EE6104 ADAPTIVE CONTROL SYSTEMS (ADVANCED)

Liu Weihao

A0232935A

Continuous Assessment 2

Department of Electrical and Computer Engineering

National University of Singapore

18 April 2023

Contents

1 Introduction

Iterative feedback tuning is a flexible methodology for tuning controllers of arbitrary structures. The key feature is that closed-loop experimental data is used to directly compute a change of the controller parameters such that some performance objective is improved. Since no modelling step is required, the method is relatively simple to use.

In this project, replace the GPC controller with the PI controller and apply IFT to tune the PI parameters for the above semiconductor manufacturing problem. Show through simulation that the PI controller can adapt to new batches of photoresists and wafers. There is no need to consider constraints on the control signal.

2 Problem Formulation

The experimental setup used to control resist thickness consists of three main parts (see Fig. 1): a multizone bakeplate, thickness sensors, and a computing unit. In all our experiments, commercial i-line resist Shipley 3612 is spin-coated at 2000 rpm on a 4-in wafer. Thicknesses at three sites, each 1 in apart, are monitored and controlled to demonstrate the control strategy (see Fig. 2). Fig. 3 shows the photograph of the experimental setup with a 4-in wafer sitting on top of the mutlizone bakeplate. An array of three thickness sensors is mounted directly above the wafer at sites where the resist film thicknesses are being controlled. Currently, the setup is for a 4-in wafer (radius: 2 in; 3 points monitored). This can be easily scaled to a 12-in wafer (radius: 6 in; 7 points monitored). The number of sensors and hence the amount of computation required for 12-in wafer is roughly doubled and this should not be an issue.

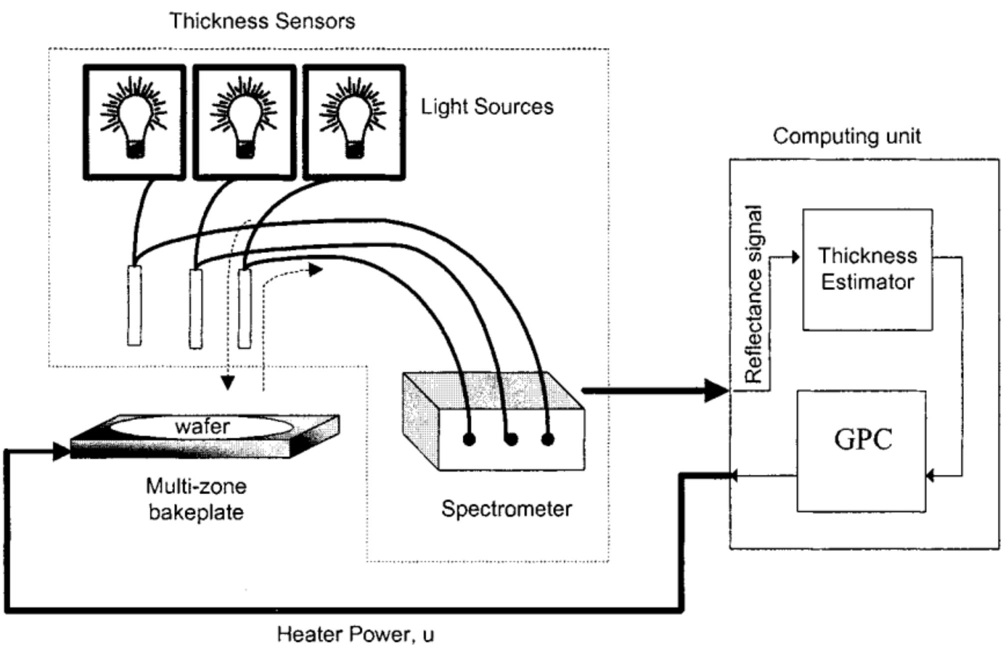

Figure 1: Schematics of the experimental setup used to control resist thickness in real time, which consists of three main parts: a multizone bakeplate, thickness sensors, and a computing unit.

