Release Notes for Model Predictive Control Toolbox™

How to Contact MathWorks

www.mathworks.com	Web
comp.soft-sys.matlab	Newsgroup
www.mathworks.com/contact_TS.html	Technical Support

suggest@mathworks.com	Product enhancement suggestions
bugs@mathworks.com	Bug reports
doc@mathworks.com	Documentation error reports
service@mathworks.com	Order status, license renewals, passcodes
info@mathworks.com	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Release Notes for Model Predictive Control Toolbox™

Trademarks

Patents

Contents

R2010b

No New Features or Changes

R2010a

R2009b

R2009a

R2008b

R2008a

No New Features or Changes

R2007b

R2007a

R2006b

No New Features or Changes

R2006a

R14SP3

No New Features or Changes

No New Features or Changes

R2012b

Version: 4.1.1
New Features: No
Bug Fixes: Yes

R2012a

Version: 4.1
New Features: Yes
Bug Fixes: Yes

Run-Time Preview of Reference and Measured Disturbance Signals with MPC Controller Block
Compatibility Considerations: Yes

This release introduces the ability to preview signals by using the **ref** and **md** inports of the MPC Controller block and the Multiple MPC Controllers block.

The **ref** inport now accepts an N-by-Ny signal, where N is the number of previewing steps and Ny is the number of plant outputs.

The **md** inport now accepts an N-by-Nmd signal, where N is the number of previewing steps and Nmd is the number of measured disturbances.

You cannot preview if the input signal is a vector, unless Ny or Nmd, as appropriate, is 1.

For more information, see:

• Improving Control Performance with Look-Ahead (Previewing)

• Chemical Reactor with Multiple Operating Points

Compatibility Considerations

In the current release, if you have models with the MPC Controller block or the Multiple MPC Controllers block, you will see a warning if your blocks contain:

• A **custom reference signal** specified in the MATLAB® workspace.

• A **custom disturbance signal** specified in the MATLAB workspace.

Custom Reference Signal Specified in MATLAB Workspace

You must clear this warning. If you ignore the warning, the block will assume that the **ref** signal is zero. This behavior is equivalent to leaving the **ref** inport unconnected.

• **Without Look-Ahead (Previewing) Option.** To eliminate this warning:

  1 Add a From Workspace block to your model.

2 Specify your reference signal variable name as the **Data** parameter of the From Workspace block.

3 Connect the output of the From Workspace block to the **ref** inport of the MPC Controller block or the Multiple MPC Controllers block.

- **With Look-Ahead (Previewing) Option.** To eliminate this warning:

  1 Copy the Reference Previewer block from the mpc_preview model and place it in your model. See Improving Control Performance with Look-Ahead (Previewing) for more information.

  2 Specify your reference signal variable name as the **Signal** parameter of the Reference Previewer block. Also specify appropriate values for the **Sampling time** and **Number of previewing steps** parameters.

  3 Connect the output of the Reference Previewer block to the **ref** inport of the MPC Controller block or the Multiple MPC Controllers block.

Custom Disturbance Signal Specified in MATLAB Workspace

You must clear this warning. If you ignore the warning, the block will assume that the md signal is zero. This behavior is equivalent to leaving the md inport unconnected.

- **Without Look-Ahead (Previewing) Option.** To eliminate this warning:

  1 Add a From Workspace block to your model.

  2 Specify your disturbance signal variable name as the **Data** parameter of the From Workspace block.

  3 Connect the output of the From Workspace block to the **md** inport of the MPC Controller block or the Multiple MPC Controllers block.

- **With Look-Ahead (Previewing) Option.** To eliminate this warning:

  1 Copy the Measured Disturbance Previewer block from the mpc_preview model, and place it in your model. See Improving Control Performance with Look-Ahead (Previewing) for more information.

  2 Specify your measured disturbance signal variable name as the **Signal** parameter of the Reference Previewer block. Also specify appropriate values for the **Sampling time** and **Number of previewing steps** parameters.

3 Connect the output of the Measured Disturbance Previewer block to the **md** inport of the MPC Controller block or the Multiple MPC Controllers block.

R2011b

Version: 4.0
New Features: Yes
Bug Fixes: Yes

C Code Generation Improvements for All Targets with MPC Controller Block

The MPC Controller block has been re-implemented using a MATLAB Function block and now supports code generation for all Simulink® Coder™ targets.

For more information, see Code Generation with Simulink Coder.

Faster QP Solver Algorithm for Improving MPC Controller Performance

This release implements a new quadratic problem (QP) solver that uses the KWIK algorithm. KWIK is faster and more numerically robust than the previous solver for ill-conditioned QP problems. You can use this solver without default constraints on decision variables.

For more information, see MPC QP Solver.

Run-Time Weight Tuning and Constraint Softening for MPC Controller

This release introduces three new run-time tuning parameters for the MPC Controller block:

- **Weights on plant outputs**
- **Weights on manipulated variables rate**
- **Weight on overall constraints softening**

You can use these parameters to tune the weights on plant outputs, manipulated variables rate, and overall constraint softening. These capabilities are available in real time, without redesigning or re-implementing the MPC controller, and help adjust the controller performance.

