# The Matlab Source Code of Comparisons between an ASDB ILC[1] Method and Existing ILC Methods

Zi-Bo Wei, Quan Quan, Kai-Yuan Cai

## Introduction

This file set is a supporting material for our technical note [1]. This note proposes an additive-state-decomposition-based (ASDB) iterative learning control (ILC) method. The note compares our method with an existing ILC method proposed by reference [2] and a classical D-type method. The file set contains the simulations used in this note. The simulation results in reference [3] are reproduced, and they are also contained by this file set. Please use Matlab® to run all of the files in the file set.

## File List and Usage

1. **Folder "Our method--ASDB ILC"**: Use ASDB ILC method[1] to solve the example in our note[1], and plot the simulation results. The Subfolders in this folder are listed as follows:
   a. **Subfolder "Compare the two controllers"**: Compare the effect of two controllers with different parameters, which are designed by using our method. Fig. 2 shows the simulation results. The files in this folder are listed as follows:

      Adjoint_m7_constraint4Resubmit.m,

      Adjoint_Test7.slx,

      Adjoint_Test7_supp.slx.

*Usage:* *Run Adjoint_m7_constraint4Resubmit.m. In the program, use "K_Gs = [2 5 2]" to generate Fig. 2(a)(b), and use "K_Gs = [-1 -5 -1]" to generate Fig. 2(c)(d).*

b. **Subfolder "Robustness of the controller"**: Illustrating the robustness of the controller designed by using our method. Fig. 3 shows the simulation results. The files in this folder are listed as follows:

   Adjoint_m8_constraint4Resubmit.m,

   Adjoint_Test8.slx,

   Adjoint_Test8_supp.slx.

   ***Usage:*** *Run Adjoint_m8_constraint4Resubmit.m.*

2. **Folder "Paper Sogo 2000"**: Use the ILC method proposed by reference [2] to solve the example in our note[1], and plot the simulation results as shown in Fig. 4. The files in this folder are listed as follows:

   Adjoint_m6s_constraint_nASDB.m,

   Adjoint_Test6s_nASDB.slx,

   Adjoint_Test6s_nASDB_supp.slx.

***Usage:*** *Run Adjoint_m6s_constraint_nASDB.m. In the program, use "delta_A = 0" to generate Fig. 4(a) and use "delta_A = -0.1" to generate Fig. 4(b).*

3. **Folder "Classical method"**: Use the classical D-type ILC method to solve the example in our note [1]. The D-type ILC controller is expressed by

$$u_{k+1} = u_k + \alpha \dot{e}_k .$$

   Fig. 5 shows the simulation results. The files in this folder are listed as follows:

   LC_My2_1_Classic_de.m,

   Adjoint_Test6_classic.slx.

   ***Usage:*** *Run LC_My2_1_Classic_de.m.*

4. **Folder "Paper Ghosh 2002"**: Reproduce the simulation results in reference [3]. Fig. 6 shows the simulation results. The files in this folder are listed as follows:

   Main.m,

   Example.slx.

***Usage:*** *Run Main.m.*

Please read the specification in the files to get the further information. If you have any questions, then please feel free to contact Zi-Bo Wei (whisper@buaa.edu.cn) or QuanQuan (qq_buaa@buaa.edu.cn). If you use these files or results in your paper, please cite it as: Zi-Bo Wei, Quan Quan, Kai-Yuan Cai, “The Matlab Source Code of Comparisons between an ASDB ILC Method and Existing ILC Methods”, http://rfly.buaa.edu.cn/, October, 2015.

# Notice

To run these simulations, Matlab® create mex functions, so the corresponding Visio Studio® is needed to compile these simulations. For example, Matlab 2013® needs Visio Studio 2012® or Visio Studio 2010®. Then, run “mex -setup” in matlab to install the mex compiler.

# Reference


[1] Z.-B. Wei, Q. Quan, and K.-Y. Cai, “Output Feedback ILC for a Class of Nonminimum Phase Nonlinear Systems with Input Saturation: An Additive-state-decomposition-based Method.” (Under Review)

[2] T. Sogo, K. Kinoshita, and N. Adachi, “Iterative learning control usingadjoint systems for nonlinear non-minimum phase systems,” *the 39th IEEE Conference on Decision and Control*, pp. 3445-3446, 2000.

