is then defined by a vector of state handles. An MPC solver class it's also implemented to get an initial condition and an objective trajectory and return a control input. Each MPC solver implementation can include its own member states or modes (as well as other optimization setup variables) and internally decides how to handle them.

At the same time, an Euler integration simulator is implemented. This simulator uses it's own state classes to describe the dynamics of the pusher slider system. These states are independent from the optimization ones so that discrepancies between model and reality can be simulated (for example the MPC solver can use linearized dynamics or cone constraints while the Euler integrator uses the nonlinearized version, which is more simililar to the real experimental setup than using linearized dynamics in both places). The Euler integration class also holds a member MPC solver that returns the control input to be applied. The integrator then computes the value of the state variable derivative $\dot{x}$ and uses a simple quadrature to approximate the next state variable value.

1.1 Global Cost Function

During the previous work at MCubelab, there was no systematic and objective way to evaluate the global trajectory resulting from the FOM control law, the exit or failure of an experiment was mainly determined in a subjective binary way. If the pusher slider didnt show appreciable divergence (at the naked eye) from the planned trajectory it was considered a success, otherwise it was a failure. In a similar fashion if the system was perturbed (by an external interaction), the success lied in recovering the original trajectory in a reasonable time determined subjectively. The main goal of the study was just to demonstrate that FOM could effectively control the pusher slider system to track previously planned trajectories, and that all this process could be done online. Because of this proof-of-concept nature, this approach was sufficient.

In this thesis we want to optimize the performance of the FOM and this requires a certain notion of metric or score for the result obtained. We decided to ignore the cost of the controller and we adopted a quadratic cost on the state variables. To keep consistent with the local problem we used the same cost matrix Q as in the MPC obtaining $S = \sum_{i=0}^{N} x_i^T Q x_i$.