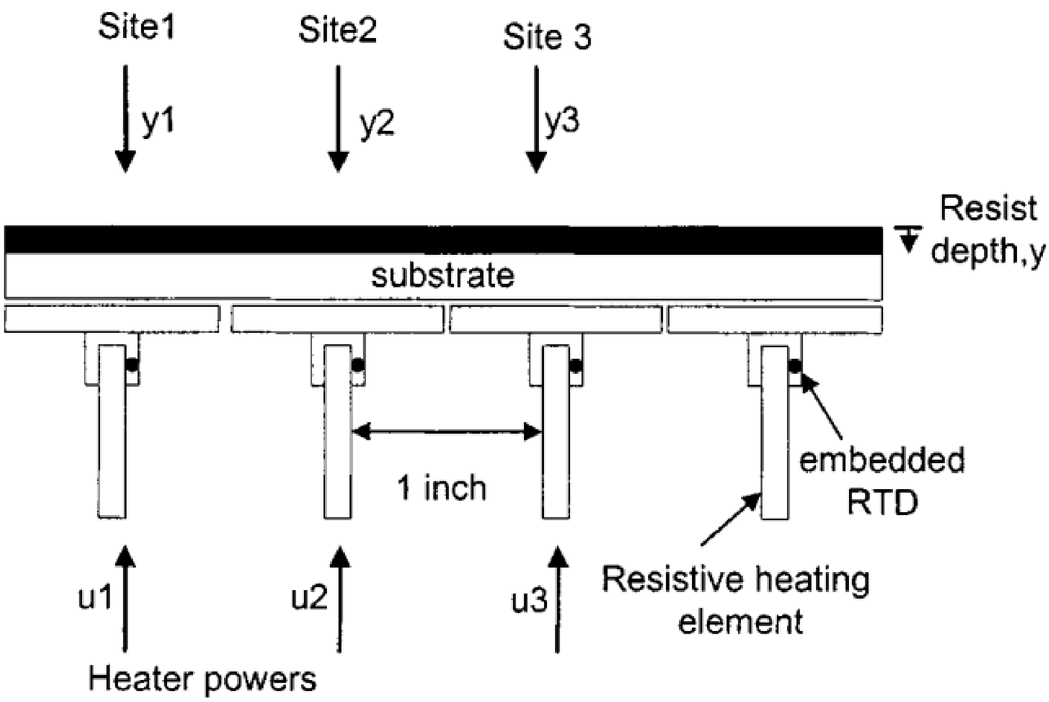

Figure 2: Cross section of the multizone bakeplate.

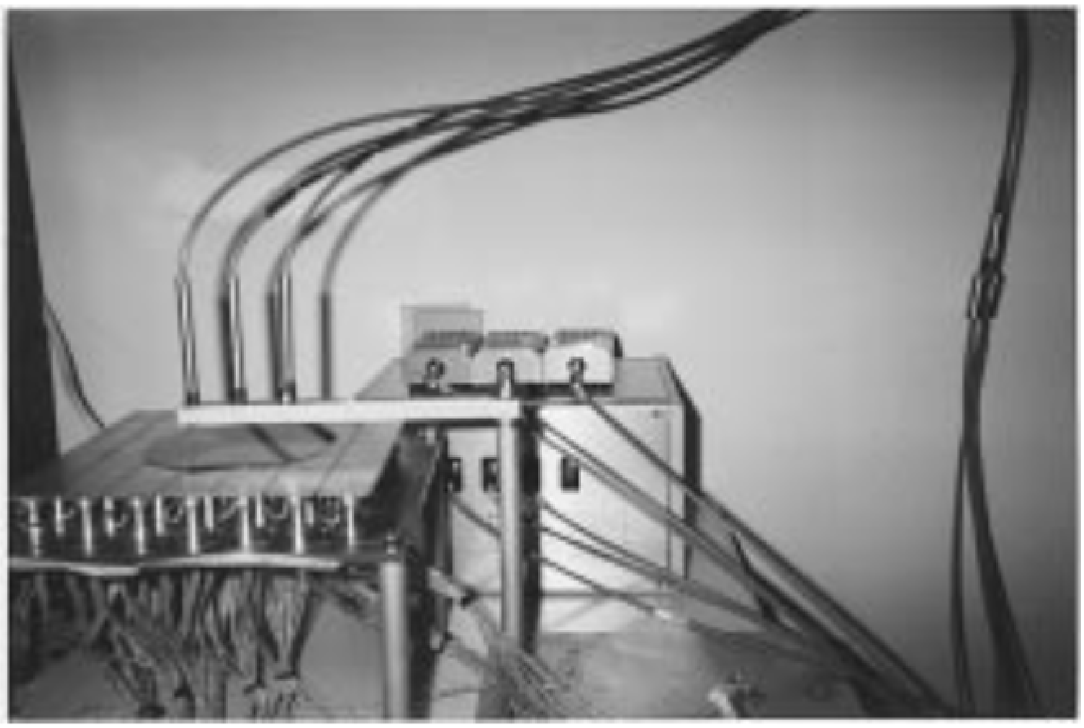

Figure 3: Photograph of the experimental setup.