[3] J. Ghosh and B. Paden, “A pseudoinverse-based iterative learning control,” IEEE Transactions on Automatic Control, vol. 45, no. 5, pp. 831–837, May 2002.

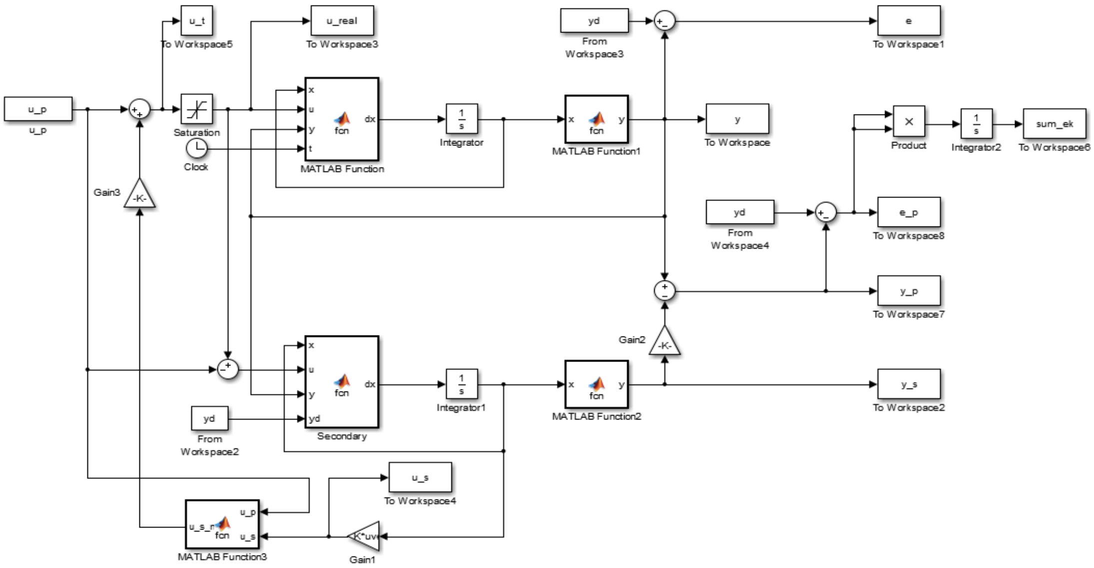


Fig. 1 The simulation structure of our method[1]

