EE6104 CA2 Report: Adaptive PI Controller Design using IFT

Hongxuan Wang
A0228929M
hongxuanwang@u.nus.edu

Abstract

A proportional-integral (PI) controller is used to improve the resist thickness control and uniformity of the softbake process in the area of semiconductor. The iterative feedback tuning (IFT) algorithm is implemented to autotune the parameters of the PI controller, for the controller to adapt to different batches of photoresist and wafers. Simulations are conducted using MATLAB Simulink. Intermediate data and MATLAB/Simulink files are available at: `https://github.com/hxwangnus/EE6104-Adaptive-Control.git`.

1 Introduction

In this project, the photoresist film thickness is controlled during the softbake process, using an adaptive proportional-integral (PI) controller. Iterative feedback tuning (IFT) is implemented as the self-tuning method, which is essentially a data-driven control optimization process, without the need for the plant information. An initial controller generates the initial control signal, which is used to control the plant and get a plant output. A performance criterion is calculated based on the plant output, and the optimization mechanism updates controller parameters based on the performance criterion. In IFT, the performance criterion is a quadratic function related to the output tracking error and the control signal, and the parameters update strategy is gradient descent.

2 IFT method implementation

IFT is used to update the parameters of the PI controller, $C(K_c, T_I)$, where K_c is the proportional gain and T_I is the integral time:

$$C(K_c, T_I) = K_c(1 + \frac{1}{sT_I}), \tag{1}$$

where s is the complex frequency variable in the Laplace transform.

The error function which is the input of the controller is defined as the difference between the thickness measurement y and the reference thickness r:

$$e = y - r, \tag{2}$$

and thus the performance criterion can be defined as [1]:

$$J(K_c, T_I) = \frac{1}{2N}(\sum_{t=1}^{N} e_t^2 + \eta \sum_{t=1}^{N} u_t^2), \tag{3}$$

where N is the discrete time sequence and e_t, u_t denote the error signal and control signal value at time t. The first term is the tracking error, and the second term is the penalty of the control signal.

The parameters update strategy is to minimize J over K_c and T_I using gradient descent:

$$\frac{\partial J}{\partial K_c} = \frac{1}{N}(\sum_{t=1}^{N} e_t \frac{\partial e_t}{\partial K_c} + \eta \sum_{t=1}^{N} u_t \frac{\partial u_t}{\partial K_c}), \tag{4}$$

$$\frac{\partial J}{\partial T_I} = \frac{1}{N}(\sum_{t=1}^{N} e_t \frac{\partial e_t}{\partial T_I} + \eta \sum_{t=1}^{N} u_t \frac{\partial u_t}{\partial T_I}), \tag{5}$$

and then the parameters can be updated by:

$$\rho_{i+1} = \rho_i - \gamma R_i^{-1} J'(\rho_i), \tag{6}$$

where ρ is $[K_c, T_I]^T$, γ is a positive real scalar and R is an appropriate positive definite matrix:

$$R_i = \frac{1}{N} \sum_{t=1}^{N} (\frac{\partial e_t}{\partial \rho} \frac{\partial e_t}{\partial \rho}^T + \eta \frac{\partial u_t}{\partial \rho} \frac{\partial u_t}{\partial \rho}^T). \tag{7}$$

Considering the inner terms of Eq. (4), (5) and (7), according to [1], there are:

$$\frac{\partial e}{\partial K_c} = -\frac{1}{K_c} \frac{CP}{1+CP} e, \tag{8}$$

$$\frac{\partial e}{\partial T_I} = -\frac{1}{T_I(sT_I+1)} \frac{CP}{1+CP} e, \tag{9}$$

and

$$\frac{\partial u}{\partial K_c} = -\frac{1}{K_c} \frac{C}{1+CP} e, \tag{10}$$

$$\frac{\partial u}{\partial T_I} = -\frac{1}{T_I(sT_I+1)} \frac{C}{1+CP} e. \tag{11}$$

Eq.(4), (5) and (7) can be solved based on Eq. (8) - (11), then according to Eq. (6), the controller parameters will be properly updated.

