PID Control with Dynamic Disturbance Compensation

SIMATIC PCS 7

Application Example • October 2009
Industry Automation und Drives Technologies Service & Support Portal


Bei Fragen zu diesem Beitrag wenden Sie sich bitte über folgende E-Mail-Adresse an uns:

online-support.automation@siemens.com
SIMATIC
Dynamic Disturbance
Application Example

Preface

Basis Principles of Dynamic Disturbance Compensation

Implementation of Dynamic Disturbance Compensation

Generation of Measurement Data for the Identification

Identification of the Relevant Process Models

Parameterization and Commissioning

Simulation Example

Conclusion

Internet Links

History
Warranty, liability and support

Note

The application examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible in ensuring that the described products are correctly used. These application examples do not relieve you of the responsibility in safely and professionally using, installing, operating and servicing equipment. When using these application examples, you recognize that Siemens cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications - e.g. Catalogs - then the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us - based on whatever legal reason - resulting from the use of the examples, information, programs, engineering and performance data etc., described in this application example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health. The above provisions does not imply a change in the burden of proof to your detriment.

Copyright® 2009 Siemens I IA AS. It is not permissible to transfer or copy these application examples or excerpts of them without first having prior authorization from Siemens I IA AS in writing.
# Table of Contents

Warranty, liability and support .............................................................. 4

1 Preface .................................................................................................. 6
  1.1 Objective of the Application .......................................................... 6
  1.2 Main Contents of this Application Note ......................................... 6

2 Basis Principles of Dynamic Disturbance Compensation .................. 7
  2.1 Area of Application ...................................................................... 7
  2.2 Mode of Operation ...................................................................... 8
  2.3 Application Examples .................................................................. 9

3 Implementation of Dynamic Disturbance Compensation ................. 10
  3.1 Installation .................................................................................. 10
  3.2 Configuration: Creating an Instance of the Process Tag Type ....... 10

4 Generation of Measurement Data for the Identification ................. 14
  4.1 Excitation of Process and Recording of Training Data ................. 14

5 Identification of the Relevant Process Models ................................... 18
  5.1 MPC Configurator ....................................................................... 18
  5.2 Examination of the Model Parameters ....................................... 21

6 Parameterization and Commissioning .............................................. 24

7 Simulation Example ............................................................................ 25

8 Conclusion .......................................................................................... 26

9 Internet Links ........................................................................................ 27

10 History .................................................................................................. 28
1.1 Objective of the Application

The objective is fast and tight control of processes affected by strong disturbances. Thereby the disturbances must be known and measurable. The implementation and the potential for improvement in comparison to a conventional PID controller will be shown with the “APL_Example_EU”.

1.2 Main Contents of this Application Note

The following issues are discussed in this application:

- How to create an instance of the process tag type
- How to collect the necessary measurement data for modelling
- How to identify the relevant models with the MPC configurator
- Benchmark simulation with and without disturbance compensation, to show the potential benefits

Validity

… valid for PCS 7 V7.1, in principal transferable to V7.0 from SP1.
Basis Principles of Dynamic Disturbance Compensation

Note

A general overview of the APC functionalities (Advanced Process Control) of PCS 7 is provided by the White Paper „How to improve the Performance of your Plant using the appropriate tools of SIMATIC PCS 7 APC-Portfolio?“ (see link [3] in chapter 9 “Internet Links”)

2.1 Area of Application

Figure 2-1 Disturbance compensation

A signal, which has a significant impact on the process variables (particularly on the control variables) although it is not part of the considered process and cannot be actively manipulated by the controller, is called disturbance variable. You can see it in Figure 2-1

Feedforward disturbance control can be used when a known, strong disturbance affects the process and its physical cause can be measured. In these cases, the following general strategy applies: "Feedforward control as much as possible (as much as known in advance and described by a model), feedback control as much as necessary (the rest including the model error and immeasurable disturbances)".
2.2 Mode of Operation