As it shown in Fig. 2, the resist thickness at Site 1 was monitored while the resist was baked at a nominal processing temperature of $90\,^{\circ}$C. The thickness and temperature are denoted as y_{p2} and T_{p2}, respectively. The difference $(y_{p1} - y_{p2})$ between the thickness measurements, y_1, gives the effect of the heater power on resist thickness. The sampling interval was chosen to be 1 s. Using least squares estimation, the first-order plus dead time process model was identified as eqn. 1

$$Site1 : (1 - aq^{-1})y_1(k) = bq^{-d}u_1(k-1) + \frac{e(k)}{1-q^{-1}} \tag{1}$$

Where $a = 0.9897$, $b = 0.05263$, $d = 15$, $e(k)$ is the zero mean white noise and q^{-1} is the backward shift operator. Then, the eqn 1 can be rewritten as eqn. 2

$$Site1 : (1 - 0.9897q^{-1})y_1(k) = 0.05263q^{-15}u_1(k-1) + \frac{e(k)}{1-q^{-1}} \tag{2}$$

Note:

In this project, we don't need the GPC controllor, all we need from is the system model, 2.

3 Iterative feedback tuning(IFN)

Despite the intuitively appealing ideas behind many of the approaches that iteratively adjust the model fit and, based on this, the controller in order to improve closed control performance, it has turned out, to be difficult to actually prove convergence to the optimal model (and controller). A different approach was taken where it was observed that the model bias problem could be avoided by replacing the information carried by the model by information obtained directly from the system itself. This lead to an iterative method where the controller parameters were successively updated using information from closed loop experiments with the most recent controller in the loop. This approach has since then become known as iterative feedback tuning (IFT). Having thus circumvented the bias problem, it was possible to establish convergence of IFT to a stationary point of the control criterion under the assumptions of a linear and time-invariant (LTI) system and closed loop stability throughout, iterations. The LTI assumption had also dominated the model-based approaches. Rather surprising, simulations and practical experiments soon indicated that, IFT also could improve performance of non-linear systems. This prompted further interest in this method and in this paper we will give an overview of the current state of affairs.

3.1 The performance objective

We consider the SISO discrete time closed loop system in Fig. 4. It holds that

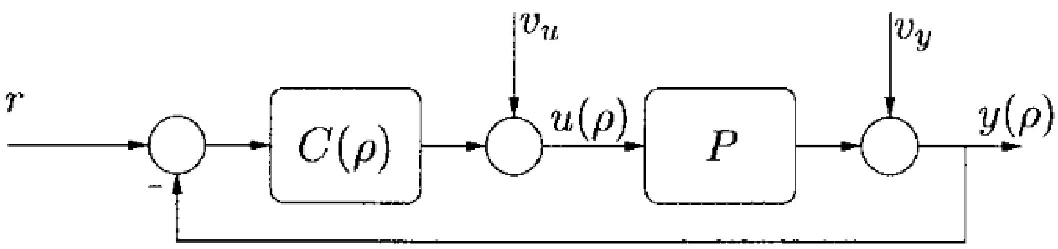

Figure 4: Closed loop system.

$$y(t) = P(q)u(t) + v_y(t) \tag{3}$$

$$u(t) = C(q,\rho)(r(t) - y(t)) + v_u(t) \tag{4}$$

where $P(q)$ is an unknown **SISO LTI** operator(q is the shift operator), $u(t) \in R$ s the process input and $y(t) \in R$ is the corresponding process output. Furthermore, $v(t) = [v_y(t)v_u(t)] \in R^2$ is an unmeasurable (process) disturbance which will be assumed to be stochastic. $C(q,\rho)$ is a linear time-invariant, transfer function parametrized by some parameter vector $\rho \in R^{n_\rho}$ and $r(t) \in R$ is an external deterministic reference signal, independent of $\{v(t)\}$. Whenever signals are obtained from the closed loop system with the controller $C(q,\rho)$ operating, we will indicate this by using the ρ-argument. The time, t, and the shift operator, q, arguments will from now on be omitted whenever not needed. We will assume that the parametrization of the controller is such that all signals in the system are differentiable w.r.t. to ρ.

