****

**Smart-Stepping Model Identification**

**Charles Thabo Ngwamuse**

**12284922**

**Introduction**

Process identification is an important stage in modern model-based controller installations. Several techniques exist for this identification, and the problem is that the procedure for introducing disturbances into the system to allow for identification is very time consuming, requiring both engineering time and time on the plant. The objective of this project is to develop a stepper which can keep a plant within control limits while stepping automatically. This research is limited to identification of linear processes with white noise, the existence of non-linearity do not affect the outcome of this research project.

The need for process identification may rise from a “black box” problem where the concerned process is poorly understood or too complex (Doren & Vance, 2003). In most cases, the process identification is employed for tuning of the controller to obtain the optimal controller parameters. It should be established whether the process identification will be performed for controller tuning or for obtaining an insight of the process parameters such as the physical constants. The latter is more concerned about the quality of identification rather than the optimal control action.

For the purpose of controller tuning, a low-order model may be appropriate for estimating a high-order process in which a PID controller is implemented (Doren & Vance, 2003).

Several algorithms or methods exist for process identification. These algorithms are favourable under certain conditions such as the noise and the input characteristics. Identification is classified into two categories; the off-line and the online identification. An offline identification involves the collection of the sampled processes data such as the process input signal and the process output over a specified time span, the subsequent step being the mathematical manipulation of data in order to come up with adequate parameter estimates of the process, this involves matrix algebra. The offline technique is sometimes referred to as the classical technique because it precedes modern techniques and is the oldest technique. In contrast to offline identification, online identification involves the collection of the most recent process data in order to update the previous estimated process parameters, the online procedure is recursive and is often referred to as the recursive identification (Soderstrom & Stoica, 1989).

Online identification is responsible for adaptive controllers or the self-tuning regulators (STR). Frequent changes in the operating conditions are expected in a real process where the process parameters may change with time which will result in a mistuned controller, hence an adaptive controller insures optimal control action at changing conditions (Doren & Vance, 2003).

Not all identification algorithms will converge to true values under certain conditions, for example the least square method will break down under high levels of noise while the prediction error method is more resistant to noise (Soderstrom & Stoica, 1989).

**Theory**

**Offline Identification**

Offline identification is relatively the oldest techniquefor model prediction, it is easier to perform in terms of mathematical complexity than its counterpart, online identification. The online identification is essentially a modification of offline technique and is popular in modern plants that implement adaptive controllers. Not every plant needs an online identification, for example if a process is well known to be time-invariant then an offline is efficient (Doren & Vance, 2003).

The system investigated in this paper is a 2-input-2-output multivariable case both in open loop and closed loop. To facilitate the understanding of system identification in general a method for identification will be presented, particularly the least-square method. A general single-input-single-output linear system is described by

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Where y(t) is output and u(t) is the input (Soderstrom & Stoica, 1989). The difference equation difference equation above can be easily be transformed to either the time or the Laplace domain, this paper is making use of the difference equation for simulation and identification. Equation 1 can be written in matrix format as

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Where

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

This paper will also employ the ARMAX model using shift operators, this model structure is written as

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

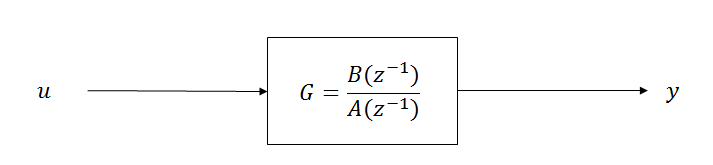
|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Where

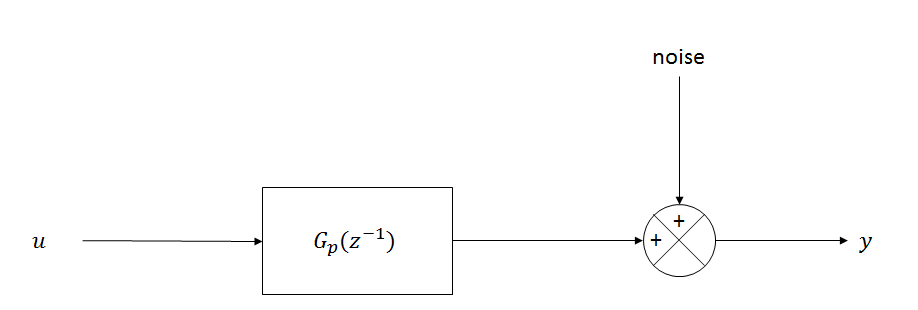
|  |  |  |
| --- | --- | --- |
|  |  | (6) |

In the equation above, z-1 denotes the backward shift operator so that (Hastings & Sage, 1969)

|  |  |  |
| --- | --- | --- |
|  |  | (7) |



Unlike the model, the real process contains disturbance elements such as noise that are neither correlated to the input nor the process output. The real process for SISO system is shown in the figure below (Soderstrom & Stoica, 1989).



The system above is assumed to be generating data for offline identification. This system is represented by

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

**Least-Squares Estimator**

The least square method minimizes the error between the process output and the model output. By evaluating the gradient of the function that gives the residual error, it can be shown that the minimum error occurs at (Soderstrom & Stoica, 1989)

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Equation (9) above is used for offline identification.

**Online Identification**

If we define a variable

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

Then it follows that

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

The algorithm above is known as the recursive identification by least-square method. This algorithm can be put into more useful form that eliminates matrix inversion, particularly the updating of P(t), the equation for P(t) is equivalent to (Soderstrom & Stoica, 1989)

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

Thus

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

The algorithm above also applies to multivariable systems since multiple-input-multiple-output is merely an addition of multiple-input-single-output systems. It is as easy to identify a MISO system as is to identify as SISO system. The least-square method is, however, valid under certain condition, these conditions are

* The system should have a white noise for estimated parameters to converge to true values.
* For high-order systems, the deviation of the estimated parameters from true values is often more substantial than with low-order systems.
* The least-square method is sensitive to signal-to-noise ratio (Soderstrom & Stoica, 1989).

The process used in this paper has a white noise with a noise-to-signal ratio of < 0.1, this should be fine as the maximum signal-to-noise ratio is 0.2 according to the literature (Doren & Vance, 2003). The system is a 2x2 multivariable system which is relatively a low-order system. Different noise levels will be investigated also to determine the noise-to-signal ratio at which the least-square method breaks down.