The impact of a measureable disturbance can be estimated in the form of a transfer function \( g_z(s) \) when the controller is running in manual mode so that no variations of the controlled variable are caused by the manipulated variable and all changes can be attributed to the disturbance \( z \). The transfer function of an ideal feedforward control \( c(s) \) can be derived from the requirement that the impact of \( z \) on the controlled variable \( y \) should be zero for any disturbance signal \( z \) (condition of invariance):

\[
g_z(s)z + c(s)g(s)z = (g_z(s) + c(s)g(s))z = 0
\]

To meet this equation, the compensation block \( c(s) \) must approximate the equation

\[
c(s) = -\frac{g_z(s)}{g(s)}
\]
as well as possible. In order to achieve this, the disturbance transfer function \( g_z(s) \) must be known and the transfer function of the main controlled system \( g(s) \) must be inverted. If both transfer functions can be modelled as first order plus dead time (PT1T1-element)

\[
g_{z}(s) = \frac{k_{sz}}{1 + t_1s} e^{-\bar{\theta}_z}, \quad \text{and} \quad \theta < \bar{\theta}_z \text{ applies, the resulting compensation element}
\]

must represent the lead-lag transfer function

\[
c(s) = \frac{k_{sz}}{k_S} \frac{1 + t_1s}{1 + t_1s} e^{-s(\bar{\theta}_z - \theta)} = -k_c \frac{1 + t_ds}{1 + t_s} e^{-s\bar{\theta}_z},
\]
i.e. a PDT1T1-element (derivative action element with lag and dead time). Such a transfer element is available as a standard function block in many process control systems or can be created by a combination of elementary function blocks. An additional input of the PID function block allows adding this signal to the MV value.

It is important that any addition of sideline contributions to the MV value is performed in front of the MV limitation of the controller block (in opposite to the simplified signal flow chart in Figure 2-1), in order to ensure proper limitation of the overall MV including anti-windup logic (avoiding overflow of the internal integrator inside of the PI(D) controller during active limitations).

However for general transfer functions \( g(s) \) and \( g_z(s) \) there will be more complicated or even unfeasible compensation functions. If e.g. both transfer functions are determined as time-lag elements of order \( n \):

\[
g(s) = \frac{k_s}{(1 + t_1s)^n} \quad \text{and} \quad g_z(s) = \frac{k_{sz}}{(1 + t_1s)^{n_z}},
\]
the compensation block is only feasible if \( n_z \geq n \).

Sometimes the compensation function has to be simplified by order reduction, which might reduce the efficiency of disturbance compensation. This simplification
can go that far, that the process dynamics is completely neglected, and only \( c(s) = k_{sz} / k_{sz} \) is implemented (static feedforward control).

### 2.3 Application Examples

- Temperature control of an industrial oven: at the oven inlet, the disturbance variable feed flow is measured and fed-forward to the output of the temperature controller. The impact of varying flows on the oven temperature is anticipated and compensated for by modifying the heating power.
- Controlling the outlet temperature of a heat exchanger via steam pressure or heating/cooling medium flow: flow and inlet temperature of the medium are the measurable disturbance variables.
- Fill level control in a drum steam generator using the inlet volume: the outflow is the measureable disturbance variable that is determined by the variable steam consumption in the plant.
- Temperature control in a distillation column using the reflux ratio or heating steam flow: the measurable disturbance variable is the mixture feed flow.
- Temperature and concentration control in an agitated tank reactor using cooling medium flow and discharge volume: the temperature and possibly also the concentration of the inflow are measurable disturbance variables.
Implementation of Dynamic Disturbance Compensation

The principal approach to configure dynamic disturbance compensation is very similar to the design of a PID controller with an additive identification of the disturbance model. The configuration is done in several steps as it is explained in the next chapters:

- Generation of the process tag type
- Parameterization of the PID controller
- Recording of the step responses for the identification of the process and the disturbance model with the CFC trend display and exporting to an archive file
- Identification of the process and the disturbance model with the MPC configurator
- Parameterization of the disturbance compensation and download to the AS

3.1 Installation

The installation of the PCS 7 Advanced Process Library is performed automatically by the PCS 7 master setup V7.1.

