Design Review for Model Predictive Controller "mpcobj"

Summary of Performed Tests

Test	Status
MPC Object Creation	**Pass**
QP Hessian Matrix Validity	**Warning**
Closed-Loop Internal Stability	**Warning**
Closed-Loop Nominal Stability	**Fail**
Closed-Loop Steady-State Gains	**Warning**
Hard MV Constraints	**Pass**
Other Hard Constraints	**Pass**
Soft Constraints	**Pass**
Memory Size for MPC Data	**Pass**

Individual Test Result

MPC Object Creation

Tests whether your specifications generate a valid MPC object. If not, the review terminates.

The MPC object is OK. Testing can proceed.

Return to list of tests

QP Hessian Matrix Validity

The MPC quadratic programming (QP) problem must have a unique solution. You must choose cost function parameters (penalty weights) and horizons such that the QP's Hessian matrix is positive-definite.

Penalty Weights On Manipulated Variable Rates

You can guarantee a positive-definite Hessian by increasing penalties on manipulated variable increments (MV rates). A potential disadvantage is a more sluggish controller response.

The following table lists your controller's minimum MPCobj.Weight.MVrate parameter for each MV.

MV	Weights.MVRate
MV1	0.1
MV2	0.1

Finding: no problems detected.

Penalty Weights On Output Variables

Your output variable (OV) penalty weights also affect the Hessian. Non-zero values emphasize the importance of OV target tracking, making a unique QP solution more likely.

The following table lists the minimum weight for each OV along the prediction horizon.

OV	Weights.OV
MO1	1
MO2	1
MO3	0
MO4	0

Warning: at least one OV weight is zero or very small.

Horizon Parameters For Plants With Delay

If your plant includes delay T, the prediction and control horizons (P and M) must be set appropriately The last MV movement in the optimal control sequence occurs at step M. In order for it to affect the OVs within the prediction horizon, the following condition should be satisfied: P - M > T. Otherwise, the last MV movement is irrelevant to OV control and the QP might not have a unique solution.

The maximum total delay in your prediction model is T = 0 sampling periods, and P - M = 11.

Finding: no problems detected.

Scale Factors

Scaling converts the relationship between output variables and manipulated variables to dimensionless form. Scale factor specifications can improve QP numerical accuracy. They also make it easier to specify tuning weight magnitudes.

In order for the outputs to be controllable, each must respond to at least one manipulated variable within the prediction horizon. If the plant is well scaled, the maximum absolute value of such responses should be of order unity.

Outputs whose maximum absolute scaled responses are outside the range [0.1,10] appear below. The table shows the maximum absolute response of each such OV with respect to each MV.

	MV1	MV2
MO1	164414	8939.93
MO2	25111.1	5155.95
MO3	2550.19	234.633
MO4	33115.1	43180.1

Warning: at least one output variable response indicates poor scaling. Consider adjusting MV and OV ScaleFactors.

Return to list of tests

Closed-Loop Internal Stability

Certain prediction model and controller settings combinations result in an internally-unstable feedback system that will perform poorly in practice due to prediction error. If your plant is stable, your feedback system is internally stable if all controller modes are exponentially stable or pure integrators.

This test extracts the A matrix from the unconstrained controller's state space realization, and then calculates its eigenvalues. If the absolute value of each eigenvalue is 1 or less and the plant is stable, your feedback system is internally stable.

Finding: your controller includes unstable modes but since your plant is also unstable, an unstable controller might be needed. If your closed-loop system is nominally stable (see the next test), you can ignore this warning.

Return to list of tests

Closed-Loop Nominal Stability

The feedback connection of the plant and controller should be stable for the nominal case: when the controller's prediction model represents the plant perfectly and no constraints are active.

This test obtains the discrete-time state-space realization of the closed-loop system -- the plant and controller connected in a feedback configuration. It extracts the A matrix from this and calculates its eigenvalues. If the absolute value of each such eigenvalue is 1 or less, the nominal (unconstrained) system is stable.

