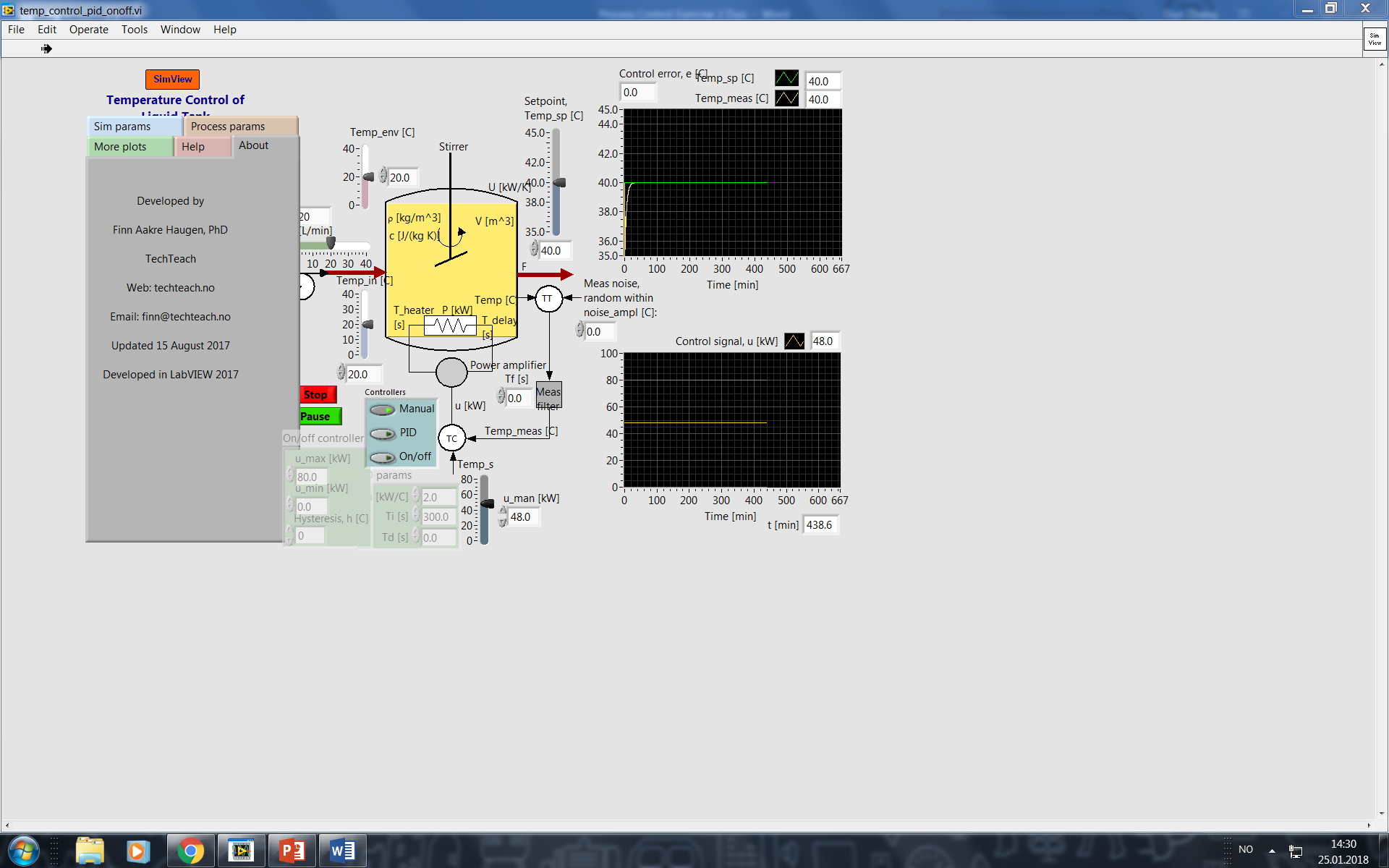
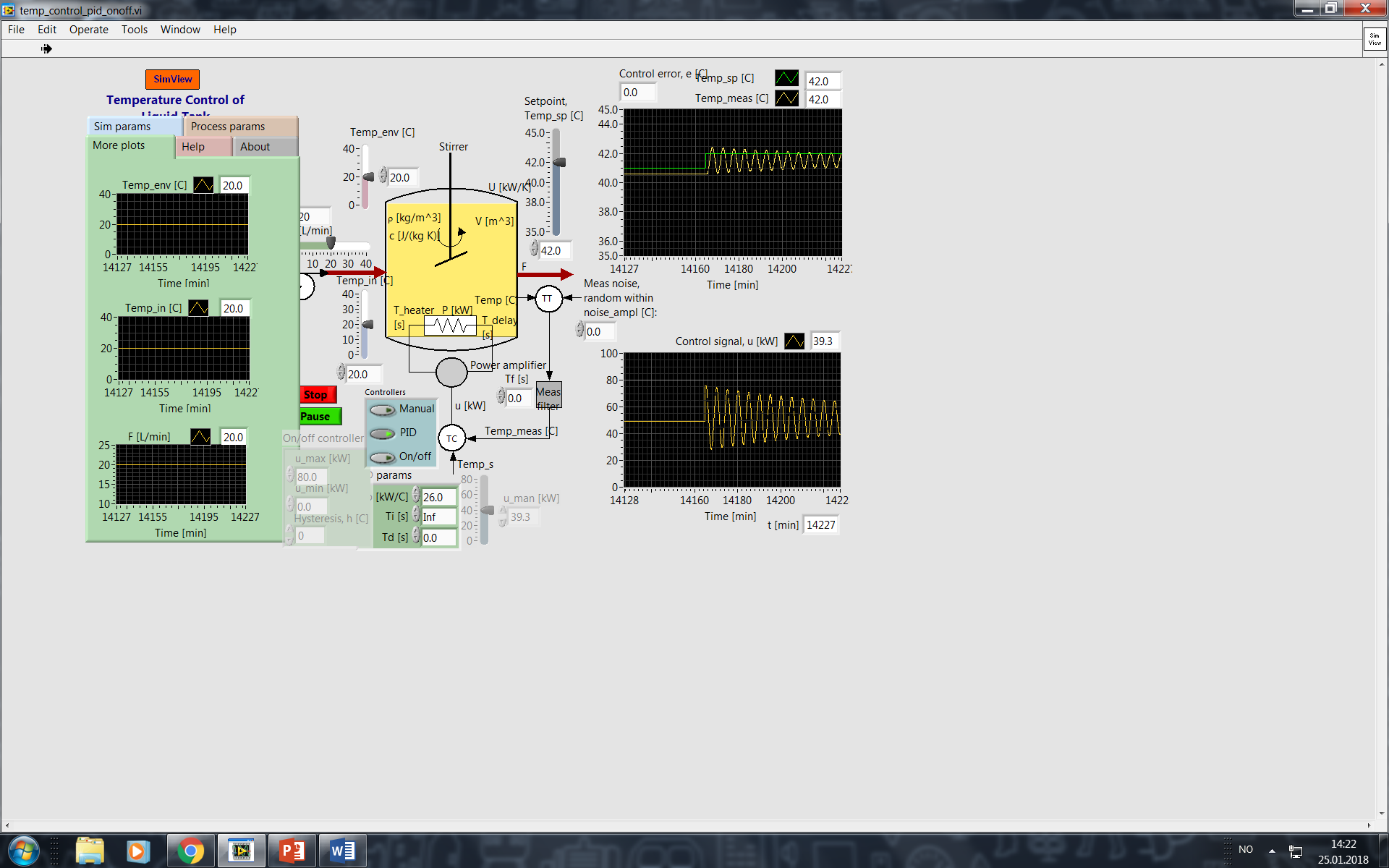
1. **Controller tuning**:
2. Tune the controller as a PI controller using the Ziegler-Nichols’ method. Check the stability with the setpoint step response and also with a disturbance step response (the ambient tank temperature can be regarded as disturbance).