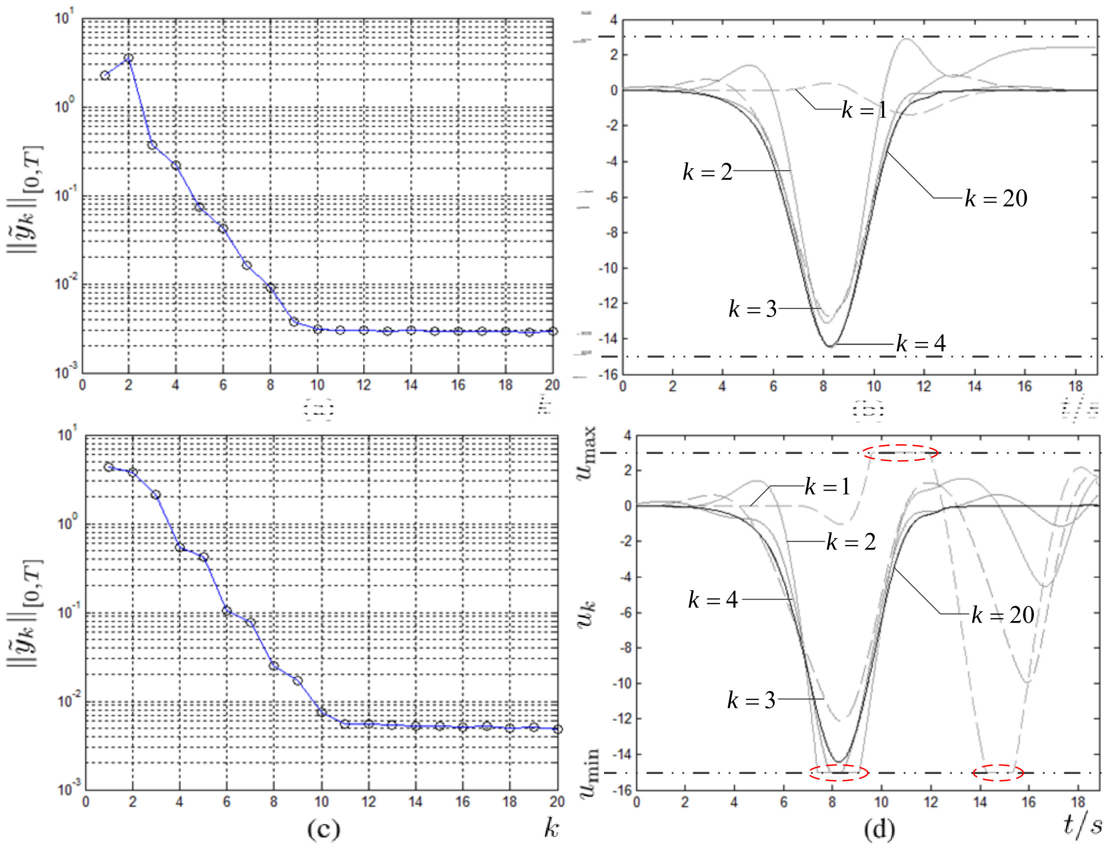


Fig. 2 The simulation results of Adjoint_m7_constraint4Resubmit.m

((a)(b) Convergence of the error of output and the input serial of u1;

(c)(d) Convergence of the error of output and the input serial of u2)

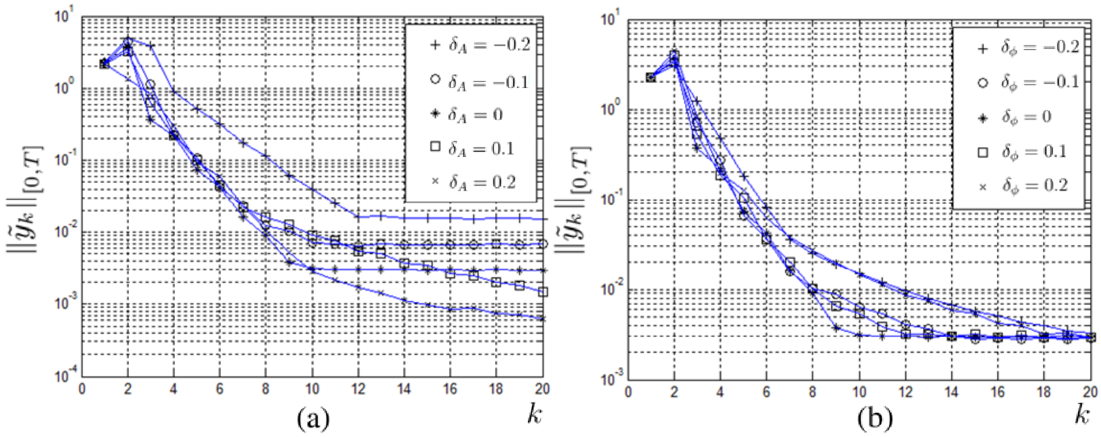


Fig. 3 The simulation results of Adjoint_m8_constraint4Resubmit.m

((a) $\delta_\phi = 0, \delta_A = \{-0.2, -0.1, 0, 0.1, 0.2\}$ (b) $\delta_A = 0, \delta_\phi = \{-0.2, -0.1, 0, 0.1, 0.2\}$ )