*Initial Values*

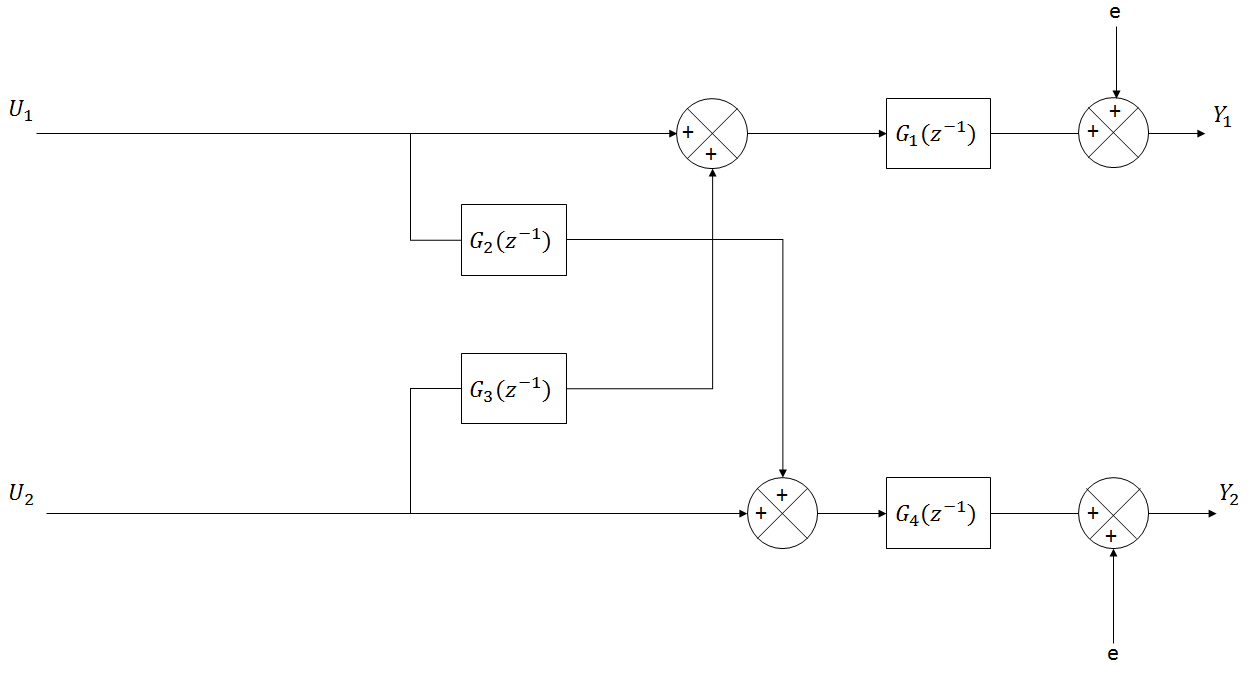
The algorithm from (12) to (13) requires initial values and. The parameters can be assumed to all start at zero but if the variable P(t) starts as a zeros matrix it will continue as a zeros matrix. The literature recommendation is that

|  |  |  |
| --- | --- | --- |
|  |  | (14) |

where σ is a very large scalar number. If P(0) is small then the parameters will converge slowly, this is evident from (13) since K(t) is proportional to P(t), small K(t) implies a small change in (Soderstrom & Stoica, 1989).

*Process Configuration*

The system to be investigated is presented in the configuration below, the controller will be implemented later in this paper.



The figure above represent the real process, Ui is the input, Yi is the output and e is noise. For simplicity, this system will be interpreted as a combination of two MISO systems where all the transfer functions, Gi(z-1), are first-order. This system can be mathematically modeled as

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

In the matrix form, the above system above becomes

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

Where

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

The noise will treated as white, i.e. with zero mean and a variance of σ2 thus this favors the use of the least-square method.

The system above can also be modeled using the Laplace domain, but the configuration does not change and one can easily convert the transfer functions to be in the Laplace domain. The parameters are given in terms of the Laplace domain and the z-domain below.

|  |  |  |
| --- | --- | --- |
|  | **τ** | **K** |
| G1 | 1 | 1 |
| G2 | 5 | 0.5 |
| G3 | 6 | 0.2 |
| G4 | 2 | 1 |
|  | **a** | **b** |
| G1 | 0.6321 | 0.3679 |
| G2 | 0.0907 | 0.8187 |
| G3 | 0.0307 | 0.8465 |
| G4 | 0.3935 | 0.6065 |

Substituting the transfer functions into equation (15) results in

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

Which can be written in a more useful way for simulation as

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

Where

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

The expression above appears to have twelve degrees of freedom (6 to each output) while the system in (18) has eight (4 to each output), the manipulation in (20) still has the same DOF as in (18), the “extra” parameters are just functions of the original parameters in (18) i.e. they are dependent. There is no way to reduce the expression in (20) to only eight independent parameters unless some parameters are equal.

*Model Assumption*

The assumed model has twelve degrees of freedom, this model is guaranteed to capture all the process behavior since it has extra four DOF, if this model is consistent, then the extra four parameters should be functions of the other eight parameters, for example if ,, and are successfully estimated, a consistency check can be done to see if the relationship in (20) holds. The assumed model is shown below

|  |  |  |
| --- | --- | --- |
|  |  | (21) |

**Inputs**

When identifying a process, the selection of the input signal is crucial in the convergence of the process parameters. Different inputs have different characteristics and will essentially cause a different reaction curve. It is a necessary condition that an input be independent of the process output for process identifiability, this condition is not met in the closed loop system because the input to the process depends on the output. This, however, does not mean that a closed loop system is not identifiable.

*Step function as an input*

A step input is not dependent on the process output and therefore will be a potential candidate for process identification. For low-order processes, a process reaction to a step change may reveal some information about the process such as the time constant and the process gain. For high-order and non-linear processes, this approach of identification may prove to be impossible.

To estimate process parameters and especially online, a persistently exciting signal is required to obtain good quality results. A step input is not exciting enough since it maintains a single value almost all the time. For offline identification it can be shown that a step input will give parameter estimates that converge to the true values as the number of samples increase and on a condition that the noise is white. On a closed loop system, the input to the process is not necessarily a step function, again, it will depend on the deviation of the process variable from the setpoint (Soderstrom & Stoica, 1989).

*An impulse as an input*

An impulse is also not dependent on the process variable. It is hard to reveal the process information using an impulse as an input, this is because an impulse does not excite the process sufficiently and it is equal to zero too often. This means that a reasonable model will not be obtained even in an offline identification with a very large sample, thus an impulse as an input will not be considered as a potential candidate for identification (Soderstrom & Stoica, 1989).

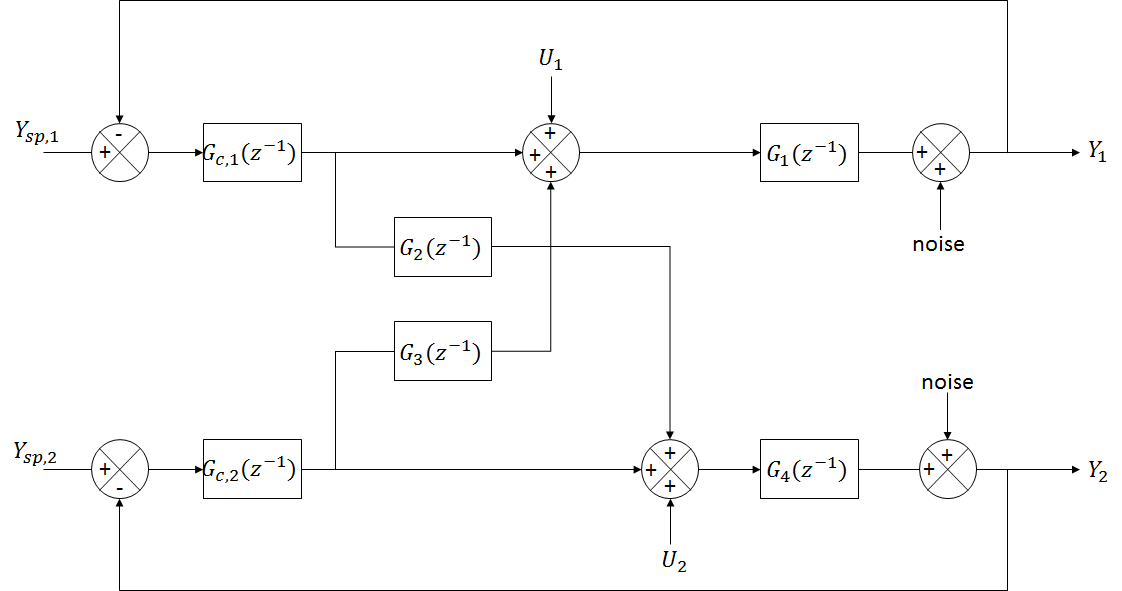
*A pseudorandom binary sequence (PRBS) as an input*

A PRBS signal shifts between two values such that its mean is zero. This signal is similar to a square wave except that its period is random. A PRBS signal can be shown to give consistent parameters in both online and offline identification. This is also apparent because this signal persistently excites the process and has no pattern at all. PRBS speed up identification i.e. faster convergence of the process parameters and hence this is an advantage during the start-up when the controller performance is a prime concern (Soderstrom & Stoica, 1989).