Let $T_0(\rho)$ and $S_0(\rho)$ denote the achieved closed loop response and sensitivity function with the controller $C(\rho)$, i.e.

$$T_0(\rho) = \frac{PC(\rho)}{1+PC(\rho)}, S_0(\rho) = \frac{1}{1+PC(\rho)} \tag{5}$$

Let y^d be a desired output response to the reference signal r for the closed loop system, generated for instance by a reference model T^d, i.e. $y^d = T^d r$. The error between the achieved and the desired response is

$$\widetilde{y}(\rho) \triangleq y(\rho) - y^d \tag{6}$$

It is natural to formulate the control design objective as a minimization of some objective function of $\widetilde{y}(\rho)$. In this section we will, for reasons of simplicity, use the quadratic function

$$J(\rho) = \frac{1}{2N}\sum_{t=1}^{N} E[\widetilde{y}(t,\rho)^2] \tag{7}$$

where $E[.]$ denotes expectation w.r.t. the disturbance v.

3.2 A necessary condition for optimality

A necessary condition for optimal performance is that the first derivative of the objective function eqn. 7 w.r.t. the controller parameter ρ is zero:

$$\frac{dJ}{d\rho}(\rho) = \frac{1}{N}\sum_{t=1}^{N} E[\widetilde{y}(t,\rho)\frac{\partial\widetilde{y}}{\partial\rho}(t,\rho)] = \frac{1}{N}\sum_{t=1}^{N} E[\widetilde{y}(t,\rho)\frac{\partial y}{\partial\rho}(t,\rho)] = 0 \tag{8}$$

Hence, to achieve optimal performance it is necessary to be able to detect this condition for any controller which means that (approximations of) the following quantities are required:

1. the signal, $\widetilde{y}(t,\rho)$
2. the gradient, $\partial y/\partial\rho(t,\rho)$

The signal $\widetilde{y}(\rho)$ can easily be obtained only by measuring the output of the closed loop system when the reference r is exciting the system and subtracting the measured output from the desired response y^d. It is the gradient 2 that gives rise to problems. To understand the reason for this observe that, differentiation of eqns. 3 and 4 gives

$$y'(t,\rho) = P(q)u'(t,\rho) \tag{9}$$

$$u'(t,\rho) = C'(q,\rho)(r(t) - y(t,\rho)) - C(q,\rho)y'(t,\rho) \tag{10}$$

3.3 Gradient-based minimization

Many of the early adaptive control methods were directly based on the gradient dJ =dr and used a gradient search of the type

$$\rho(i+1) = \rho(i) - \gamma_i R^{-1}(i) \frac{d\widehat{J}}{d\rho}(\rho(i)) \tag{11}$$

Here $d\widehat{J}/d\rho$ is an approximation of the gradient, R_i is some appropriate positive definite matrix while γ_i is a positive real scalar that determines the step size.

4 ITERATIVE FEEDBACK TUNING

About 30 years later, the idea of using the system itself to generate the signal $T_0.(\rho)(r-y(\rho))$ reappeared. We can use the true closed loop system to generate $y^{'}$ and $u^{'}$. However, using the configuration requires one closed loop experiment per element in ρ. In order to get a more efficient procedure it was observed that for a **SISO LTI** system, for which commutativity holds, we can use Fig. 5. This led to the following basic algorithm which estimates the gradient $\partial y/\partial\rho$. What we need in this project if shown in Fig. 6

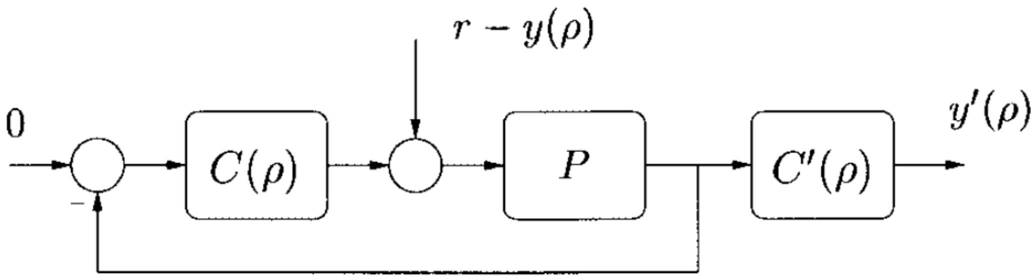

Figure 5: Closed loop system corresponding to gradient.