Note

There is a dynamic disturbance compensation process tag type named “PIDCTRL_DistComp” already available in the PCS 7 APC Library V7.0 SP1. Although the Advanced Process Library of PCS 7 V7.1 is used in this Application Note, the principal procedure is also applicable with the APC library of PCS 7 V7.0 SP1. Using an even older version a CFC can be projected manually according to this tag type. However, the identification of the process model in an older version has to be done with an external tool.

3.2 Configuration: Creating an Instance of the Process Tag Type

The following steps are carried out for the dynamic disturbance compensation in the same way as for any other process tag type.

Please open the “PCS7 AP Library V71” via “File”/”Open”/“Libraries” in the Simatic Manager.
Copy the process tag type “FfwdDisturbCompensat” from the subfolder “Templates” into the master data library of your PCS 7 multiproject and modify it if this is necessary according to your general application requirements.

Copy the process tag type from the master data library to the application part <project name>_Prj of your multiproject, in the appropriate target folder (Process cell/Unit etc.) in the plant view. You obtain an instance of the process tag type i.e. a CFC chart, which indicates its origin by its symbolic representation.

Rename the new CFC chart and check if the cyclic interrupt OB is correct (in the CFC chart “Edit”/“Open run sequence”).

Open the CFC chart and implement the following connections:

- **Control variable**: Connect the analog input driver Pcs7AnIn for the control variable “PV” (see Figure 3-4, number 1) with the symbolic name of the corresponding peripheral signal from the hardware configuration. The unit of the signal can be adjusted at the input “PV_InUni”.
Implementation of Dynamic Disturbance Compensation

- Disturbance variable: Connect the analog input driver Pcs7AnIn for the disturbance variable “DV” (see Figure 3-4, number 2) with the symbolic name of the corresponding peripheral signal from the hardware configuration. The unit of the signal can be adjusted at the input “PV_InUni”.

- Manipulated variable: The manipulated variable “MV” has to be connected to the periphery via the output “PV_Out” of the analog output driver Pcs7AnOu (see Figure 3-4, number 3). The unit of the signal can be adjusted at the input “PV_InUni” of the analog output driver. The input “PV_In” of the analog input driver Pcs7AnIn named “MV_Rbk” must be connected to the actual achieved manipulated variable of the periphery (see Figure 3-4, number 4). If no analog position feedback is available, delete the function block “MV_Rbk” and its connections.

Compile and download your changes to the AS. Compile the OS again to include the new PID faceplate as well as the faceplate to activate the disturbance compensation in your runtime application.

Now you have successfully integrated the dynamic disturbance compensation to your process. In the next step the algorithm must be configured adequately. Adjust the PID controller with the PID tuner according to the application note “PID Control with Gain Scheduling”. Therefore, start the PID tuner and run through all the steps. Please be sure to run the controller in manual mode, otherwise switch off the disturbance compensation while the PID tuner is operating in automatic mode of the controller.

In some cases a conventional controller is already existing and parameterized. These parameters can be kept if you are satisfied with the performance of the controller. The behaviour of the controller is only affected by the dynamic disturbance compensation during a variation of the disturbance variable. As long as the disturbance is constant the disturbance compensation has no impact on the variations of the manipulated variable. The setpoint response of the control loop is not affected by the disturbance compensation.

However, the controller and the disturbance compensation are not completely independent of each other. If the disturbance compensation is not perfect, a part of the compensation work remains for the controller. In such a case, please verify if the interaction of controller and disturbance compensation meets the requirements during operation. If the performance is not satisfying due to uncertainties in the disturbance model, the disturbance compensation should be modified (tuned less aggressive) rather than the controller. Example: The gain $k_c$ of the compensation block (see section 2.2) should be rounded down rather than up, if the gain of the disturbance transfer function is not known exactly.