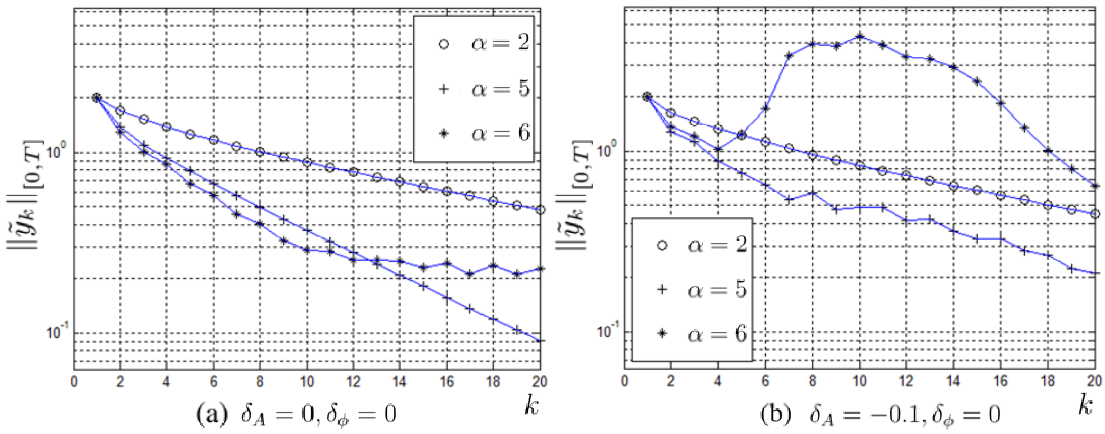


Fig. 4 The simulation results of Adjoint_m6s_constraint_nASDB.m

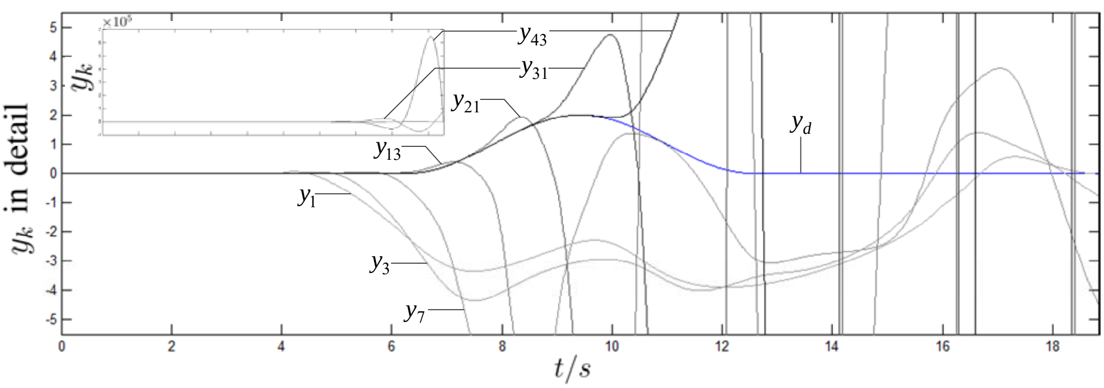


Fig. 5 The simulation results of LC_My2_1_Classic_de.m

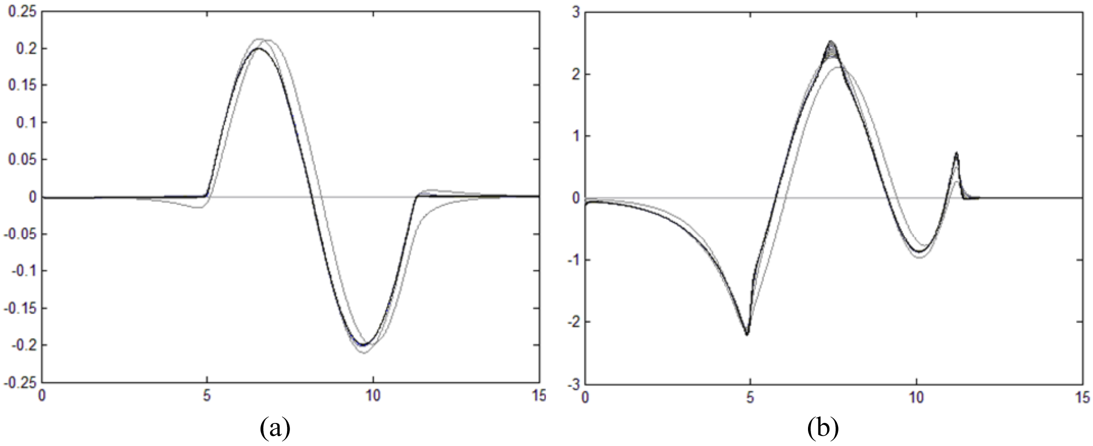


Fig. 6 The simulation results of Main.m ((a) The output serial (b) The input serial)