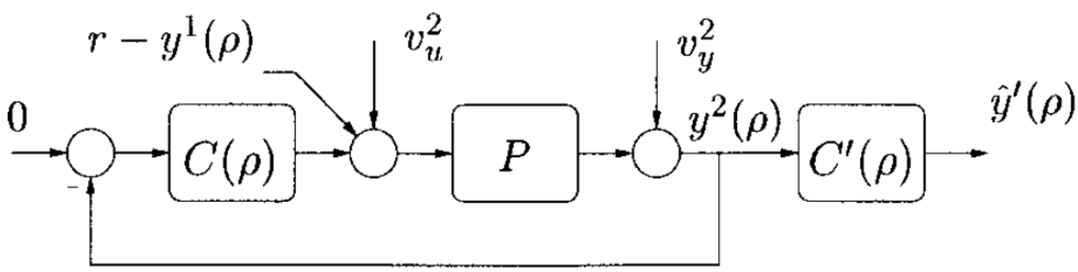

Figure 6: Gradient experiment.

Take as gradient approximation

$$\frac{\widehat{\partial y}}{\partial\rho}(\rho) = \frac{\partial C}{\partial\rho}(\rho) y^2(\rho) \tag{12}$$

The iterative method obtained by using eqn. 12 in the gradient estimate of the objective function and then iterating is one basic version of iterative feedback tuning (IFT).

$$\frac{\widehat{\partial J}}{\partial\rho}(\rho) = \frac{1}{N}\sum_{t=1}^{N} E[\widetilde{y}(t,\rho)\frac{\widehat{\partial y}}{\partial\rho}(t,\rho)] \tag{13}$$

The matrix $R(i)$ in eqn. 11 determines the update direction and is therefore crucial for the performance of the algorithm. A good choke in general is to let $R(i)$ be an approximation of the Hessian. Especially $\widetilde{y}$ is small, the Gauss–Newton direction is shown as eqn. 14

$$\frac{1}{N}\sum_{t=1} N\frac{\partial y}{\partial \rho}(\rho(i))\frac{\partial y^T}{\partial \rho}(\rho(i)) \tag{14}$$

5 Implementation

In this project, we need to use PI controller replace the GCP controller and control the PI controllor can be expressed as eqn. 15.

$$C(z) = K_p + K_i\frac{T_i}{1-z^{-1}} \tag{15}$$

So, what we need to do is using the eqns. 1 and 15 to replace the C and P in eqns. 3 and 4. Besides, because our PI controller only have two parameter can be tunned, when $d = 15$, the order of the system is to high that can be controlled. So, in this project, we set $d = 2$. The initial PI parameter is: $K_p = 20$, $K_i = 20$. The simulink structure is shown as Fig. 7. The experiment result is shown as Fig. 8. We can get that, at begining, the initial parameter of PI controller is not stable, the blue line always can not converge. However, after we apply the IFT, the output of the system can follow the idle model after some time to tune the PI parameters.