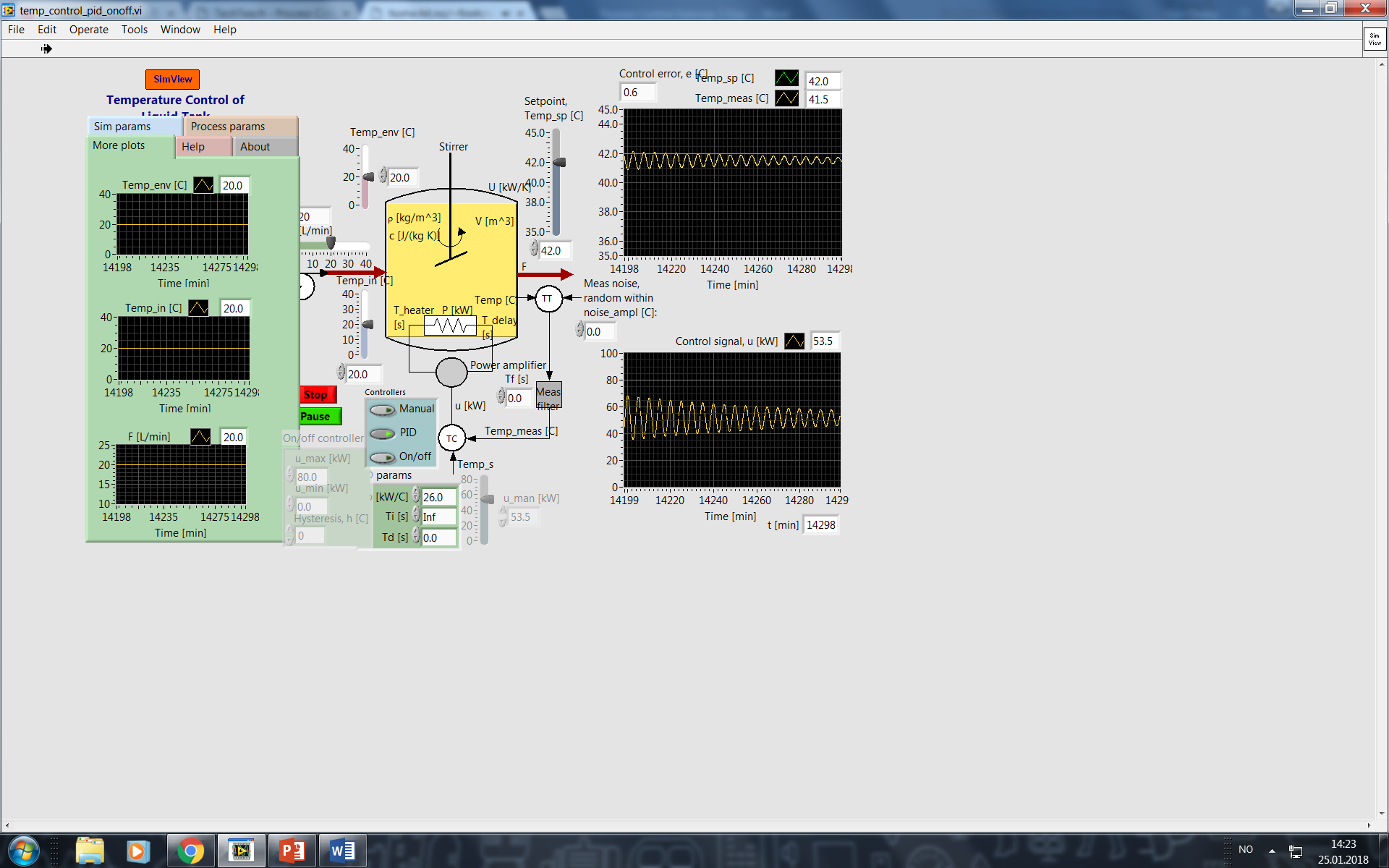
According to process of Ziegler-Nichols’ method:

With controller in manual mode: Bring the process to the specified operating point by manually adjusting the control variable until the process variable is approx. equal to setpoint.



Turn PID controller into P controller with gain Kp = 0. (Ti = very large. Td = 0.). Set the controller to automatic mode. Increase gain Kp (you may start with Kp = 1) until there are steady oscillations in the loop due to a small setpoint step.



  
This controller gain value is the ultimate gain, Kpu.