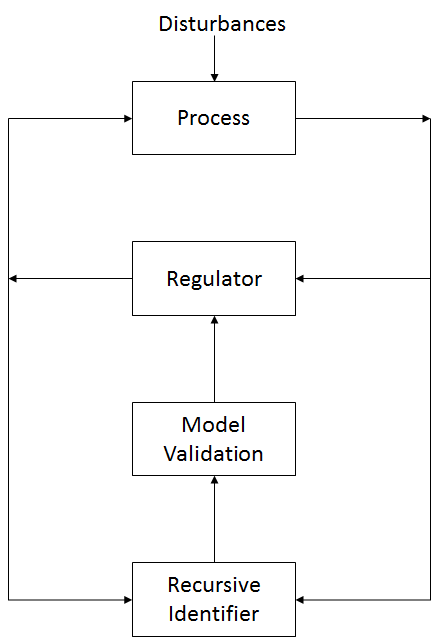
*Closed loop system.*

The closed loop system is relatively hard to identify, this is due to the input being dependent to the process variable as stated above. It is, however not impossible to identify a closed loop system. To solve the problem of the input being dependent on the output, a known disturbance is added to the controller output. This disturbance can be a step, an impulse or a PRBS signal dependent on the type of identification and the purpose of identification.

Most often, an online closed loop identification is done to find the optimum controller settings automatically, the quality of the identification in this case is not a prime concern. In this study, the main focus will be on the quality of the identification. The figure below represent the implementation of a discrete PID controller to the open loop process above.



The model above will be treated as an approximate to a real process where parameters may change with time, for example a process gain change that is due to fouling on the heat exchanger surface. The inputs to be sampled are; the first controller output, the second controller output and the two disturbances, the outputs will be sampled as well. The figure below illustrates the identification procedure that will typically be employed online in the industry.



*Model Validation*

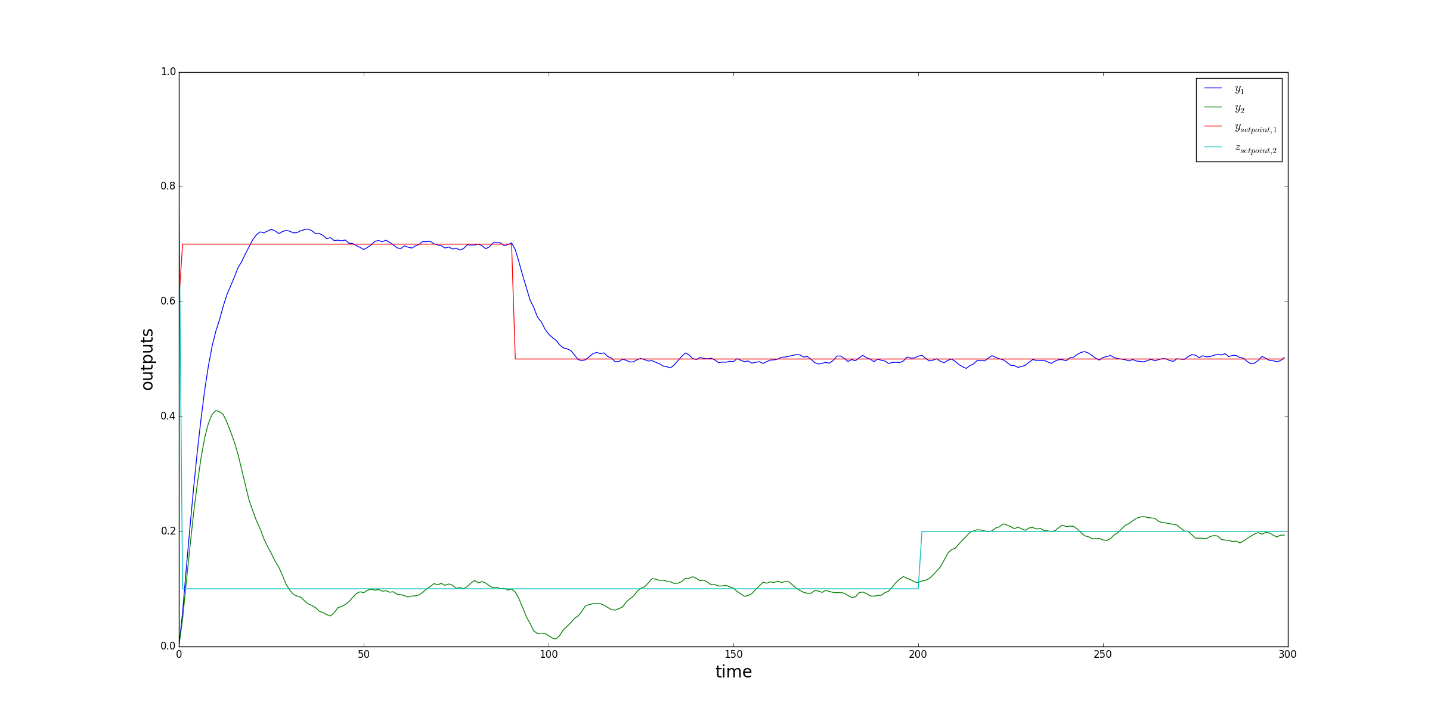
The subsequent stage after recursive identification is to decide if the model is adequate or not. A simple way would be to compare the estimated parameters to the true values but we would not have the true values in a real process. The only way we may validate the model is to compare the model output to the process output, an adequate model should explain all the patterns that are due to the input (Doren & Vance, 2003).

The model output will always deviate from the noise-free process output since the estimated parameters minimize the error between the model output and the process output.

*Controller*

The controller is not set to self-tuning mode, thus the controller parameters are constant throughout the whole run, this is because the process parameters are assumed to constant throughout. The controller parameters are summarized in the table below.

|  |  |  |
| --- | --- | --- |
|  | **τi** | **K** |
| GC,1(s) | 5 | 0.1 |
| GC,2(s) | 2 | 0.5 |



The figure shows the closed loop response to a step change in the setpoint, it can be seen that the controller keeps track of setpoint changes but this controller setting may not be the optimum setting. For closed loop simulations, this controller setting will be used.