For more information, see Tuning Controller Weights.

You can also use an `mpcmoveopt` object as an input to `mpcmove` to tune the weights and constraints.

For more information, see the following:

- Switching Controllers Based on Optimal Costs
- Varying Input and Output Constraints

Run-Time Monitoring of MPC Controller Performance to Detect When an Optimal Solution Cannot Not Be Found

This release introduces a new outport parameter—**Optimization status** in the MPC Controller block. You can use this outport to monitor the status of the optimization and take the necessary action when an optimal solution cannot be found. For more information, see Monitoring Optimization Status to Detect Controller Failures.

You can also use the `Info.QPCode` field of the output of `mpcmove` to monitor the status of the optimization.

For more information, see the `mpcmove` reference page.

review Command for Diagnosing Issues with MPC Controller Parameters That Could Lead to Run-Time Failures

You can now use `review` to detect potential stability and robustness issues (both offline and at run time) with an MPC Controller design. The following aspects of the system are inspected:

- Stability of the model predictive controller and the closed loop
- Potential for contradictory settings in the specified constraints and mitigation of an ill-conditioned QP problem by softening constraints
- Validity of QP Hessian matrix

Use this command before implementing the MPC Controller, in conjunction with simulation.

For more information, see the following:

- `review` reference
- Reviewing Model Predictive Controller Design for Potential Stability and Robustness Issues.

mpcmove Returns Aligned Time Horizons for Optimal Control, Predicted Output and Estimated State

mpcmove now returns Info with a time horizon of t=*k*,...,*k+p*, where *k* is the current time and *p* is the prediction horizon for the following fields:

- Info.Uopt — Optimal manipulated variable adjustments

- Info.Yopt — Predicted output

- Info.Xopt — Predicted state

- Info.Topt — Time horizon

You can now plot Info.Uopt,Info.Yopt and Info.Xopt using Info.Topt as the time vector.

For more information, see the mpcmove reference page.

Functionality Being Removed or Changed
Compatibility Considerations: Yes

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
getmpcdata	Still runs	- get - getconstraint - getestim - getindist - getoutdist	Not applicable
pack	Still runs	Not applicable	Not applicable
qpdantz	Warns	quadprog (requires Optimization Toolbox™)	Replace all instances of qpdantz with quadprog.
setmpcdata	Still runs	- set - setconstraint - setestim - setindist - setoutdist	Not applicable

R2011a

Version: 3.3
New Features: Yes
Bug Fixes: Yes

Support for Custom Constraints on MPC Controller Inputs and Outputs

In addition to upper and lower bounds, you can now specify constraints on linear combinations of an MPC controller inputs ($u(t)$) and outputs ($y(t)$). Specify custom constraints, such as $u1 + u2 < 1$ or $u + y < 2$, in the mpc object using setconstraint.

For more information, see:

- Custom Constraints on Inputs and Outputs
- Custom Constraints in a Blending Process
- MPC Control with Constraints on a Combination of Input and Output Signals demo

Ability to Specify Terminal Constraints and Weights on MPC Controller

You can now specify weights and constraints on the terminal predicted states of an MPC controller.

Using terminal weights, you can achieve infinite horizon control. For example, you can design an unconstrained MPC controller that behaves in exactly the same way as a Linear-Quadratic Regulator (LQR). You can use terminal constraints as an alternative way to achieve closed-loop stability by defining a terminal region.

You can specify both weights and constraints using the `setterminal` command.

For more information, see:

- Terminal Weights and Constraints

- Using Terminal Penalty to Provide LQR Performance

- Implementing Infinite-Horizon LQR by Setting Terminal Weights in a Finite-Horizon MPC Formulation demo

Ability to Access Optimal Cost and Optimal Control Sequence

This release introduces two new parameters **Enable optimal cost outport** and **Enable control sequence outport** in the MPC Controller block. Using these parameters, you can access the optimal cost and control sequence along the prediction horizon. This information helps you analyze control performance.

You can also access the optimal cost and control sequence programmatically using the new Cost and Yopt fields, respectively, of the structure info returned by mpcmove.

For more information on using optimal cost and control sequence, see the following demos:

- MPC Control with Input Quantization Based on Comparing the Optimal Costs

- Analysis of Control Sequences Optimized by MPC on a Double Integrator System

R2010b

Version: 3.2.1
New Features: No
Bug Fixes: No

No New Features or Changes

R2010a

Version: 3.2
New Features: Yes
Bug Fixes: Yes

New Ability to Analyze SISO Generalized Predictive Controllers (GPC)

You can now use `gpc2mpc` to convert your SISO GPC controller to an MPC controller. Analyze and simulate the resulting MPC controller using available Model Predictive Control Toolbox™ commands.