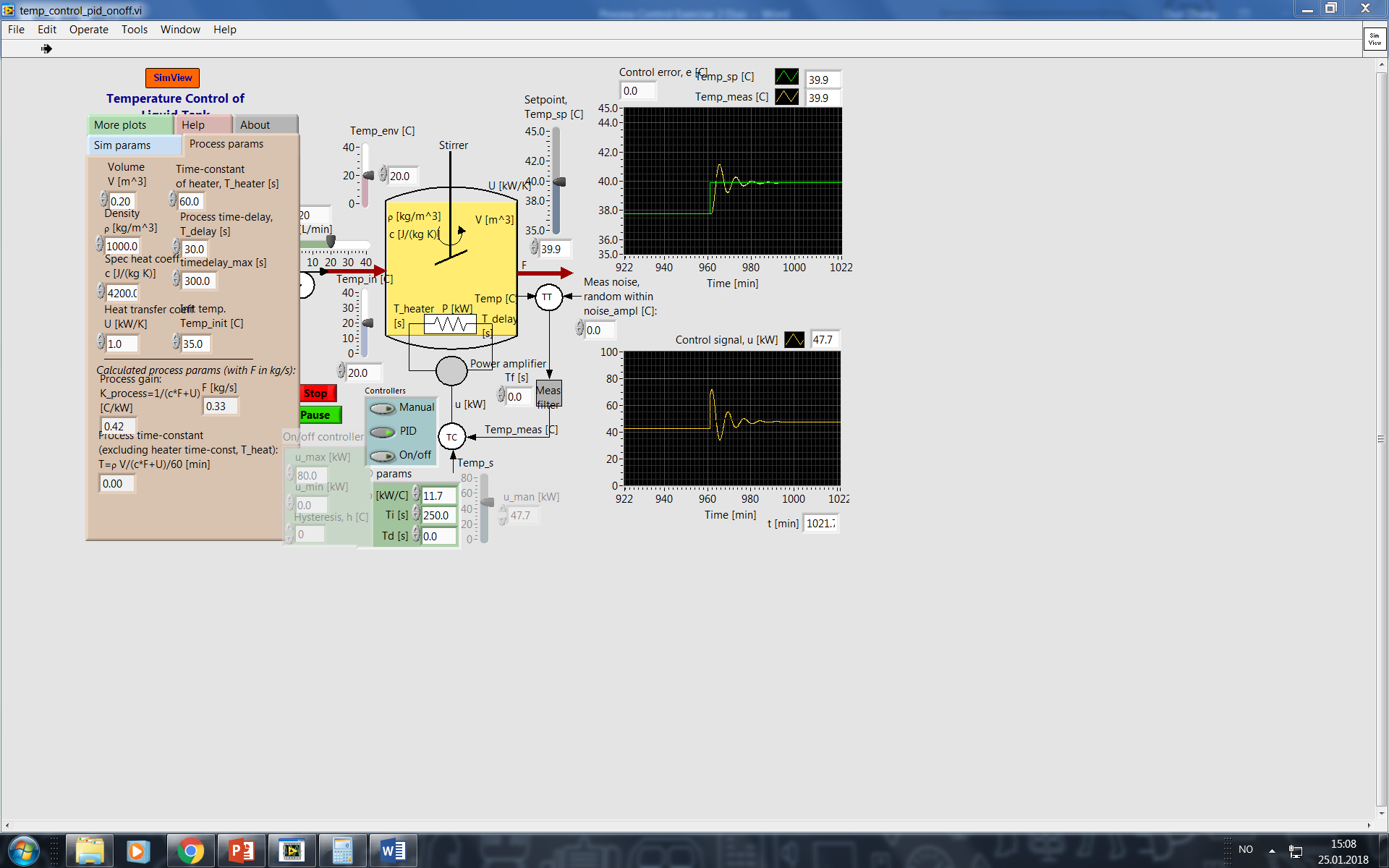
The Kpu in my case is 26

Read off the period, Pu, of the oscillations.

Pu is approximately 5 min, 300sec

Calculate controller parameters from the Z-N formulas:

According to formula, Kpu= 0.45\*Kpu=11.7, Ti=Pu/1.2=250

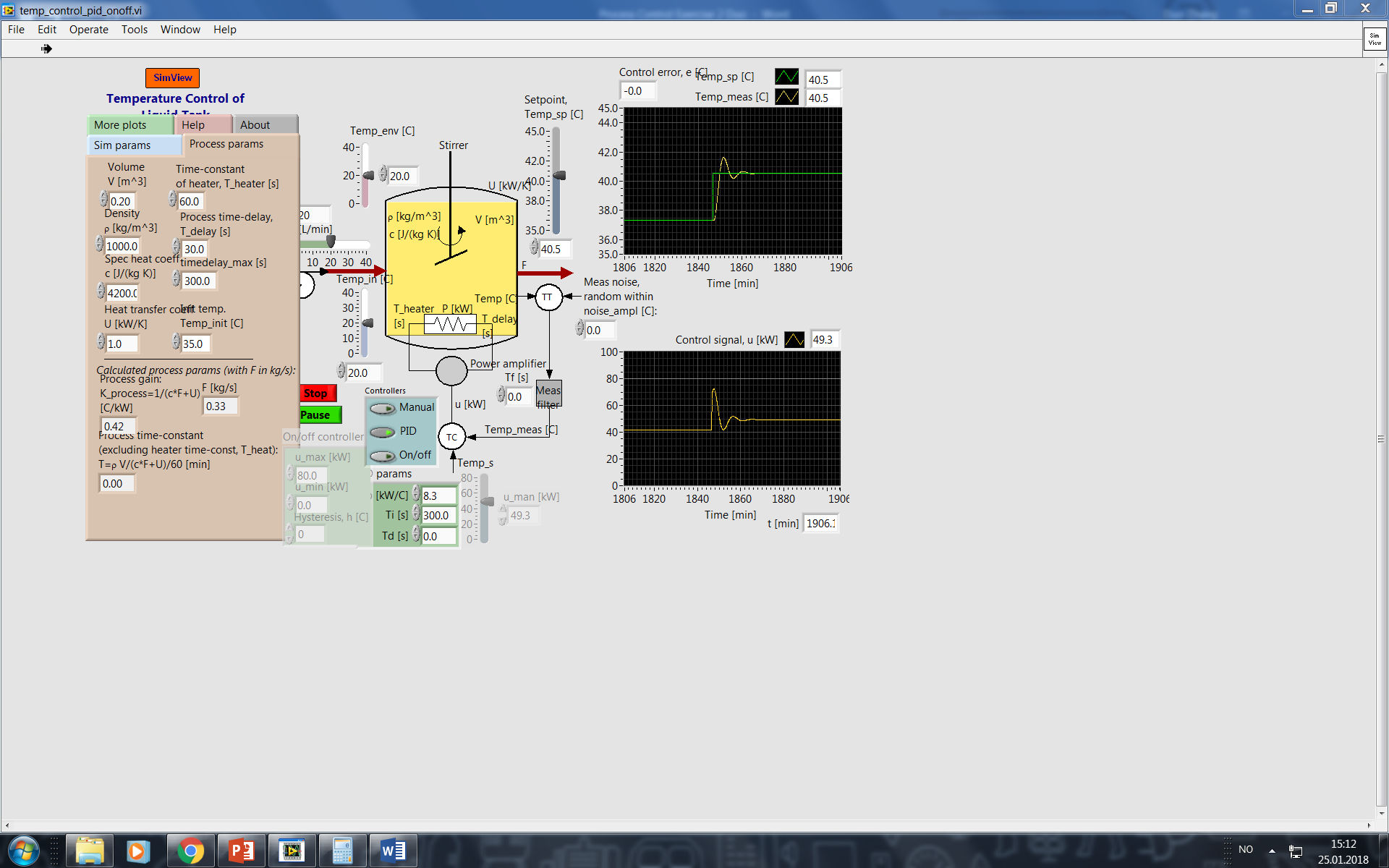


As we can see, the set up is generally fulfill the ¼ decay ratio.