*Dead Time*

Dead time is non-linear with respect to identification and therefore a linear identification cannot estimate the process dynamics and the dead time simultaneously and since this paper is restricted to linear identification, the dead time term will be intentionally omitted. Possible solutions to estimating the dead using linear identification involve:

* Identifying multiple process models, each with a different assumed value for dead-time and selecting the model with minimum error between the process output and model output (Doren & Vance, 2003).
* Employing a higher order model and then estimating the dead-time based on the significance of the identified parameters (Doren & Vance, 2003).

Dead time estimation hence introduces an addition step in process identification but does not really change the way at which the process dynamics are estimated.

**Results from Simulations**

The simulation was run for 500 minutes using a sampling interval of 1 minute. The real process was simulated using the difference equation as shown in (), hence the true parameters as shown in (theta) are presented in the table below.

|  |  |
| --- | --- |
| Parameter | True Value |
| a11 (b1 + b3) | 1.214 |
| a22 (-b1b3) | -0.3114 |
| a33 (a1) | 0.6321 |
| a44 (-a1b3) | -0.5351 |
| a55 (a3) | 0.0307 |
| a66 (-a3b1) | -0.0113 |
| b11 (b4 + b2) | -1.425 |
| b22 (-b4b2) | 0.4966 |
| b33 (a4) | 0.3935 |
| b44 (-a4b2) | -0.3221 |
| b55 (a2) | 0.0906 |
| b66 (-a2b4) | -0.05497 |

The parameters in the table above are functions of the parameters in table (), the functions are indicated in brackets, see table () for reference. Equation () is re-written as

|  |  |  |
| --- | --- | --- |
|  |  | (22) |

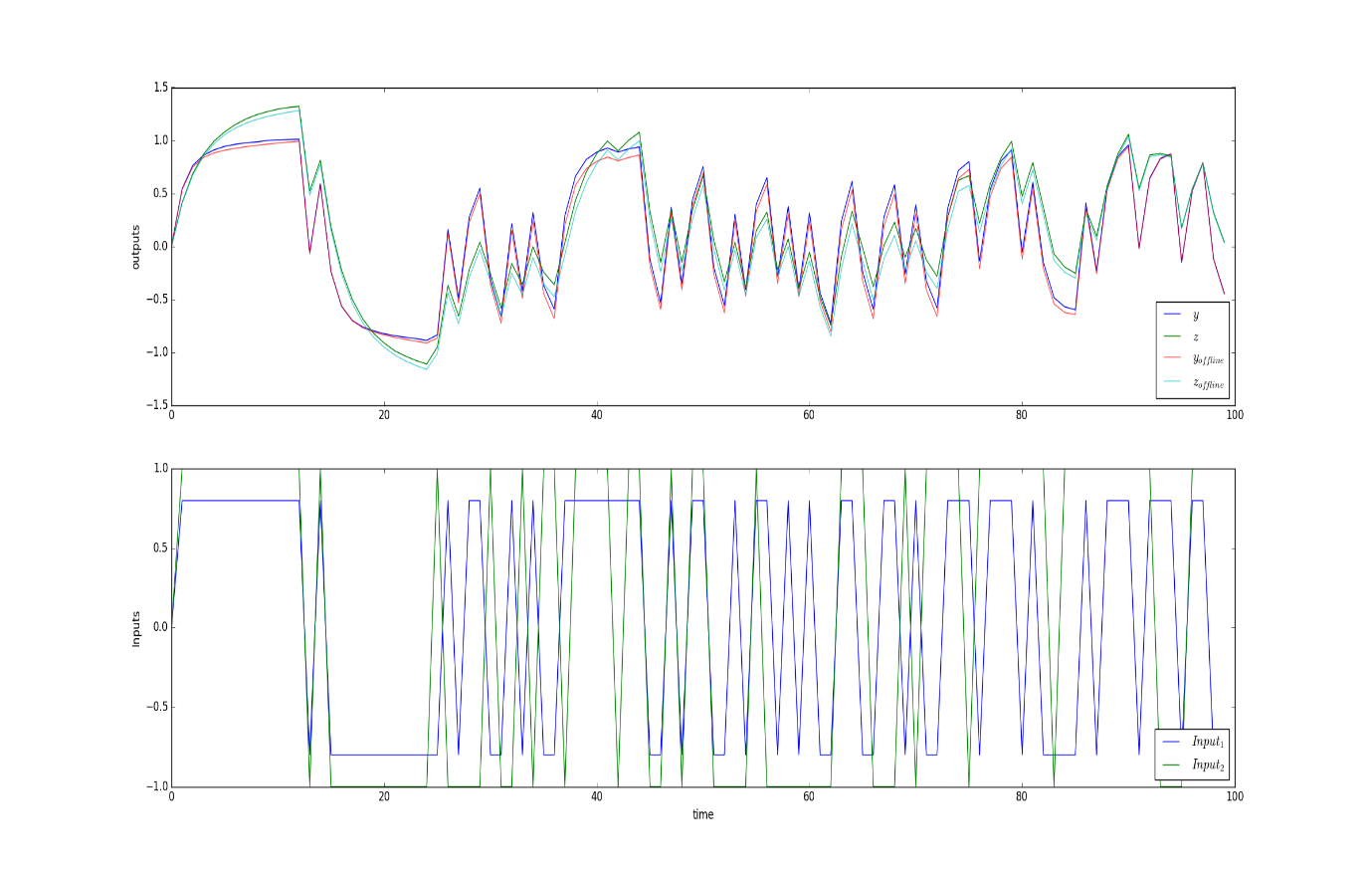
|  |  |  |
| --- | --- | --- |
|  |  | (23) |

The parameter estimates are

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

*Offline Results*

The estimated parameters are expected to converge to the true parameters as given in table() above. The open-loop identification was simulated for 100 time units in offline mode and the response is given in the figure below.



The response above is due to a PRBS signal for both inputs with the properties given below.

|  |  |  |
| --- | --- | --- |
| Input (PRBS) | Mean | Variance (σ2) |
| Input 1 | 0 | 0.8 |
| Input 2 | 0 | 1 |