In the following, record the measurement data for the identification of the partial transfer-functions of the disturbance compensation and examine the model parameters.
Figure 3-4 Important connections of the process tag type
4.1 Excitation of Process and Recording of Training Data

Both partial transfer-functions \( g(s) \) and \( g_z(s) \) from Figure 2-1 must be identified to build up the disturbance compensation. The main part of the process \( g(s) \) has already been identified during the PID tuning. However, the model representation of the PCS 7 PID tuner (PTn model) \( g(s) = \frac{k_s}{(1 + t_s s)^n} \) can be converted to form with dead time \( g(s) = \frac{k_s}{1 + t_s s} e^{-s\theta} \) used by the template only with the help of external control engineering software packages (e.g. Matlab). Hence, it is mostly easier to excite and identify both partial processes with the MPC configurator successively.

Therefore, the process is excited with steps in the manipulated and in the disturbance variable in manual mode of the controller.

If the disturbance variable cannot be modified actively (e.g. the environmental temperature), you have to wait for an autonomous change. If necessary it is useful to search through the archive of historical data for significant changes in the disturbance variable (e.g. a sudden weather change). Please be aware of avoiding other disturbing influences (load changes, maintenance work, other not measureable disturbances, etc.) during the recording, as all changes in the control variable are mathematically attributed to the single measureable disturbance variable.

The measurement data is recorded with the CFC trend display and afterwards exported to an archive file.

Select “Trend Display” in the menu “View” in the CFC to open the trend display.
Create a new trend display and specify several parameters with the “Change“ button:

- The number of collected measured values;
- The acquisition cycle: It must be an integer multiple of the related interrupt OB and should be large enough to collect sufficient step responses. The rule of thumb says that the shortest transient effect (step response) should be collected with at least 200 measured values. The maximal time range captured with these settings results from the multiplication of the number of measured values and the acquisition cycle.
In the next step the variables (manipulated, controlled and disturbance variables) to be captured must be selected. Move the corresponding signals via drag&drop from the CFC chart of the dynamic disturbance compensation to the trend display, select the value part of the data structure and apply it to a free channel.

Move your process to its operating point and wait for steady state. Now, you can start with the recording of measurement data. Be sure to decide which step changes to the manipulated and disturbance variables you want to apply before starting the recording. Some notes on this topic are given in the following. The measurement data should be symmetrical to the operating point for a successful identification. Therefore, a single step response is not adequate. Another issue is that the dynamics of the process should be excited completely. As an example of a possible excitation, step responses of the manipulated variable could be executed first followed by the ones for the disturbance variable. Inbetween the changes of the different signals, steady state must be reached. Figure 4-7 shows an example.

Additional remarks to the choice of excitation are given in the online help of the MPC configurator (“The individual design steps in detail”/“Recording the measurement series”).

If exceptionally the recording of the training data for the controller is made in automatic mode, you have to deactivate the disturbance compensation. This can be done via the Faceplate “DistCompOn” or via the inputs of the function block OpDi01 “LiOp=0” and “OnOp=0” (located on sheet 4 of the CFC).

Activate the test mode of the CFC and start the recording with the trend display. Apply your planned step changes to the manipulated and the disturbance variable successively.
If all step responses are finished and the process is in steady state again, the recording can be stopped.

Export the measurement data to a csv-file via the “Export” button. Please keep the default settings.

If a validation of the identified model during the MPC configuration is desired, you have to generate an additional measurement data set in the same way as for the identification, but with a different excitation signal.
Identification of the Relevant Process Models

5.1 MPC Configurator

The identification is done with the MPC configurator. Select the PID controller in the CFC and start the configurator via the menu “Edit”/“Configure MPC”.