3 Simulation using MATLAB and Simulink

Simulations are carried out in this section. The MATLAB codes are attached as follows, part of the parameters initialization are set the same as [1]:

```
clc
clear all;

%% Define initial parameters for the PI controllers
Kc = 5e6; % proportional gain
TI = 500; % integral time
gamma = 0.5; % step size
eta = 1.2; % weight factor in cost function

%% Define thickness
rn = 8750; % in nm, target thickness
y0 = 10000; % initial thickness, manually chosen from the paper

reference = load("reference signal2.mat");
open_system('CA2_model');
out = sim('CA2_model.slx');

%% IFT

s = tf('s'); % Define the symbolic variable for s (Laplace transform
    variable)
```

```
% Calculation need for J(Kc, TI) and R.
sum_error = 0; % sum of error
sum_yydyyKc = 0; % sum of yy*(dyy/dKc)
sum_yydyyTI = 0; % sum of yy*(dyy/dTI)
sum_uduKc = 0; % sum of u*(du/dKc)
sum_uduTI = 0; % sum of u*(du/dTI)
sum_dyy2Kc = 0; % sum of (dyy/dKc)^T * (dyy/dKc)
sum_dyy2TI = 0; % sum of (dyy/dTI)^T * (dyy/dTI)
sum_du2Kc = 0; % sum of (du/dKc)^T * (du/dKc)
sum_du2TI = 0; % sum of (du/dTI)^T * (du/dTI)

for i = 1:10 % number of iterations

    out = sim('CA2_model.slx');

    P = 4e-6 / (100 * s + 1); % Plant transfer function
    C = Kc * (1 + 1/(s * TI)); % Controller transfer function

    % simulate the controller output and plant response
    u = out.u.data; % PI controller output
    y = out.y.data; % Process output response
    r = out.r.data;
    time = out.y.time;

    for t = 1:length(time)
        % Calculate the gradients using numerical differentiation
        ye = -y(t);
        ue = -u(t);
        yy = y(t) - r(t); % thickness error

        dyy_dKc = (1/Kc) * ye;
        % dyy_dTI = (1/(TI * (s*TI + 1))) * ye;
        dyy_dTI = (1/TI) * ye; % Calculate at s=0;
        du_dKc = (1/Kc) * ue;
        % du_dTI = (1/(TI * (s*TI + 1))) * ue;
        du_dTI = (1/TI) * ue;

        % Calculate all sums
        sum_yydyyKc = sum_yydyyKc + yy * dyy_dKc; % sum of yy*(dyy/dKc)
        sum_yydyyTI = sum_yydyyTI + yy * dyy_dTI; % sum of yy*(dyy/dTI)

        sum_uduKc = sum_uduKc + u(t) * du_dKc; % sum of u*(du/dKc)
        sum_uduTI = sum_uduTI + u(t) * du_dTI; % sum of u*(du/dTI)

        sum_dyy2Kc = sum_dyy2Kc +  dyy_dKc.' * dyy_dKc;   % sum of (dyy/dKc)^T * (dyy/dKc)
        sum_dyy2TI = sum_dyy2TI + dyy_dTI.' * dyy_dTI; % sum of (dyy/dTI)^T * (dyy/dTI)
        sum_du2Kc = sum_du2Kc + du_dKc.' * du_dKc; % sum of (du/dKc)^T * (du/dKc)
        sum_du2TI = sum_du2TI + du_dTI.' * du_dTI; % sum of (du/dTI)^T * (du/dTI)
    end

    % Gradient of the cost function J
    J_prime_Kc = (sum_yydyyKc + eta * sum_uduKc) / length(time);
    J_prime_TI = (sum_yydyyTI + eta * sum_uduTI) / length(time);

    % Positive definite matrix R
    R_Kc = (sum_dyy2Kc + eta * sum_du2Kc) / length(time);
    R_TI = (sum_dyy2TI + eta * sum_du2TI) / length(time);

```

```
    % Update controller parameters using the gradient
    Kc = Kc - gamma * inv(R_Kc) * J_prime_Kc;
    TI = TI - gamma * inv(R_TI) * J_prime_TI;

    % Display current controller parameters and error
    fprintf('Iteration %d: Kc = %.2f, TI = %.2f, Error = %.2f nm\n',
        i, Kc, TI, yy);
end
```

The Simulink model is shown in Figure 1. The PI controller is simulated according to Eq. (1), and the plant is a manually designed first-order system. The reference signal r and the plant parameters will be changed, in order to explore the adaptive effectness.