For more information, see the `gpc2mpc` reference page.

R2009b

Version: 3.1.1
New Features: No
Bug Fixes: Yes

R2009a

Version: 3.1
New Features: Yes
Bug Fixes: Yes

New Sensitivity Analysis to Determine Effect of Weights on Tuning MPC Controllers

You can now perform sensitivity analysis to determine the effect of weights on the closed-loop performance of your system. You can perform sensitivity analysis using the following:

- MPC Tuning Advisor. See Tuning Advisor in the *Model Predictive Control User's Guide*.

- sensitivity command. See the sensitivity reference page.

R2008b

Version: 3.0
New Features: Yes
Bug Fixes: Yes

New Multiple MPC Controllers Block in the Model Predictive Control Toolbox Simulink Library

You can now use the Multiple MPC Controllers block in Simulink software to control a nonlinear process over a range of operating points. You include an MPC controller for each operating point in the Multiple MPC Controllers block and specify switching between these controllers in real-time based on the input scheduling signal to the block. If you need to change the design of a specific controller, you can open the MPC Design Tool GUI directly from the Multiple MPC Controllers block.

During model simulation, Model Predictive Control Toolbox provides bumpless transfer when the system transitions between operating points.

To learn more about configuring the new block, see the Multiple MPC Controllers block reference page.

Tested Code Generation Support for Real-Time Workshop Target Systems

After designing an MPC controller in Simulink software using the MPC Controller block, you can use Real-Time Workshop® software to build this controller and deploy it to the following target systems for real-time control:

- Generic Real-Time Target

- Real-Time Workshop® Embedded Coder™

- Real-Time Windows Target

- Rapid Simulation Target

- Target Support Package FM5

- xPC Target

- dSpace Target

- Target for Infineon TriCore

The following target systems are either not supported or not recommended because they result in significant performance issues:

- Embedded Target for TI C2000 DSP

- Embedded Target for TI C6000 DSP

- Target Support Package IC1 (for Infineon C166)

- Tornado (VxWorks) Real-Time Target

Note The Multiple MPC Controllers block has not been tested with the target systems supported by Real-Time Workshop software.

29

Ability to Design Controllers with Time-Varying Weights and Constraints Using the GUI

While you design an MPC controller using the MPC Design Tool graphical user interface (GUI), you can specify time-varying weights and constraints for manipulated variables, rate of change of manipulated variables, and output variables. In the previous version, you could only specify the time-varying weights and constraints at the command line.

Furthermore, you can load an MPC controller with time-varying information from the command line into the MPC Design Tool GUI.

To learn more about the new options in the MPC Design Tool GUI, see the Model Predictive Control Toolbox documentation.

R2008a

Version: 2.3.1
New Features: No
Bug Fixes: No

No New Features or Changes

R2007b

Version: 2.3
New Features: Yes
Bug Fixes: No

New Option for Specifying Time-Varying Constraints

You can now configure the Model Predictive Controller block in Simulink to accept time-varying constraint signals that are generated by other blocks. To add inports to which you can connect time-varying constraint specifications, select the new **Enable input port for input and output limits** check box in the MPC Controller block. See also the mpcvarbounds demo.

In the previous version, you could only specify the constraints during the design phase and these contraints remained constant for the duration of the simulation.

For more information about the new **Enable input port for input and output limits** check box in the Model Predictive Controller block, see the MPC Controller block reference page.

Ability to Specify Nondiagonal Q and R Weight Matrices in the Cost Function

You can now specify off-diagonal weights in the cost function. In the previous release, only diagonal Q and R matrices were supported.

To learn more about specifying off-diagonal weights, see the discussion about weights in the MPC Controller block reference pages.

To access a new demo that shows how to use nondiagonal weight matrices, type the following command at the MATLAB prompt:

```
showdemo('mpcweightsdemo')
```

R2007a

Version: 2.2.4
New Features: No
Bug Fixes: Yes

R2006b

Version: 2.2.3
New Features: No
Bug Fixes: No

No New Features or Changes

R2006a

Version: 2.2.2
New Features: Yes
Bug Fixes: No

Bumpless Transfer Added to MPC Block

Bumpless transfer between manual and automatic operation or from one controller to another has been added to the Model Predictive Controller block in Simulink. This block now allows feedback of the true manipulated variable signals, which allows the controller to maintain an accurate state estimate during periods when its calculated adjustments are not being sent to the plant. For example, the controller's output might be ignored during a startup period or during temporary intervention by a (simulated) plant operator. If the controller assumes that its adjustments are being implemented (the default behavior), its state estimate will be incorrect, leading to a "bump" when the controller is reconnected to the plant. A tutorial example has been added to the documentation.

New Bumpless Transfer Demo

A new demo illustrating bumpless transfer has been added to the toolbox.

R14SP3

Version: 2.2.1
New Features: No
Bug Fixes: No

No New Features or Changes

R14SP2

Version: 2.2
New Features: No
Bug Fixes: No

No New Features or Changes