Figure 5-8 Start window of the MPC configurator

Select your recorded measurement data via “Load data”. Afterwards all captured variables are displayed in the window.
Assign its role to each variable and specify if it is a manipulated, controlled or disturbance variable or if it is not relevant. In the lower part of the window the time range of the measurement data can be adjusted. If your process has dead times, mark the appropriate tick mark. Moreover it is possible to smooth the measurements with a noise filter and to downsample the data to reduce the amount of data.

In this application example a model with dead time has to be identified.

It is possible to load several different data sets, or equal data sets with different time ranges. All loaded data is considered for the identification. Right click on the name of the corresponding file in the data selection to remove a data set from the identification.

The identification can be started via the “Identify” button.
The results of the identification are shown in a new window (see Figure 5-10). Evaluate these results carefully and decide if this model has the adequate accuracy. The Bode diagrams can be analysed by clicking on the step responses. A click on the control variables shows a comparison of measured and simulated data of the model as can be seen in Figure 5-11. Altogether the model accuracy should be as high as possible, anyway more than 50%, the dynamic should be covered adequately and the step responses should be stable.

Figure 5-11 Accuracy of the model (Model qualities of 98% can only be achieved with simulated data.)
5.2 Examination of the Model Parameters

The particular model parameters process gain, dead time and time constant can be determined from the corresponding step response. In the following this procedure is presented for the process model.

You can open the step responses directly in the MPC configurator. Use the zoom function and the data cursor, which shows the value of the plotted characteristic directly, to get a more accurate evaluation.

Note: Due to the considered fast simulation in the example, the roughly sampled step responses cannot be accurately evaluated in the MPC configurator. Therefore, in the following all time constants of the simulation model are multiplied by 5 to get a better visualization of the step response. After finishing the determination of the model parameters, this “didactical” change will be reversed again. Such a manipulation is not necessary and even not possible in a real process!

Figure 5-12 Take a reading of the process gain and the dead time from the step response

- Process gain: The stationary process gain can be read at the rear end of the step response (see Figure 5-12). In the example you obtain a value of 0.9409.
- Dead time: The dead time can be read at the front end of the step response (see Figure 5-12). The time axis is always scaled in seconds. In the example you obtain a dead time of 8s.
- Time constant: There are two possibilities to determine the time constant. The one easier but more inaccurate is based on a tangent applied to the steepest part of the step response. The time span from the beginning of the step response (at the end of the dead time) up to the intersection of the tangent with the final value of the step response results in the time constant (see Figure
In the example using the described method leads to a time constant of 22s - 8s = 14s.

Figure 5-13 Take a reading of the time constant by the means of an inflection tangent; Due to a model of 1st order in the example considered here, the inflection tangent is equal to the tangent at the starting point.

The second possibility to determine the time constant is more complex but its results are more accurate. The time constant is determined from the area above the step response between the starting point (after the dead time) and the endpoint of the step response, after division by the process gain. It is advisable to split the area into rectangular beams to achieve the best approximation (see Figure 5-14).

In the application example you obtain a time constant of

\[
T = \frac{1}{0.9409} \cdot \left( [0.9409 - 0.1144] + [0.9409 - 0.3416] + [0.9409 - 0.5413] + [0.9409 - 0.6767] + [0.9409 - 0.7715] + [0.9409 - 0.8374] + [0.9409 - 0.8833]\right) \cdot 5s \\
\approx 12.86s
\]

For more information about the determination of the time constant see the application note “Smith Predictor for Control of Processes with Dead Times”. 
Figure 5-14 Take a reading of the time constant by the means of an area

Repeat the same procedure for the disturbance model. Note the estimated parameters and close the MPC configurator.

**Note**

The graphical determination of the model parameters is no more necessary for PCS7 Version V7.1 SP1 or higher. There the parameters can be read directly from the MPC configurator by clicking on the desired partial transfer function.
Parameterization and Commissioning

The determined model parameters of the transfer functions of the disturbance compensation can now be entered in the input parameters of the corresponding function blocks. In the application example, the disturbance compensation can be approximated to be

\[
\frac{g(z)}{g(s)} = -\frac{1}{2} \cdot \frac{2s + 1}{3s + 1} e^{-0.4s}.
\]

Therein, the “didactical” changes (the multiplication of the time constants by 5) are reversed again.

