The LQR controller, like any other unconstrained control techniques, enters into “panic mode” and generates an abruptly changing control signal.

The RHC controller handles the constraints correctly and provides a smooth transition from the saturated states.

The ability to generate control signals that are close to the maximal stroke allows utilizing the full potential of the mirror. In some cases, it may eliminate the need for the another mirror.

**Dynamics models, actuators coupling models, and constraints types used in Numerical Simulations**

The model of a low-dimension weakly-coupled system is considered (a DM with up to $10 \times 10$ actuators whose dynamical interactions can be described by first-order transfer functions).

**Motivation: Why consider the constrained Receding Horizon Control (RHC)?**

Handling the constraints on the control input is shown in Fig. 1, which are limited to $u \in [-100, 100]$.

- The LQR controller, like any other unconstrained control techniques, enters into “panic mode” and generates an abruptly changing control signal.
- The RHC controller handles the constraints correctly and provides a smooth transition from the saturated states.

**Feasibility of Receding Horizon Control and Quadratic Programming: case of no coupling between actuators**

Unconstrained case $(-\rightarrow)$ is the fastest, QP computational time is below 1 ms even for $10 \times 10$ case.

Loose constraints $(-\rightarrow-)$ constraints are reached 5% of time, the QP takes 0.5 ms for $7 \times 7$ DM but grows up to 1.7 ms for $10 \times 10$ actuators DM.

Tight constraints $(-\rightarrow-)$ constraints are reached 20% of time, the QP takes 1.3 ms for $7 \times 7$ actuators but grows fast up to 6.8 ms for DM with $10 \times 10$ actuators.

**Feasibility of Receding Horizon Control and Quadratic Programming: case of nearest neighbour actuators coupling**

Unconstrained case $(-\rightarrow)$ is the fastest, QP computational time is below 1 ms even for $10 \times 10$ actuators DM case.

Loose constraints $(-\rightarrow-)$ constraints are reached 5% of time, the QP takes 0.5 ms for $7 \times 7$ DM but grows up to 3.3 ms for $10 \times 10$ actuators DM.

Tight constraints $(-\rightarrow-)$ constraints are reached 20% of time, the QP takes 1.4 ms for $7 \times 7$ actuators but grows fast up to 7 ms for DM with $10 \times 10$ actuators.

**Feasibility of Receding Horizon Control and Quadratic Programming: case of nearest and diagonally adjacent coupling**

Unconstrained case $(-\rightarrow)$ is the fastest, QP computational time is below 1 ms even for $10 \times 10$ actuators DM case.

Loose constraints $(-\rightarrow-)$ constraints are reached 5% of time, the QP takes 0.5 ms for $7 \times 7$ DM but grows up to 3 ms for $10 \times 10$ actuators DM.

Tight constraints $(-\rightarrow-)$ constraints are reached 20% of time, the QP takes 1.5 ms for $7 \times 7$ actuators but grows very fast up to 13 ms for DM with $10 \times 10$ actuators.

**Discussion of Numerical Simulations and Conclusions**

Unconstrained case $(-\rightarrow)$ even for the largest case of $10 \times 10$ actuators, the time is below 1 ms and grows almost linearly with the size of the problem.

Loose constraints $(-\rightarrow-)$: computation time for the nearest neighbour coupling and the case diagonally adjacent coupling are similar (about 1 ms) for a moderate number of actuators (less than $8 \times 8$) but increases up to 3 ms for $10 \times 10$ actuators.

Tight constraints $(-\rightarrow-)$: the decoupled and coupled cases are almost the same in computational time. The case of coupling between diagonally adjacent actuators is the most diffi- cult: that the growth rate of the computational time accelerates quickly with the number of actuators up to 13 ms for DM with $10 \times 10$ actuators.