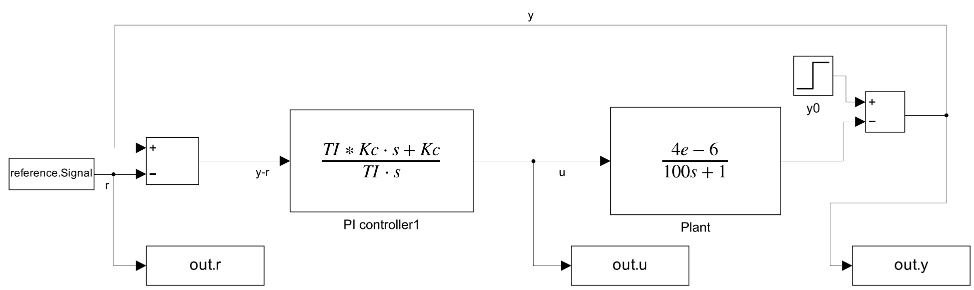

Figure 1: Simulink model for the adaptive PI controlled system.

The initial simulation is conducted with:

$$P = \frac{4e-6}{100s+1}, \tag{12}$$

and

$$r = 1333.33e^{-0.01733t} + 8666.67, \tag{13}$$

which is to simulate the condition in [1], where the target thickness is $8750nm$ and the original thickness is $10000nm$, and test time is $300s$. The simulation results are shown in Figure 2.

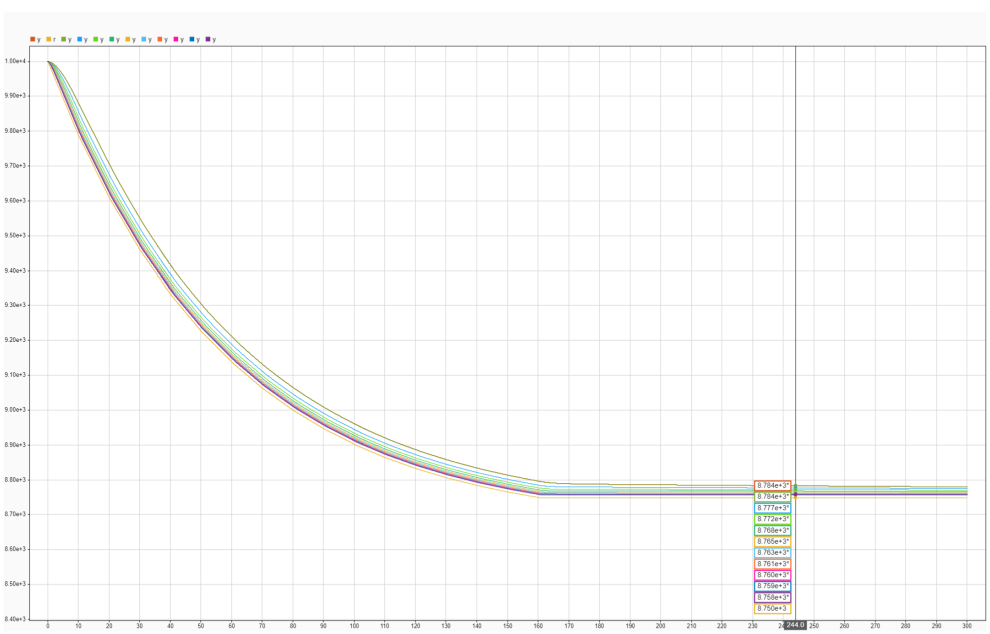

Figure 2: Initial simulation using default parameters.

According to [1], if the difference between y and r is less than $10nm$, the tracking curve is said to be converged. Here in the simulation, we only focus on the steady-state thickness, and define convergence such that the difference between the steady-state thickness and the target thickness is less than $10nm$. It can be observed that the measured thickness controlled by the initial controller had a $34nm$ difference compared with the target thickness, and after 8 iterations, the difference is less than $10nm$. If the step size γ is larger, the speed arriving optimum will be faster.

The controller gains are shown in Table 1.

Table 1: Controller gains in the initial simulation.

Gains	**Initial**	$1st$ **iteration**	$8th$ **iteration**
K_c	5e6	7.5e6	33.2e6
T_I	500	750	3317.5

The adaptive system should adapt to different plants and different initial thicknesses and target thicknesses. They will be explored in the next sections.

3.1 Photoresist thickness change

When the photoresist thickness changes, e.g., the initial thickness or target thickness changes, the reference signal will change, and then the IFT will adjust the controller parameters to adapt to the new reference signal.

3.1.1 Simulation with initial thickness $= 11000nm$ and target thickness $= 8750nm$

The reference signal is adjusted accordingly,

$$r = 2400e^{-0.01733t} + 8600, \tag{14}$$

and the simulation results are shown in Figure 3. It can be observed that the measured thickness controlled by the initial controller had a $55nm$ difference compared with the target thickness, and after 10 iterations, the difference is less than $10nm$.