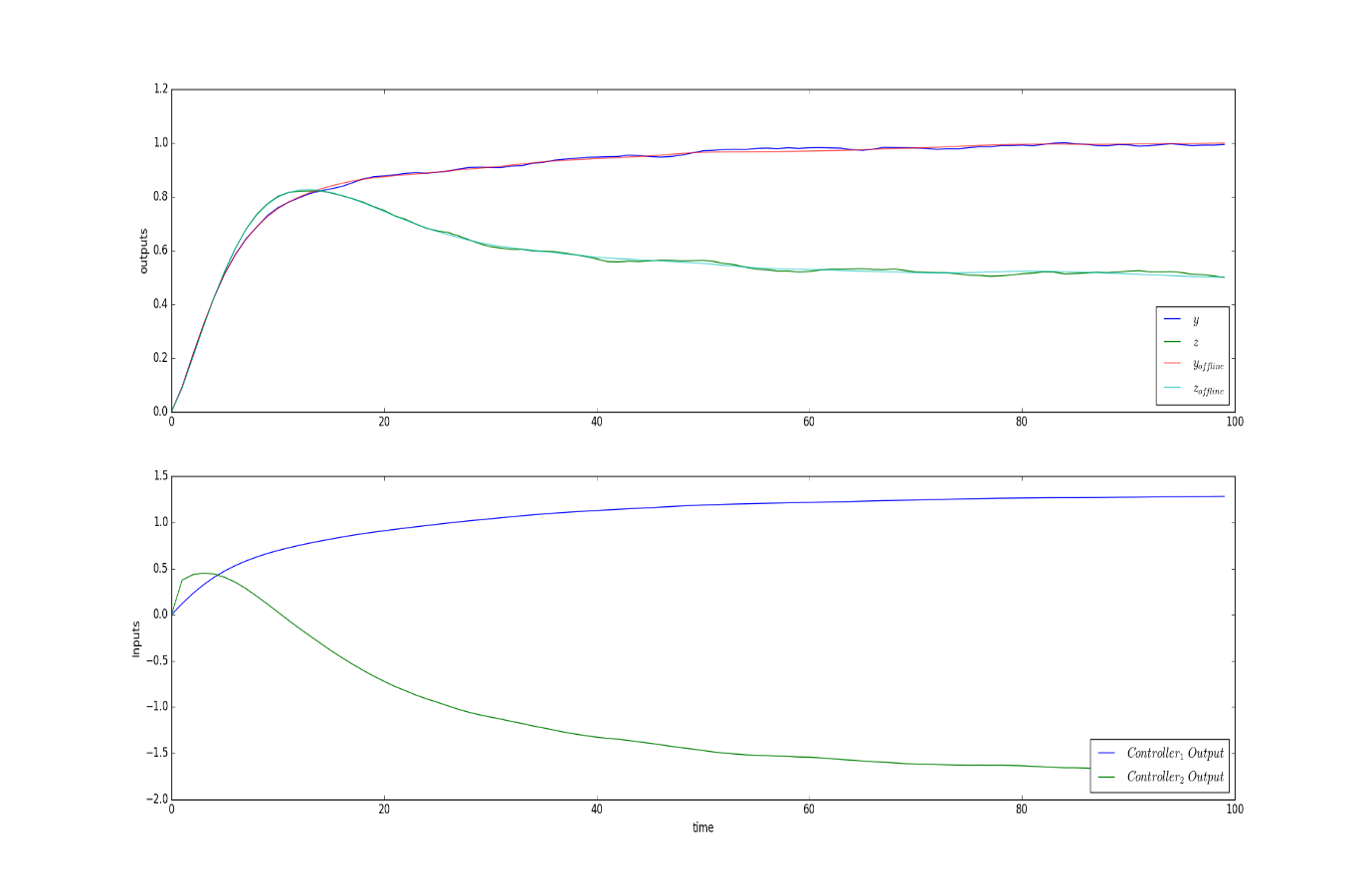
The identification results are given in the table below.

|  |  |  |
| --- | --- | --- |
| Parameter | Estimated Value | True Value |
| a11 (b1 + b3) | 1.280 | 1.214 |
| a22 (-b1b3) | -0.334 | -0.3114 |
| a33 (a1) | 0.631 | 0.6321 |
| a44 (-a1b3) | -0.576 | -0.5351 |
| a55 (a3) | 0.030 | 0.0307 |
| a66 (-a3b1) | -0.0137 | -0.0113 |
| b11 (b4 + b2) | 1.474 | 1.425 |
| b22 (-b4b2) | -0.525 | -0.4966 |
| b33 (a4) | 0.393 | 0.3935 |
| b44 (-a4b2) | -0.341 | -0.3221 |
| b55 (a2) | 0.0906 | 0.0906 |
| b66 (-a2b4) | -0.0599 | -0.05497 |

The controller was implemented, for this run, no known external disturbance was introduced. the setpoints are shown in the table below.

|  |  |
| --- | --- |
| Output | Setpoint |
| 1 | 1.0 |
| 2 | 0.5 |

The model response and the process response are shown in the figure below.

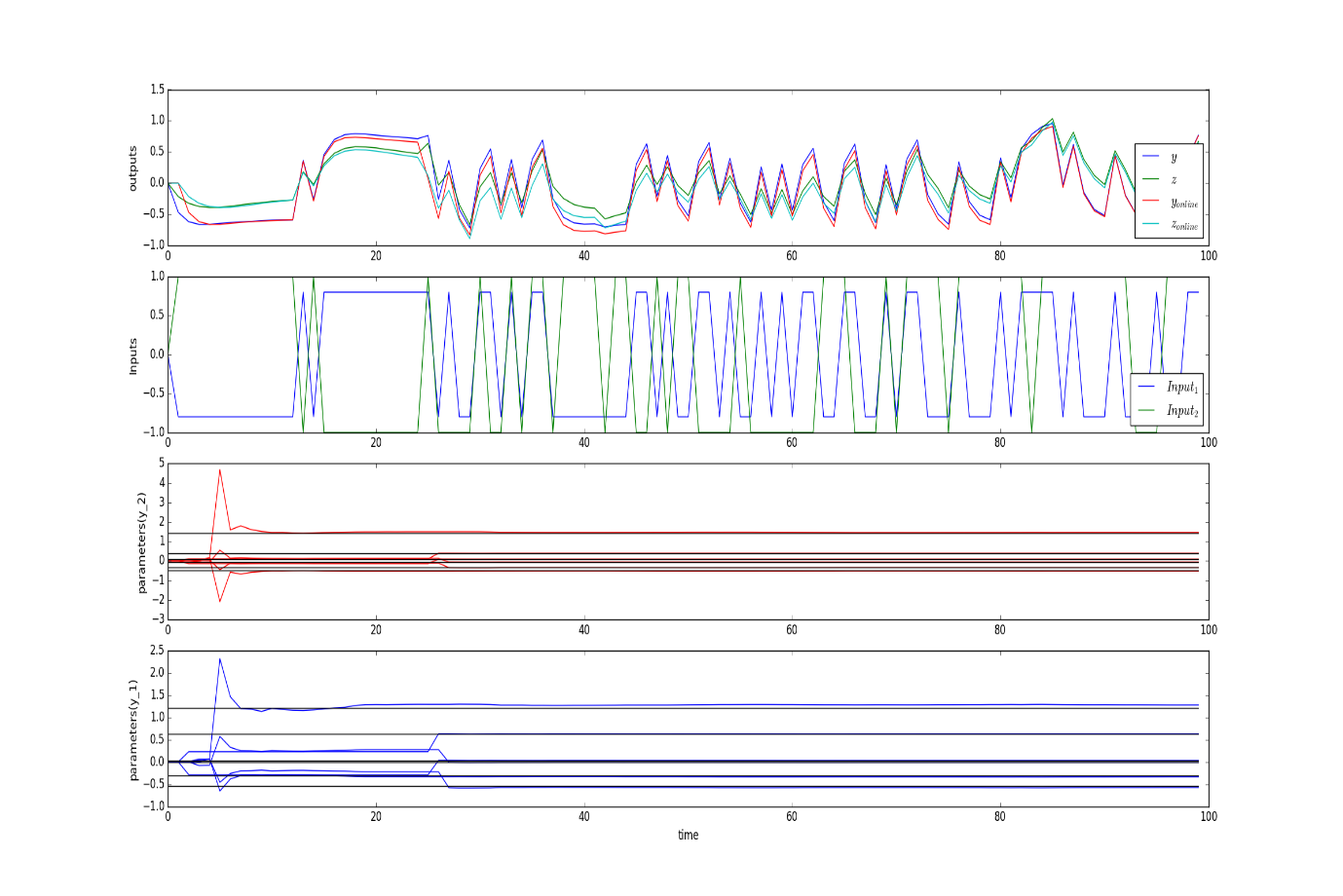


The results for closed-loop-offline identification are summarized in the table below.