Error: your nominal (unconstrained) closed-loop system is unstable. The absolute value of the maximum eigenvalue exceeds unity by 0.024658.

If your controller is internally unstable, that is the likely cause. See the previous test.

Another likely cause is a state estimator that is inappropriate for your plant. If you are not using the default estimator, review your estimator design.

Return to list of tests

Closed-Loop Steady-State Gains

`cloffset` is used to determine whether the controller forces all controlled output variables to their targets at steady state, in the absence of constraints.

The command calculates the impact of a sustained disturbance on each measured output variable (OV) in terms of an input/output gain. If a gain is zero, the controller eliminates steady-state tracking error for that disturbance-to-output mapping.

The gains with magnitudes exceeding 1e-05 are as follows:

Disturbed OV	Affected OV	Gain
MO1	MO3	0.0292811
MO2	MO3	0.235875
MO3	MO3	1
MO1	MO4	-0.016778
MO2	MO4	-0.123658
MO4	MO4	1

Warning: your application has 4 controlled outputs and 2 manipulated variables. This prevents elimination of steady-state tracking error in all controlled outputs.

You can decrease the amount of error in a given output by making its Weights.OV value relatively large. You can also sacrifice an output by setting its Weights.OV value to zero, which should reduce errors in the other outputs.

Return to list of tests

Hard MV Constraints

The controller should always satisfy hard bounds on a manipulated variable *OR* its rate-of-change. If you specify both constraint types simultaneously, however, they might conflict during real-time use.

For example, if an event pushes an MV outside a specified hard bound and the hard MV rate bounds are too small, the resulting QP will be *infeasible*.

Avoid such conflicts by specifying hard MV bounds *OR* hard MV rate bounds, but not both. Or if you want to specify both, soften the lower-priority constraint by setting its ECR to a value greater than zero.

Finding: no constraints detected.

Return to list of tests

Other Hard Constraints

It can be impossible for your controller to satisfy all its hard constraints under all conditions. If so, the controller will declare the optimal control (QP) problem to be *infeasible*, an error condition.

The manipulated variable rates (MVrates) are the QP decision variables, so their bounds are always feasible. Other variables, such as outputs (OVs), *depend* upon the MVrates. Unusual events might make OV bounds impossible to satisfy. Hard custom constraints can also become infeasible.

Finding: no constraints detected.

Return to list of tests

Soft Constraints

ECR Parameters

This test evaluates the constraint ECR parameters to help you achieve the proper balance of using hard and soft constraints. If a constraint is too soft, an unacceptable violation may occur. If it is too hard, the controller might pay it too much attention. Moreover, making a constraint harder cannot prevent a violation if the constraint is fundamentally infeasible.

Impact of delays

Delays can make it impossible to satisfy output constraints. The presence of unattainable constraints usually degrades performance. Let j be the location (within the prediction horizon) of the first finite constraint value (Min or Max) for OV(i). If all delays for OV(i) exceed j, the constraint is unattainable.

Finding: no constraints detected.

Return to list of tests

Memory Size for MPC Data

This test provides an estimation of the memory size required by the MPC controller at the run time. We assume a scalar value takes 4 bytes in single precision and 8 bytes in double precision.

The table below estimates how much physical memory, for example RAM on board, is needed to store the matrices used in online optimization. The value depends on the MPC controller settings such as horizons, plant order, plant size and the number of constraints. If the physical memory size of your hardware is less than the estimated data memory requirements of the controller, you can run out of memory when you deploy the controller. Redesign the controller to reduce its memory requirements by using shorter horizons, reducing the plant, or reducing the constraints. Alternatively, you can increase the available physical memory.

The estimation does not include source code memory size (memory required to store the generated code).

Type	Single Precision (kB)	Double Precision (kB)
MPC	10	20
MPC with Online Tuning	20	40

Return to list of tests