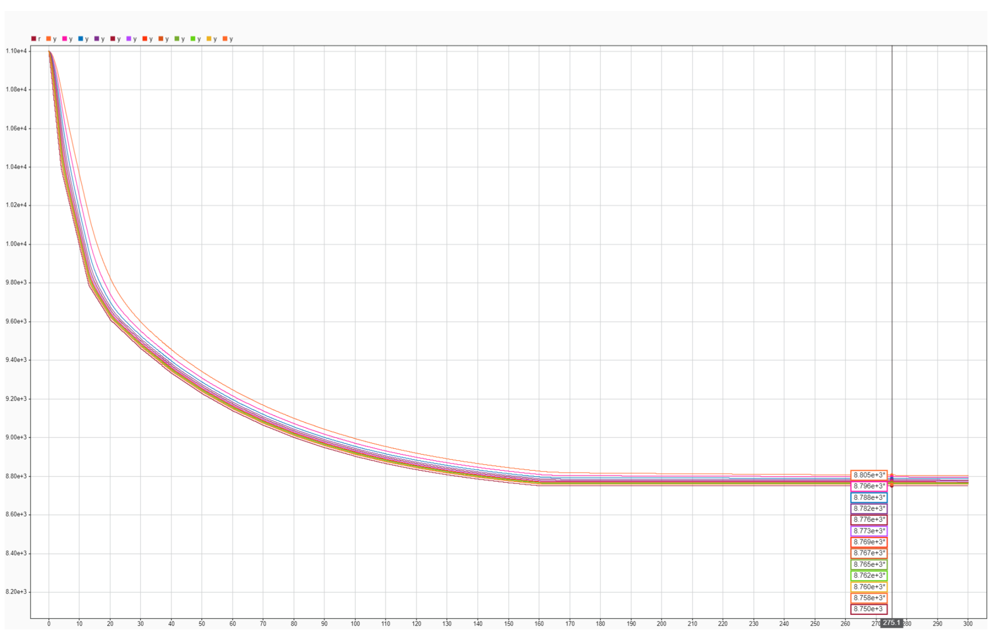

Figure 3: Simulation result when changing the initial thickness to be $11000nm$.

The controller gains are shown in Table 2.

Table 2: Controller gains in the simulation with initial thickness $= 11000nm$ and target thickness $= 8750nm$.

Gains	**Initial**	$1st$ **iteration**	$10th$ **iteration**
K_c	5e6	7.5e6	57.5e6
T_I	500	750	5745.7

3.1.2 Simulation with initial thickness $= 11000nm$ and target thickness $= 8500nm$

The reference signal is adjusted accordingly,

$$r = 2666.67e^{-0.01540t} + 8333.33, \tag{15}$$

and the simulation results are shown in Figure 4. It can be observed that the measured thickness controlled by the initial controller had a $67nm$ difference compared with the target thickness, and after 14 iterations, the difference is less than $10nm$.