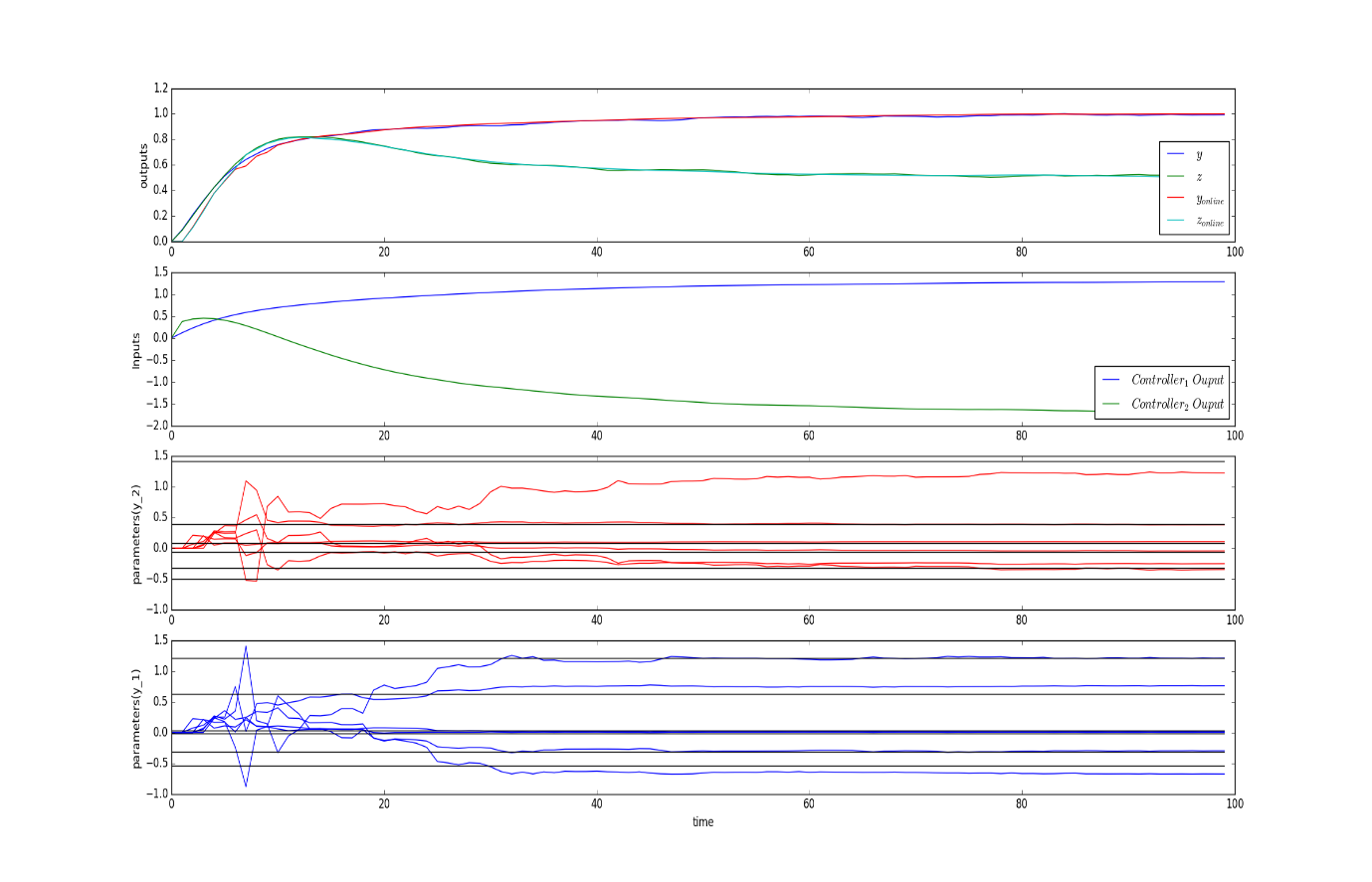
|  |  |  |
| --- | --- | --- |
| Parameter | Estimated Value | True Value |
| a11 (b1 + b3) | 1.2093 | 1.214 |
| a22 (-b1b3) | -0.2992 | -0.3114 |
| a33 (a1) | 0.7629 | 0.6321 |
| a44 (-a1b3) | -0.6758 | -0.5351 |
| a55 (a3) | 0.0031 | 0.0307 |
| a66 (-a3b1) | 0.0099 | -0.0113 |
| b11 (b4 + b2) | 1.2376 | 1.425 |
| b22 (-b4b2) | -0.3507 | -0.4966 |
| b33 (a4) | 0.3853 | 0.3935 |
| b44 (-a4b2) | -0.2496 | -0.3221 |
| b55 (a2) | 0.1121 | 0.0906 |
| b66 (-a2b4) | -0.0434 | -0.05497 |

*Online Results*

The simulation for online identification uses the same input and noise characteristics as the ones in the offline identification. The open-loop results are shown in the figure below.