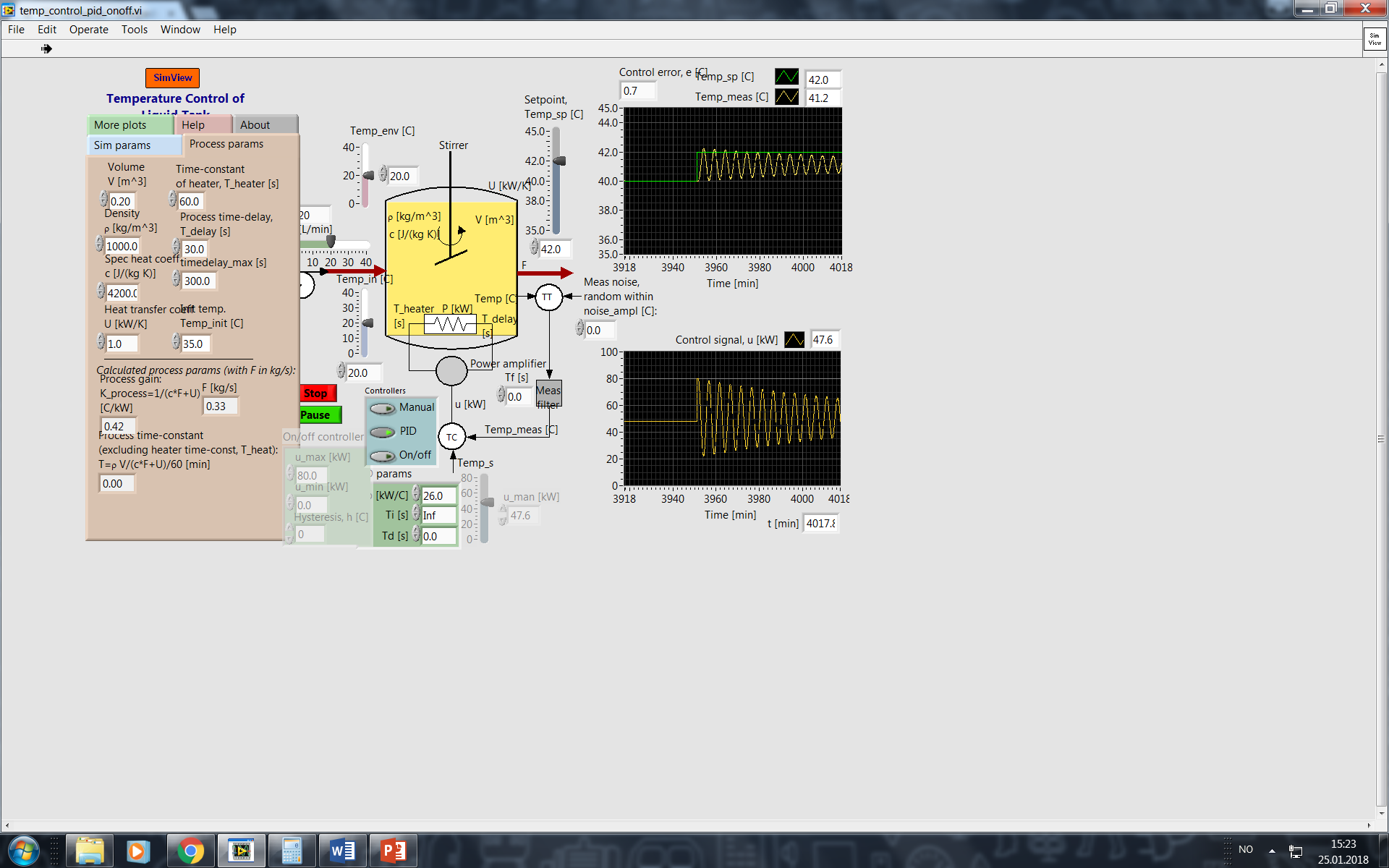
1. As in Problem 1a, but use the Relaxed Ziegler-Nichols’ method.

Use Relaxed Ziegler-Nichols’ method, Kpu= 0.32\*Kpu=8.3, Ti=Pu=300

The process converged to stable quicker than Ziegler-Nichols’ method

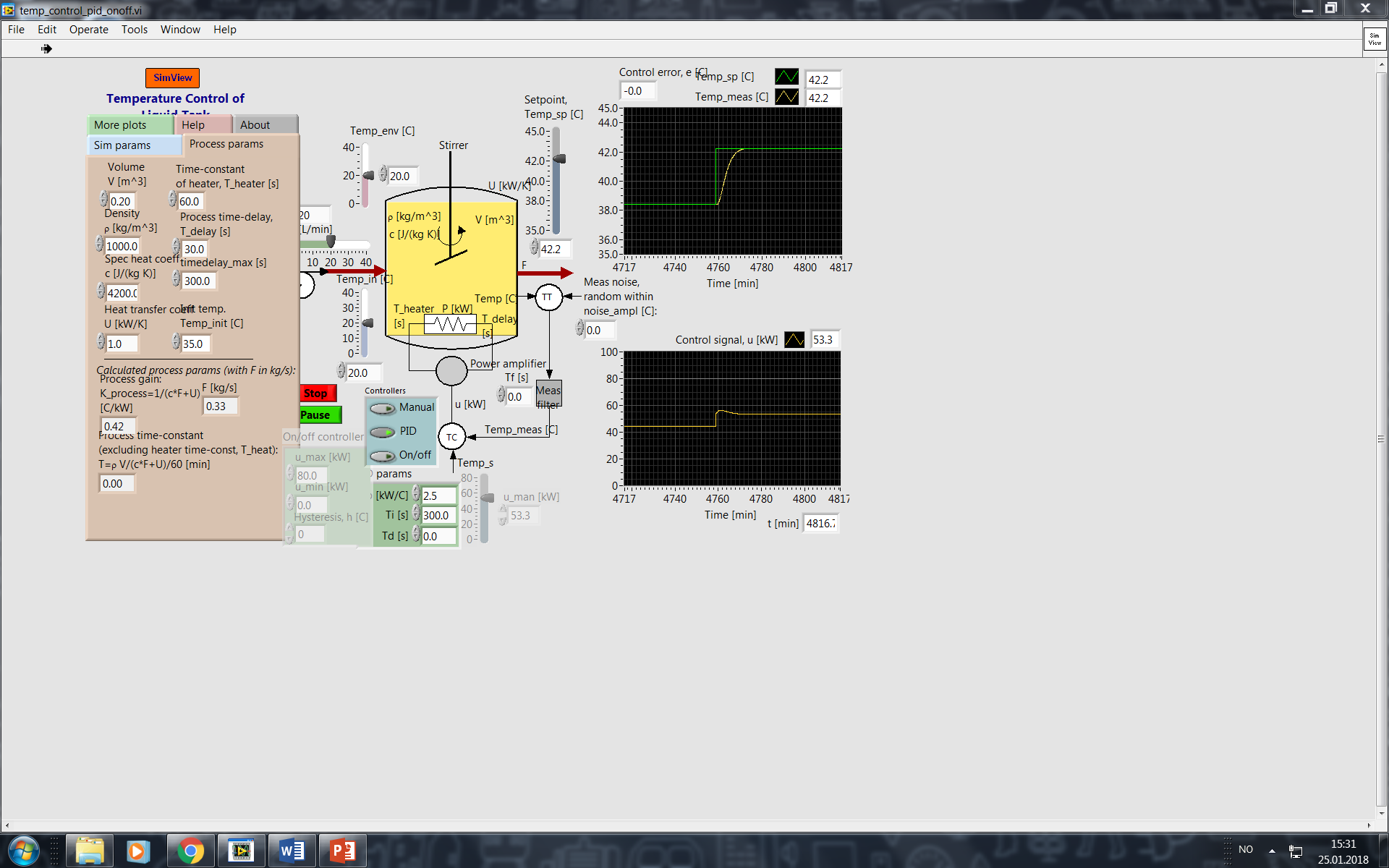


1. As in Problem 1a, but use the relay-method.



In the Relay (Åström–Hägglund) method, Kpu=4\*b/pi\*a, a is the amplitude of the process variable oscillation, and b is the amplitude of the control output change which caused it.