Open the CFC of the disturbance compensation and assign the parameters dead time “DeadTime” (see Figure 6-15, number 2), process gain “In2” (see Figure 6-15, number 1) and the time constant of the derivative action element “TD” (Figure 6-15, number 4) and the time constant of the PT1 element “LagTime” (see Figure 6-15, number 3 and 4) to the related function blocks in the sheet above the PID controller. The controller parameters are automatically assigned by the PID tuner.

Now, to successfully integrate the dynamic disturbance compensation, compile all changes and download the program to your AS.
Simulation Example

The process simulation is built up twice in the simulation example DisturbCompSim of the APL_Example_EU – one process tag with PID controller including dynamic disturbance compensation and the other one with a conventional PID controller, while all other process parameters are identical. The signal flow chart in the OS figure shows the control loop of the PID controller TIC301 with disturbance compensation. The symbol of the conventional controller TIC302 is located below the one of TIC301 (without the illustration of the corresponding signal flow). The advantages of the PID controller with disturbance compensation can be proved in a direct comparison (benchmark simulation, “parallel slalom”).

As a simulation scenario the disturbance variable is moved from 0 to 40 and to 0 again in both processes.

Figure 7-16 The APL example: comparison of the disturbance influence

While the control variable “PV” of the conventional controller shows significant over- and undershoots during the variations in the disturbance variable, the PV of the controller with disturbance compensation remains unaffected. The immediate addition of “MV_Feedf” to the manipulated variable within the disturbance compensation is the reason for this advantage. In contrast, the conventional controller “waits” for a control deviation to be adjusted by the I-part of the controller.
Conclusion

An improvement of the performance during variations in the disturbance variable can be reached by the exploitation of the measurable disturbance in the controller, especially if the impact of the disturbance on the control loop is strong and/or frequent.

The disturbance compensation consisting of few elementary CFC function blocks needs only small CPU capacity. However, some engineering effort is due to the fact that the disturbance transfer function must be known to specify the parameters of the compensation block.

A dynamic disturbance compensation can also be realized with a model predictive controller (see e.g. [1.] or the application note “Multivariable Model Predictive Control – the Distillation Column as an Application Example”), in multi-input multi-output and in single-output constellations. It provides greater flexibility and accuracy in system modelling and is more convenient thanks to the integrated design tool. However, it does require more CPU resources. The following table shows a comparison.

Table 8-1 Comparison of the PID controller with dynamical disturbance compensation and MPC C

<table>
<thead>
<tr>
<th></th>
<th>PI(D) controller (conventional)</th>
<th>PI(D) controller with dynamical disturbance compensation</th>
<th>MPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SISO/MIMO case</td>
<td>SISO only</td>
<td>SISO only</td>
<td>SISO or MIMO</td>
</tr>
<tr>
<td>(single input single output / multi variable case)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control performance</td>
<td>low</td>
<td>good</td>
<td>very good</td>
</tr>
<tr>
<td>Engineering-effort</td>
<td>low</td>
<td>high</td>
<td>medium</td>
</tr>
<tr>
<td>CPU resource-consumption</td>
<td>low</td>
<td>minimal higher than conventional</td>
<td>high</td>
</tr>
<tr>
<td>(memory, calculation time)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Internet Links

## Table 9-1 Internet links

<table>
<thead>
<tr>
<th>Topic</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>2) Siemens I IA/DT Customer Support</td>
<td><a href="http://support.automation.siemens.com">http://support.automation.siemens.com</a></td>
</tr>
</tbody>
</table>
## History

### Table 10-1 History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>02.10.2009</td>
<td>First release</td>
</tr>
</tbody>
</table>

Copyright © Siemens AG 2009 All rights reserved