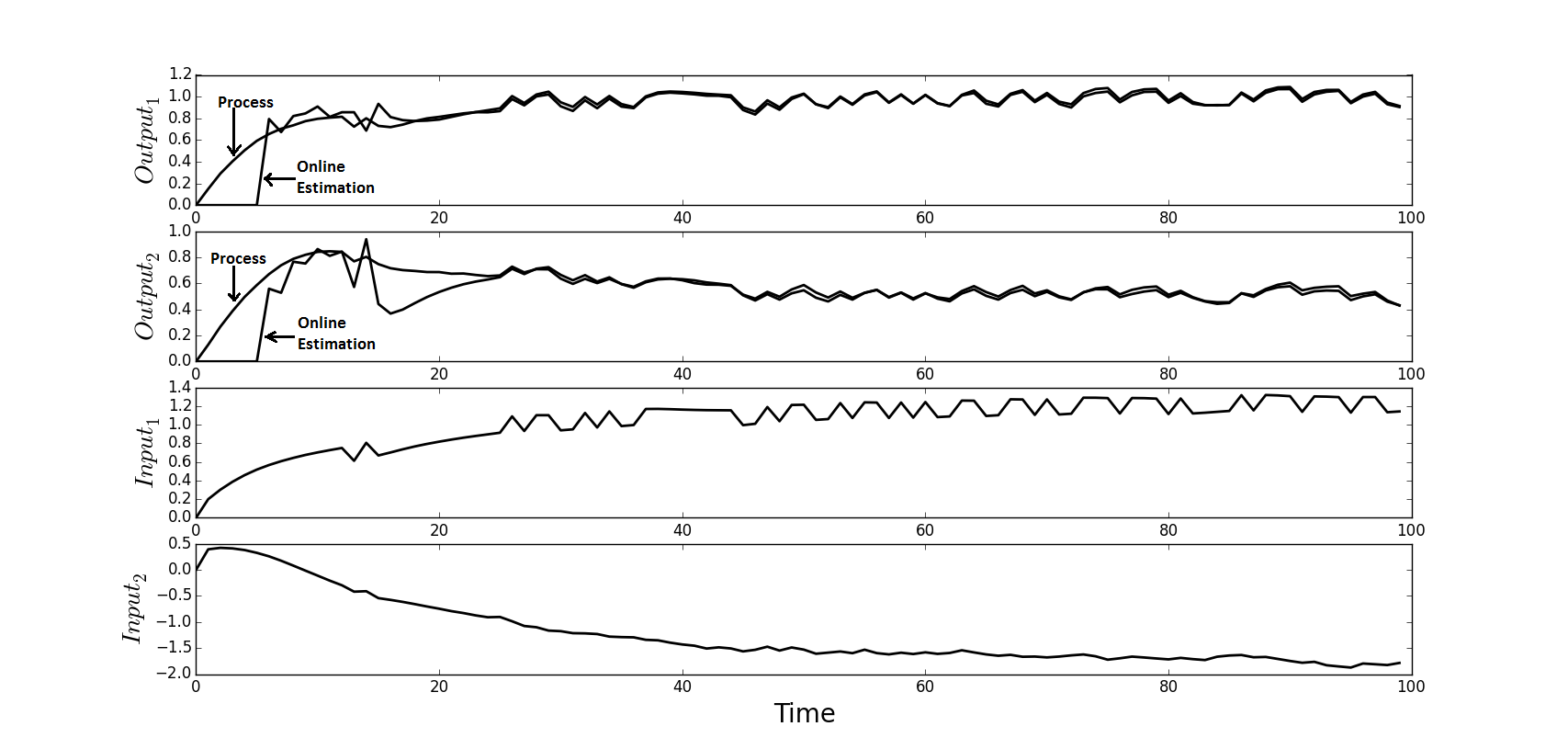
And the closed loop results are

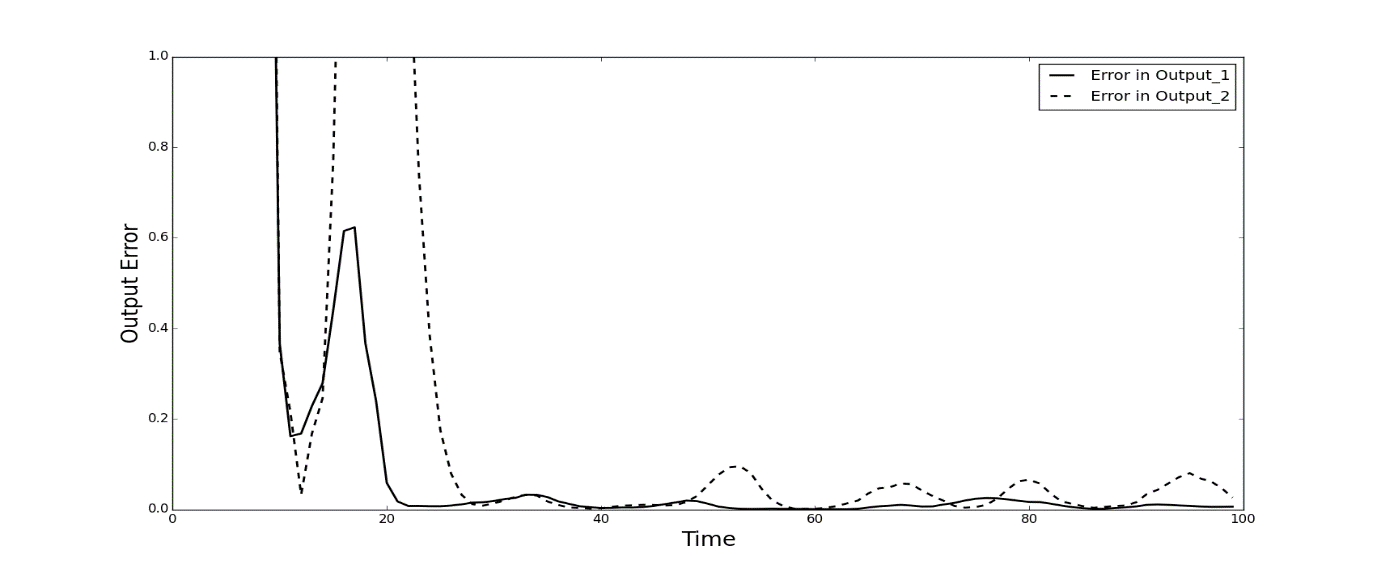


A certain time window was allowed for testing of the online parameters for instance, a 5 time unit window would predict the process output for the next 5 time units and that means the parameters at t=0 will predict a zero output up to 5 time units since the recursive identification is initialized as having all parameters to the value of zero. An external disturbance will also be added to the controller output and the properties of this disturbance are summarized in the table below.

|  |  |  |
| --- | --- | --- |
| Disturbance (PRBS) | Mean | Variance (σ2) |
| 1 | 0 | 0.02 |
| 2 | 0 | 0.1 |

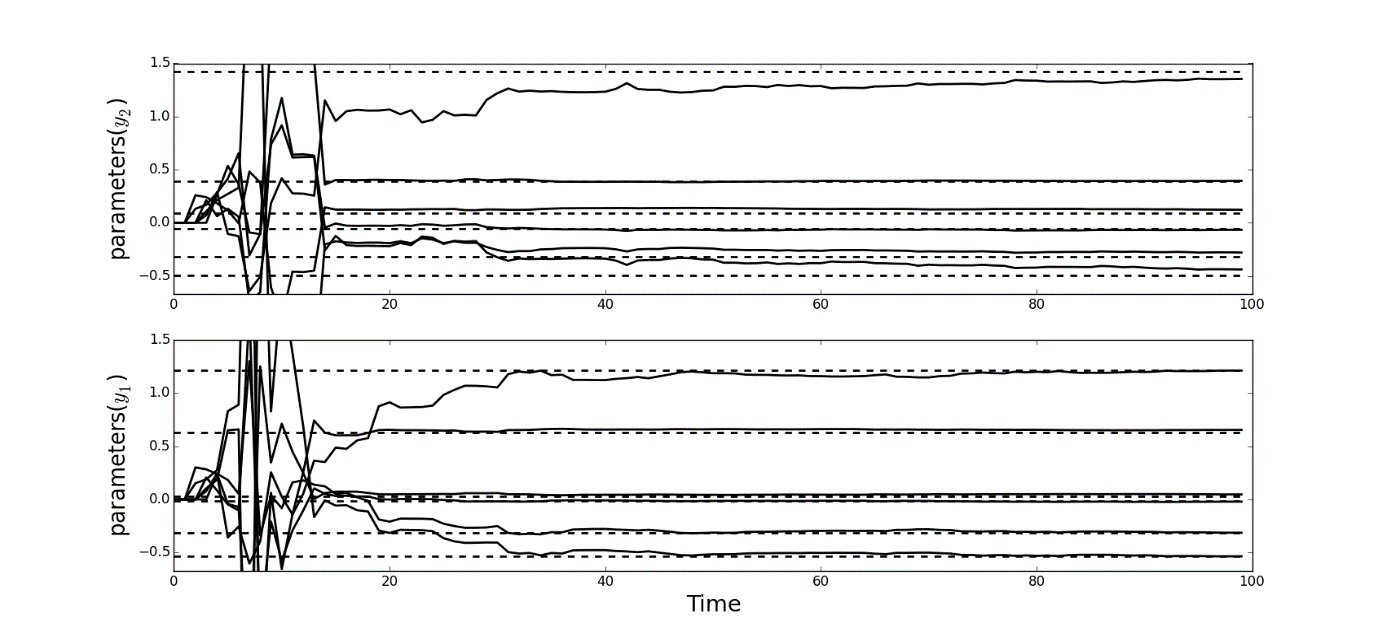
The results of this slight change are shown below.