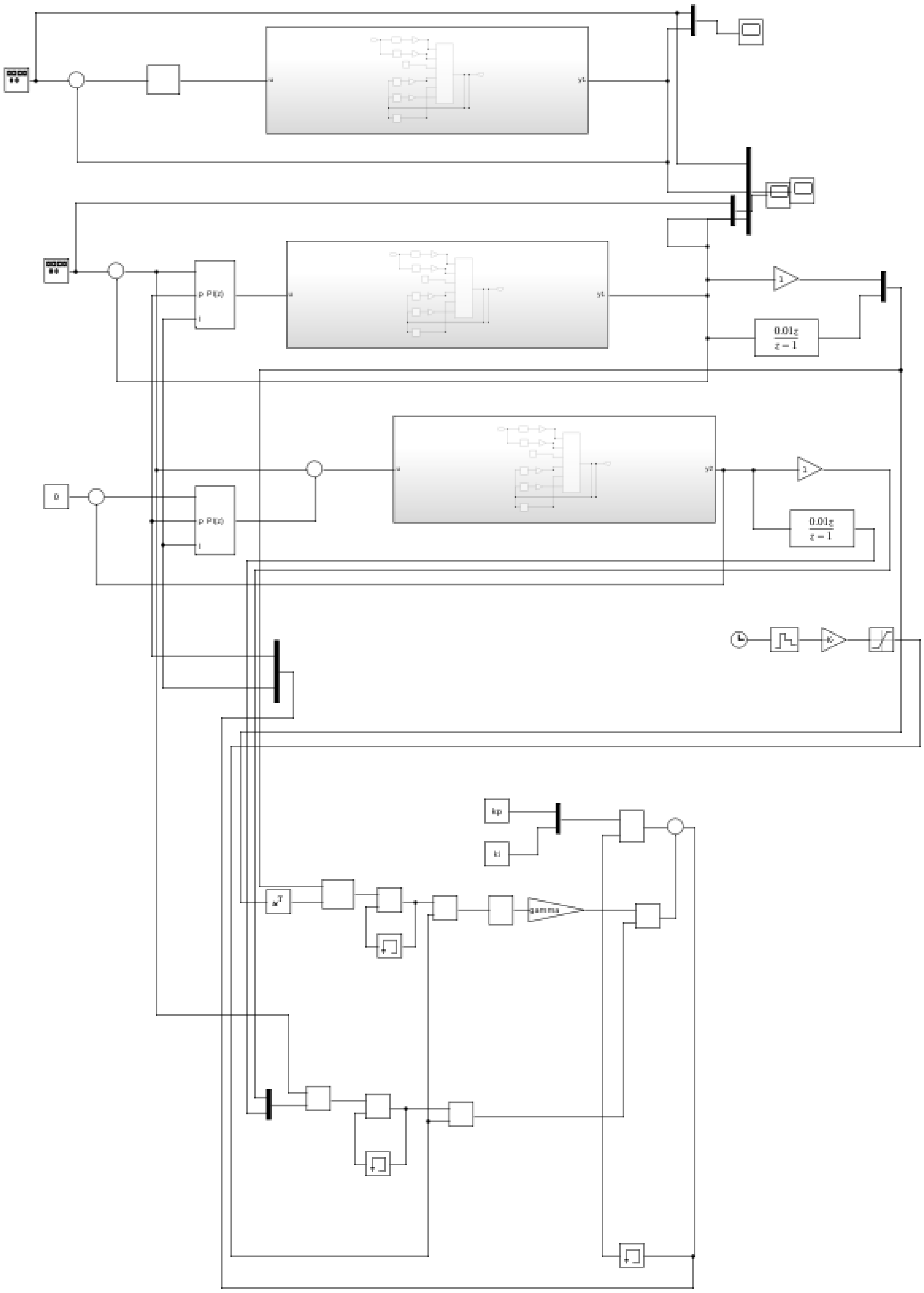

Figure 7: Simulink Structure.

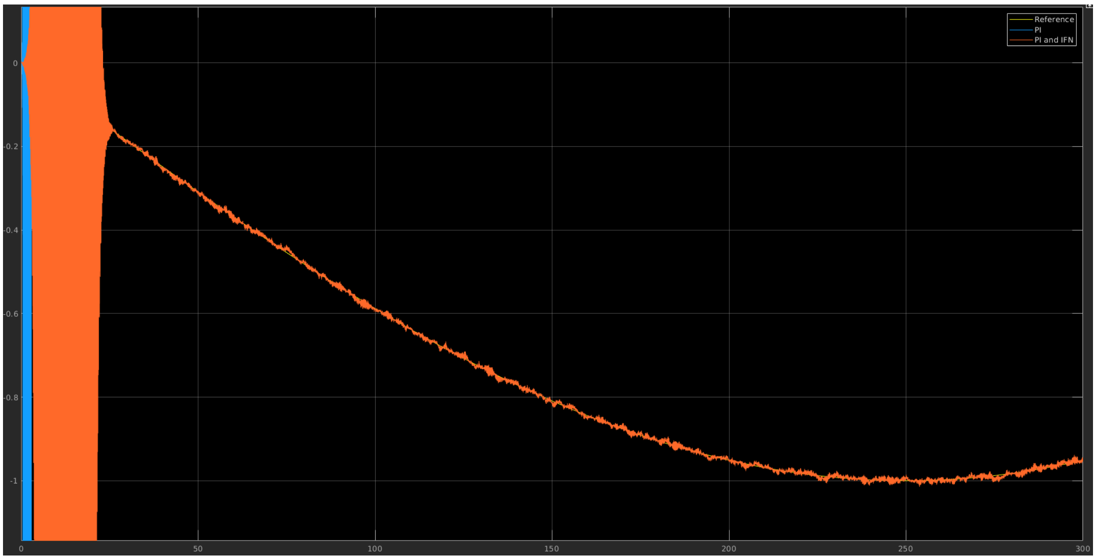

Figure 8: IFN result.