In this case, b is 2, a is around 1, so Kpu is around 2.5, Ti is equal to pu is 300



1. As in Problem 1a, but use the Skogestad method.

According to Skogestad method, for Time-constant + delay process:

Kp=T/(K\*(Tc+Tau))

Ti is min [T, c (TC + Tau)]

Skogestad suggests using TC = Tau, c = 1.5

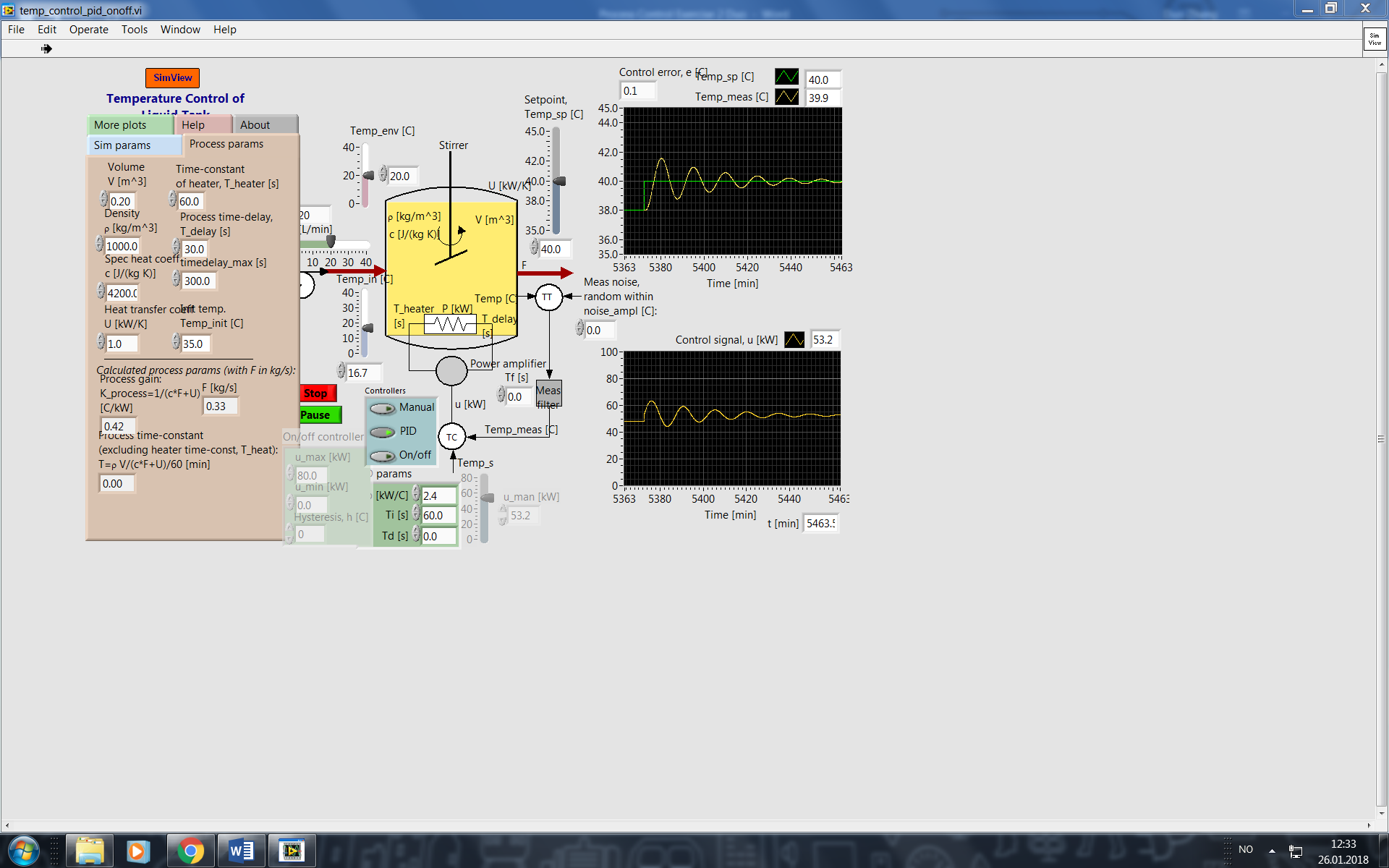
The process gain K=0.42, time constant T=60, time delay Tau=30

So we have:

Kp=2.4

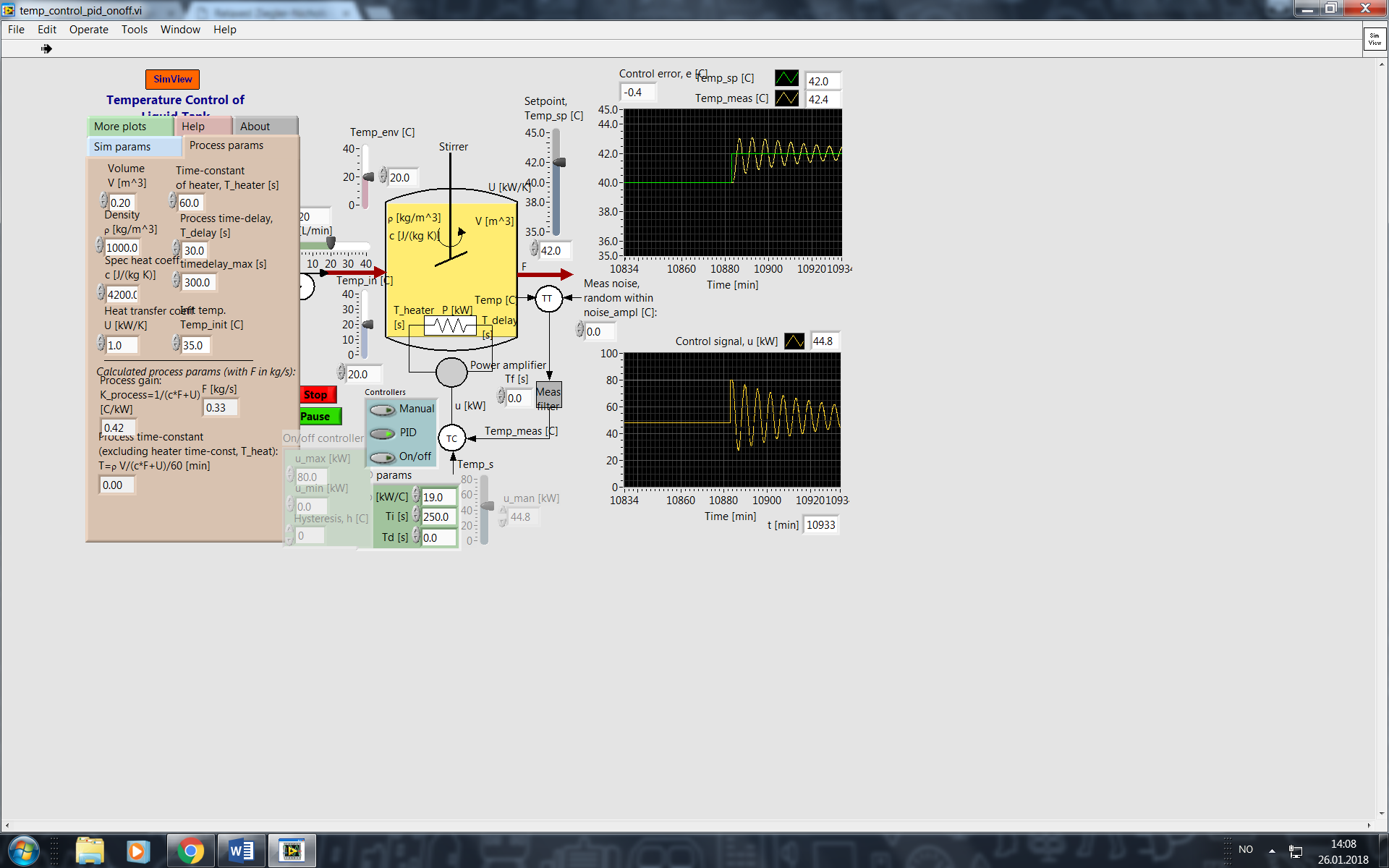
Ti=60

The response is:

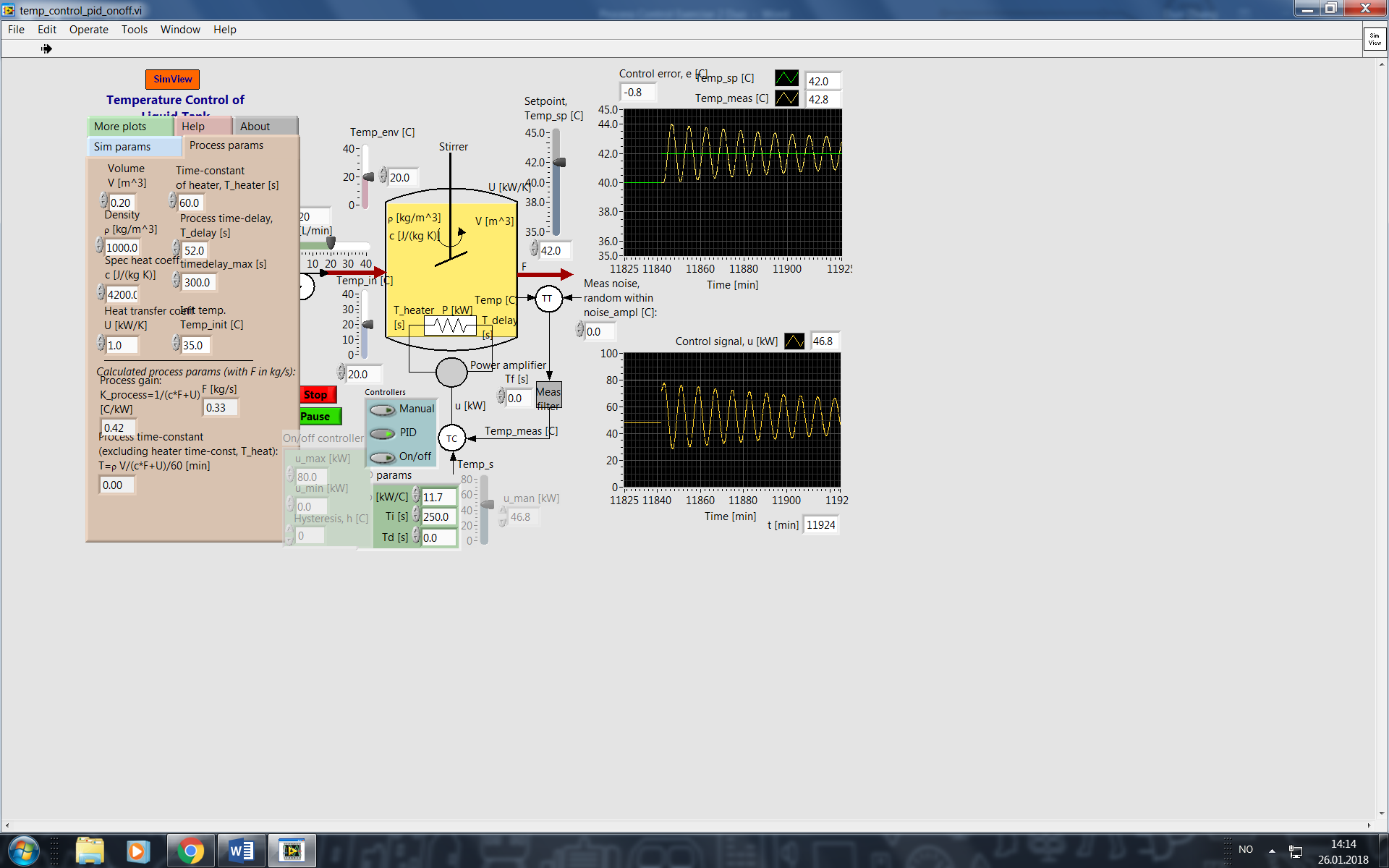


1. Find the gain margin GM and the phase margin PM of the control system for (1) Ziegler-Nichols’ tuning (cf. Problem 1a) and for (2) Skogetad tuning (1d). Are the values of GM and PM acceptable in each of the two cases?

For Ziegler-Nichols’ tuning, initial Ku is 11.7 and initial time delay is 30



Gain margin, value ∆Ku that brings the control system to sustained oscillations, is 19-11.7=7.3

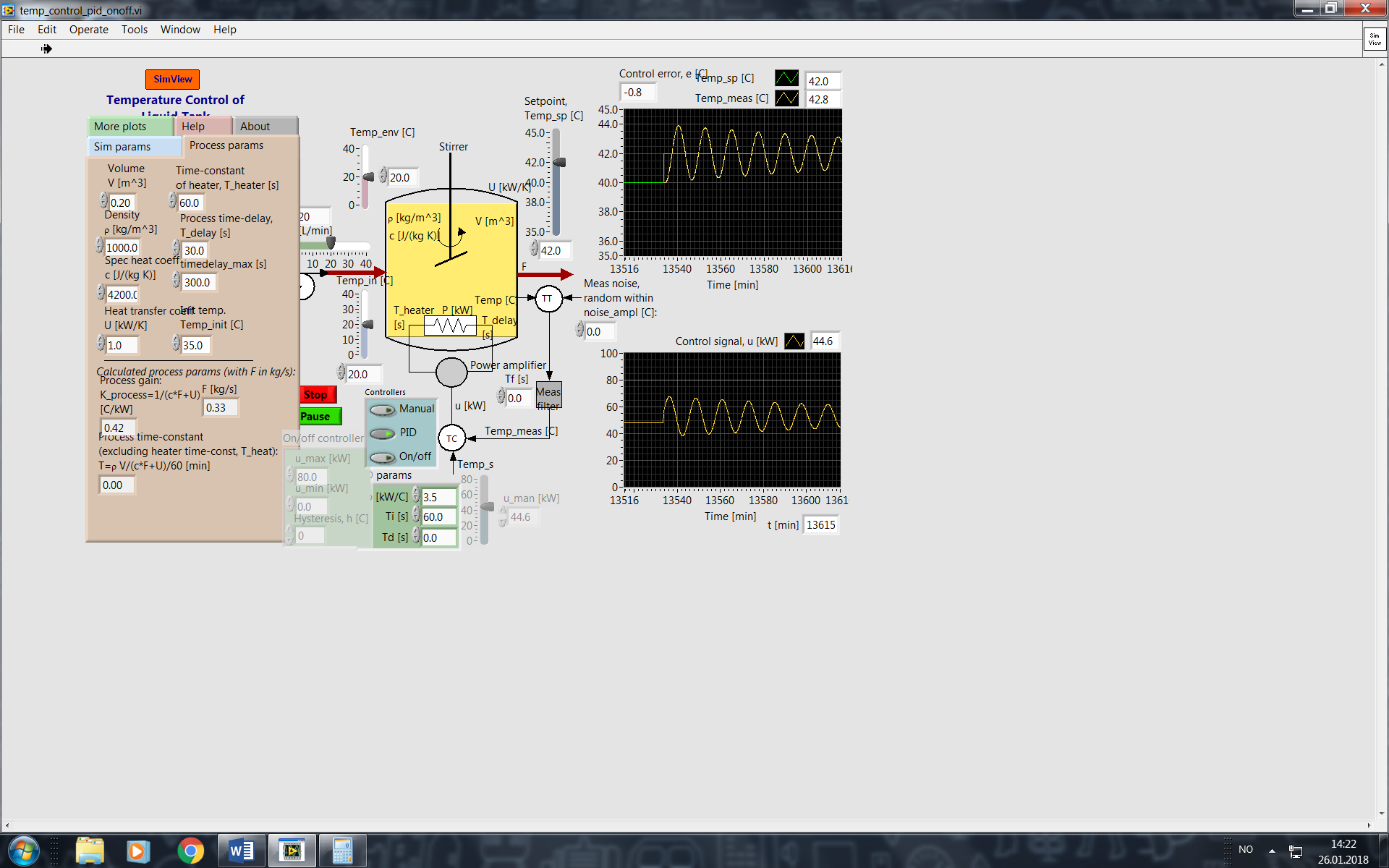


the value ∆τu that brings the control system to sustained oscillations, is 52-30=22