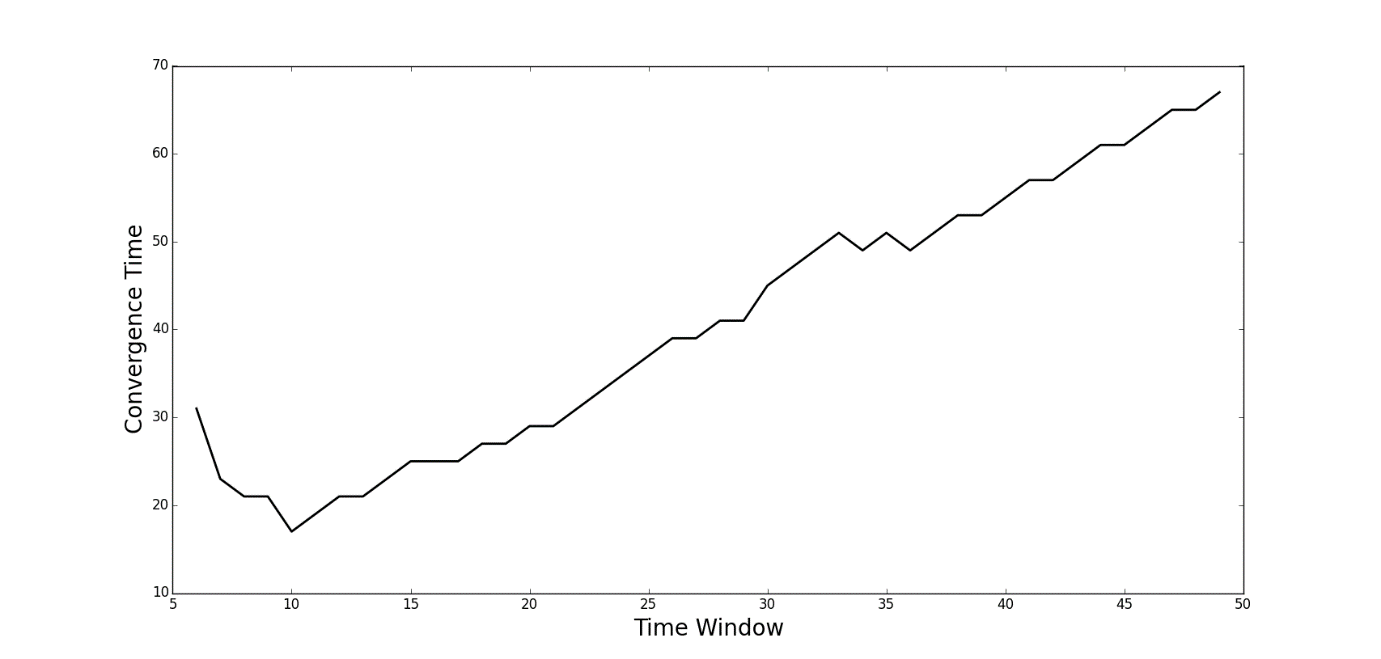


The online identification above allows for prediction of 5 time units ahead. It can be seen that the relative error in the output is minimized with time, reaching a tolerable steady oscillation. This confirms the observed difference in the model output and the process output. The model validation can only be done by this observation since the true parameters are unknown. Thus after 30 time units, the model explains all the patterns in the process output that are due to the input.

The convergence of the estimated parameters is shown below.



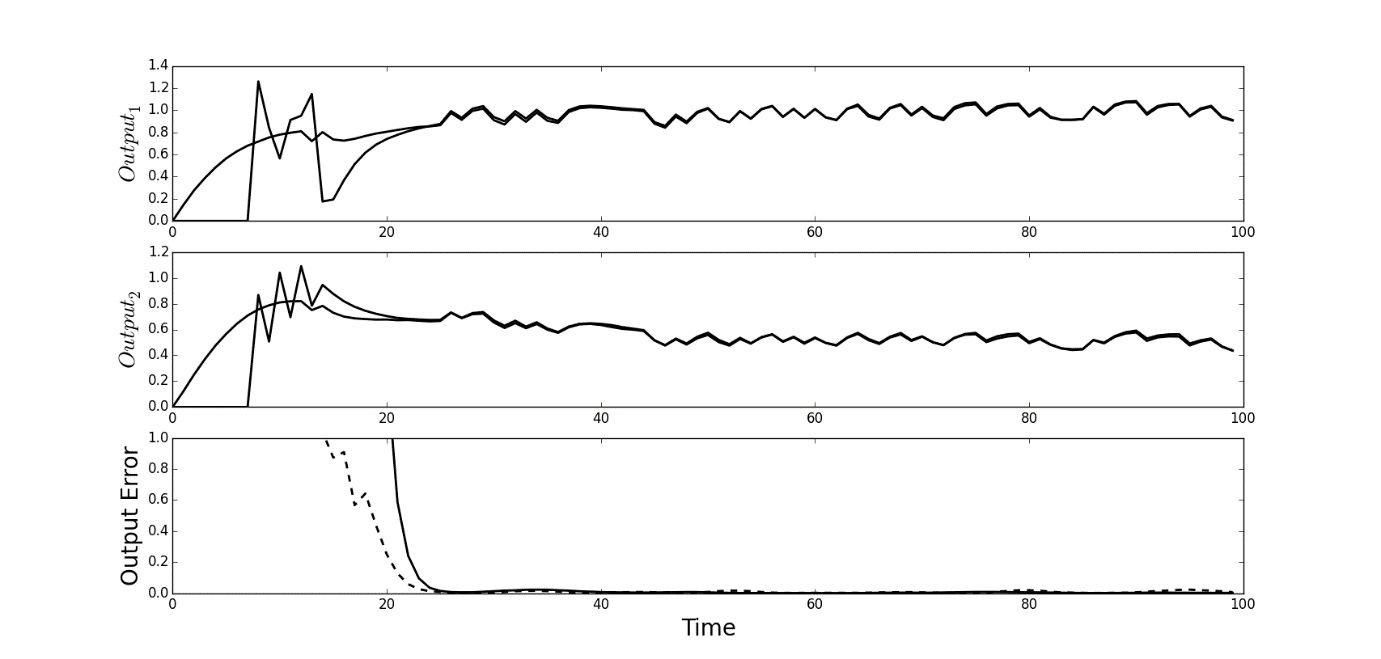
The estimated parameters, after 30 time units, have not converged to the true values yet especially the parameters of the second output. If the identification stops at 30 time units, then it is obvious from the plot above that the estimated parameters at that time are not the best parameters. This is justified by the short time window for testing, it is reasonable to expect high quality estimates if we allow for longer time window.



The convergence time is definitely increasing with more time for testing, but a minimum convergence time is observed at t = 10. This implies that we may end up wait longer for an adequate model even if we use less time for testing.

To illustrate this, an output plot is reproduced for 7 time unit window and 15 time unit window since they appear to have the same convergence time below.

*7 time unit window.*



*15 time unit window*