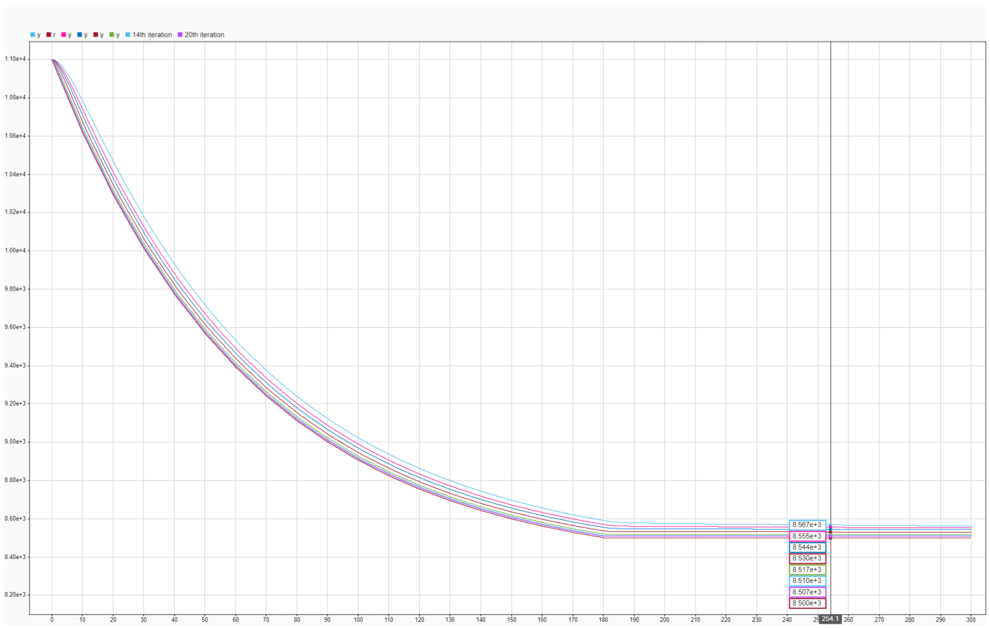

Figure 4: Simulation result when changing the initial thickness to be $11000nm$ and target thickness to be $8500nm$.

The controller gains are shown in Table 3.

Table 3: Controller gains in the simulation with initial thickness $= 11000nm$ and target thickness $= 8500nm$.

Gains	**Initial**	$1st$ **iteration**	$14th$ **iteration**
K_c	5e6	7.5e6	61.4e6
T_I	500	750	6144.7

3.2 Softbake plant change

When the sofebake plant changes, IFT should tune the controller parameters to control the new plant to track the reference photoresist thickness.

The new plant is chosen to be:

$$P = \frac{8.5e-7}{150s+2}, \tag{16}$$

and step size γ is changed to be 1. The reference signal is given by Eq. (15). The simulation results are shown in Figure 5. It can be observed that the measured thickness controlled by the initial controller had a $545nm$ difference compared with the target thickness, and after 47 iterations, the difference is less than $10nm$.

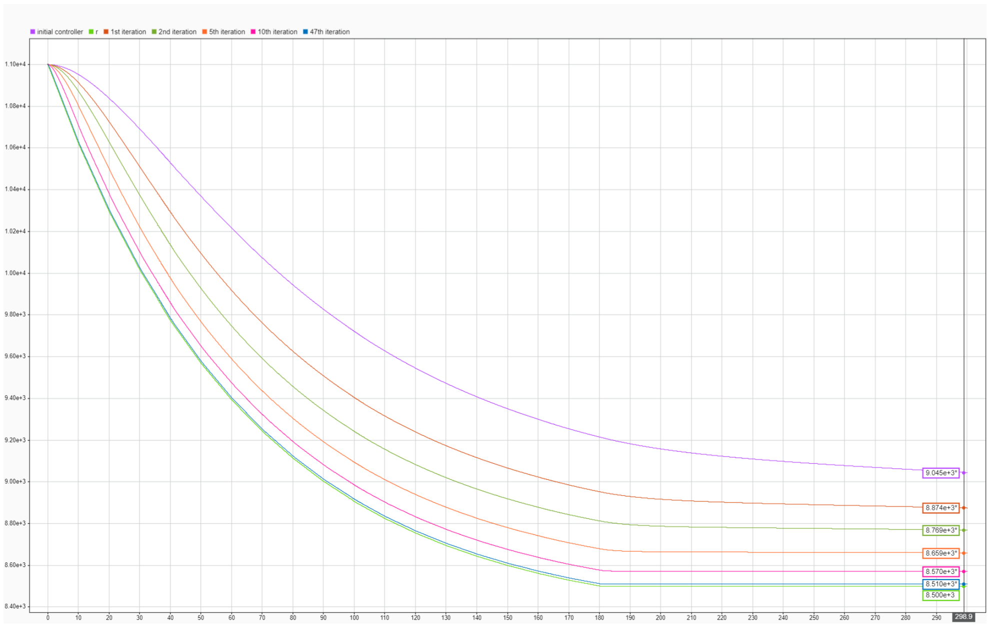

Figure 5: Simulation result when the plant is changed.

The controller gains are shown in Table 4.

Table 4: Controller gains in the simulation with changed plant.

Gains	**Initial**	$1st$ **iteration**	$47th$ **iteration**
K_c	5e6	10e6	582e6
T_I	500	1000	5.82e4

3.3 Summary

According to the various simulation results, it can be concluded that the implemented IFT method can auto-tune the PI controller parameters when the photoresist thickness changes and when the softbake plant changes. If the initial controller performs worse, it will take longer for the IFT to tune the parameters to satisfactory values. Besides, the control gains are too large for some of the simulation results, thus more constraints are to be considered in real applications.

References

[1] A. Tay, W. K. Ho, J. Deng, and B. K. Lok. Control of photoresist film thickness: Iterative feedback tuning approach. *Computers and Chemical Engineering*, 30:572–579, 2006.