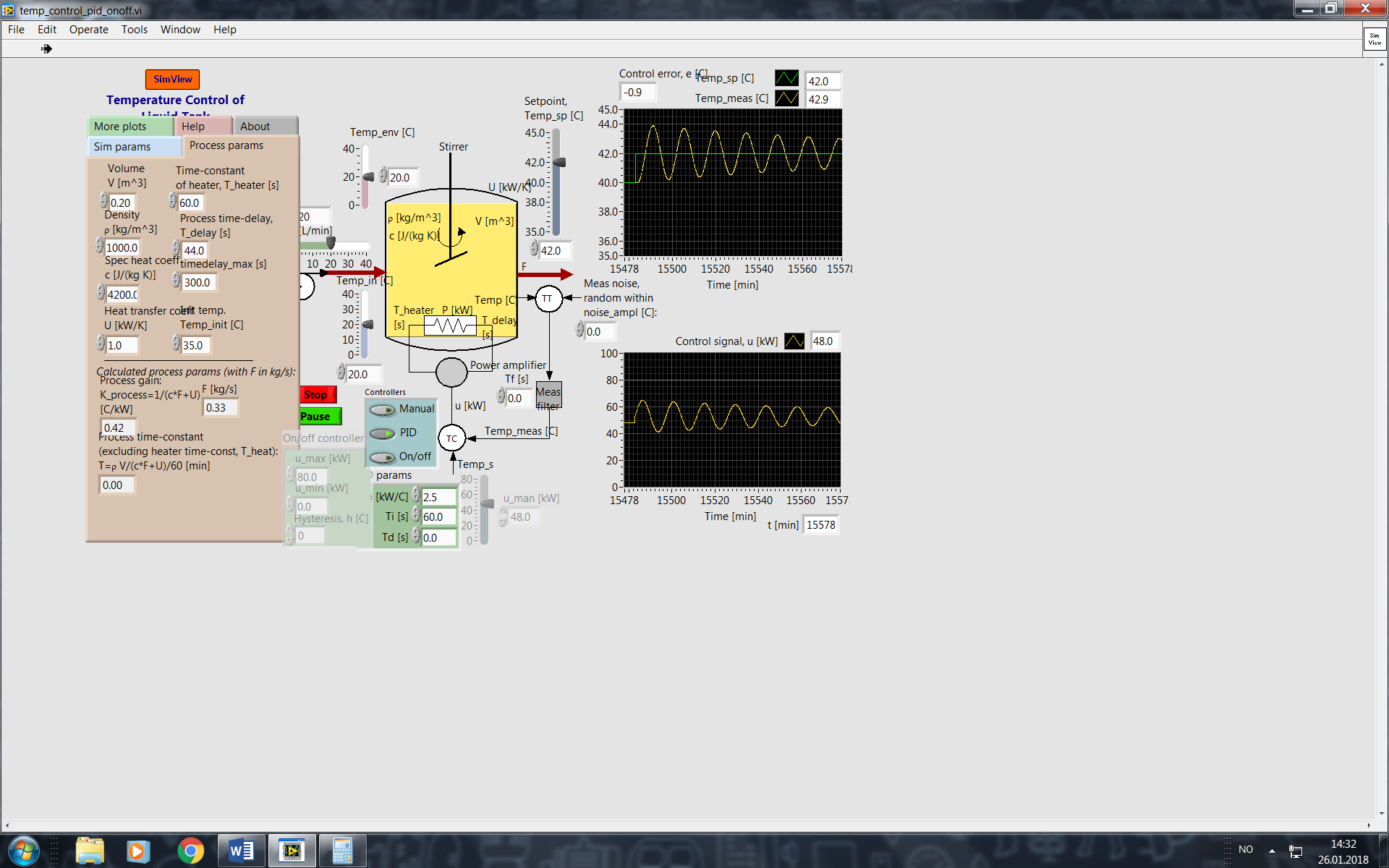
The period, Pu [s], of the oscillations is 8s.

The corresponding phase margin is PM [deg] = 360∆τu/Pu = 360\*22/8 = 990

For Skogetad tuning, initial Ku is 2.5 and initial time delay is also 30.



Gain margin is 3.5-2.5=1



∆τu is 44-30=14

Pu is 16

phase margin is PM [deg] = 360∆τu/Pu = 360\*14/16 = 315

Proper values of the stability margins:

1.7 = 4.6 dB ≤ GM ≤ 4.0 = 12.0 dB

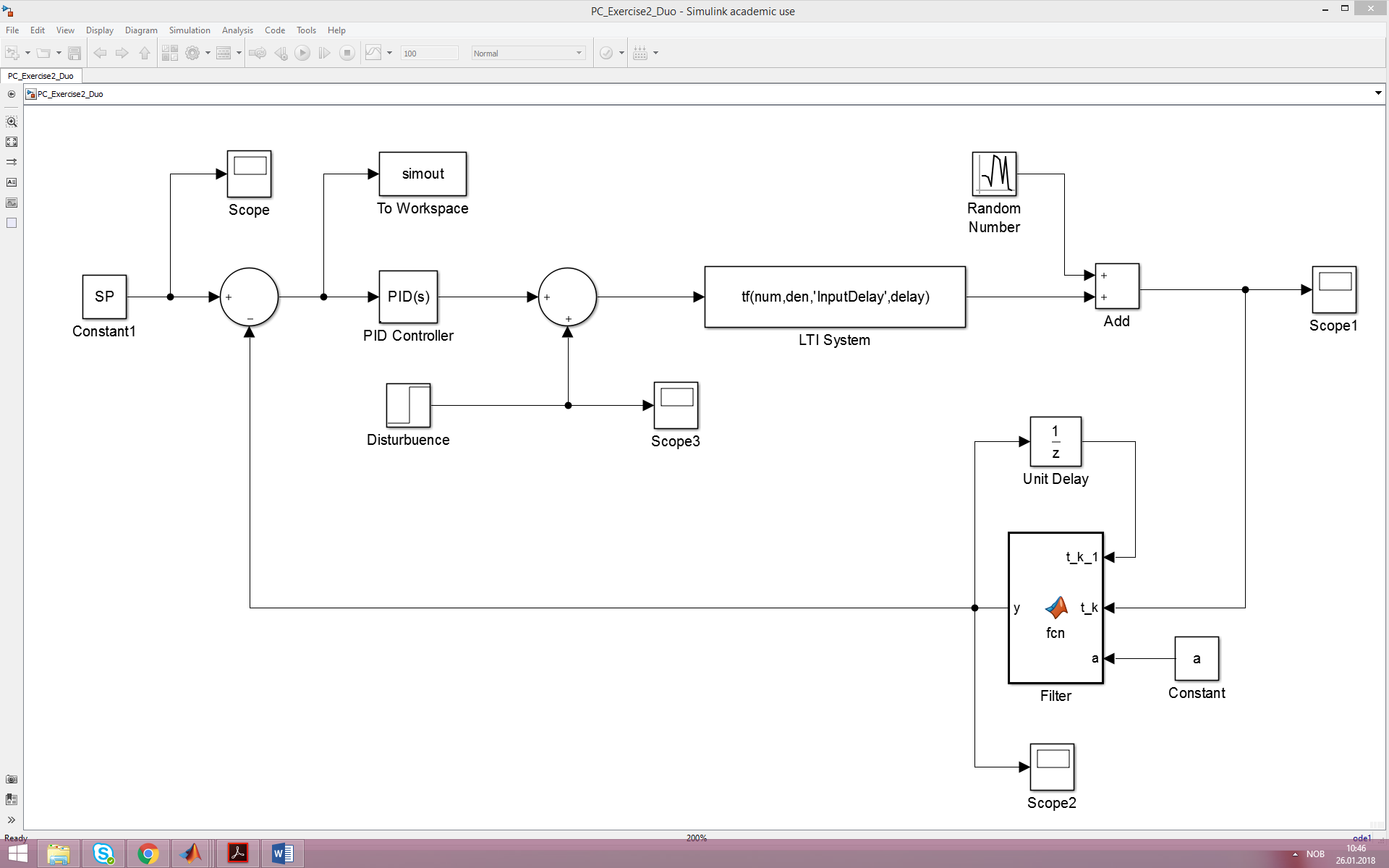
and

30 ≤ PM ≤ 45

Seems both method didn’t fit the proper range of gain margin and phase margin.

1. **Implementation of a simulator of a control system in Simulink**:
2. Implement in Simulink a simulator of the control system system containing the components described below:
   * The process is a time-constant with time-delay system (i.e., time-constant system in series with a time-delay). The process gain is 5, the process time-constant is 10 s, and the time-delay is 2 s.
   * An “input process disturbance” in the form of a step signal that acts on the process input (i.e. the disturbance is added to the control signal; most process disturbances are actually input disturbances).
   * A measurement filter in the form of a time-constant filter with time-constant 1 s. Although you may implement this filter using a transfer function block in Simulink, you shall here implement the discrete-time filter algorithm in a MATLAB Function block.
   * A PID controller. (You can use an inbuilt PID controller block in Simulink.)
   * Calculation of the IAE index.
   * Use a fixed step solver with time step of 0.01 s.

The Simulink:



Matlab code:

%Parameters

SP=2; %set point

a=0.1; %a for the measurement filter

num=5; %process gain

den=[10 1]; %time constant

delay=2; %time delay

t\_step\_D=10; %time step of disturbance

D\_0=0;

D\_1=0.5;

%simulator

t\_stop=100;

T\_s=t\_stop/10000;

options=simset('solver','ode1','fixedstep',T\_s);

%starting simulation

sim('PC\_Exercise2\_Duo',t\_stop,options)

IAE is calculated by the ‘simout’ from ‘To Workspace’ of Simulink:

IAE=sum(abs(simout))\*0.01, 0.01 is the time step.

1. Tune the PI controller with the Ziegler-Nichols’ method. Then, apply a unit step in the process disturbance. What is the IAE index for time-interval from zero until the response has virtually become constant?

To tune PI and PID controller, the mean and variance of random number is set to 0, to avoid the influence of noise.

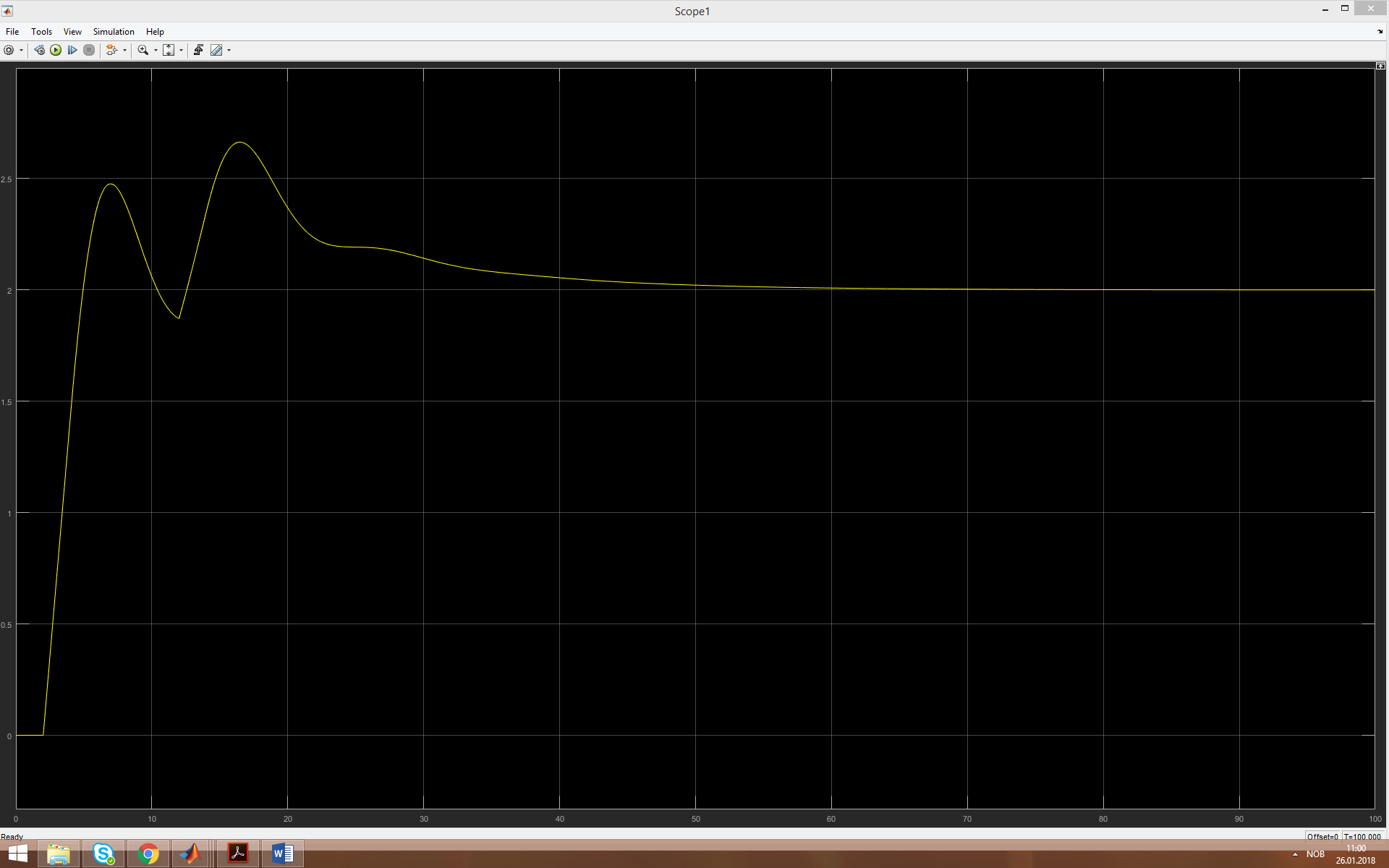
When set the Kpu=1.6, the pu is around 0.08s



Kpu = 0.45\*1.6 = 0.72

Ti=pu/1.2=0.067

We can see that the process is quite stable.

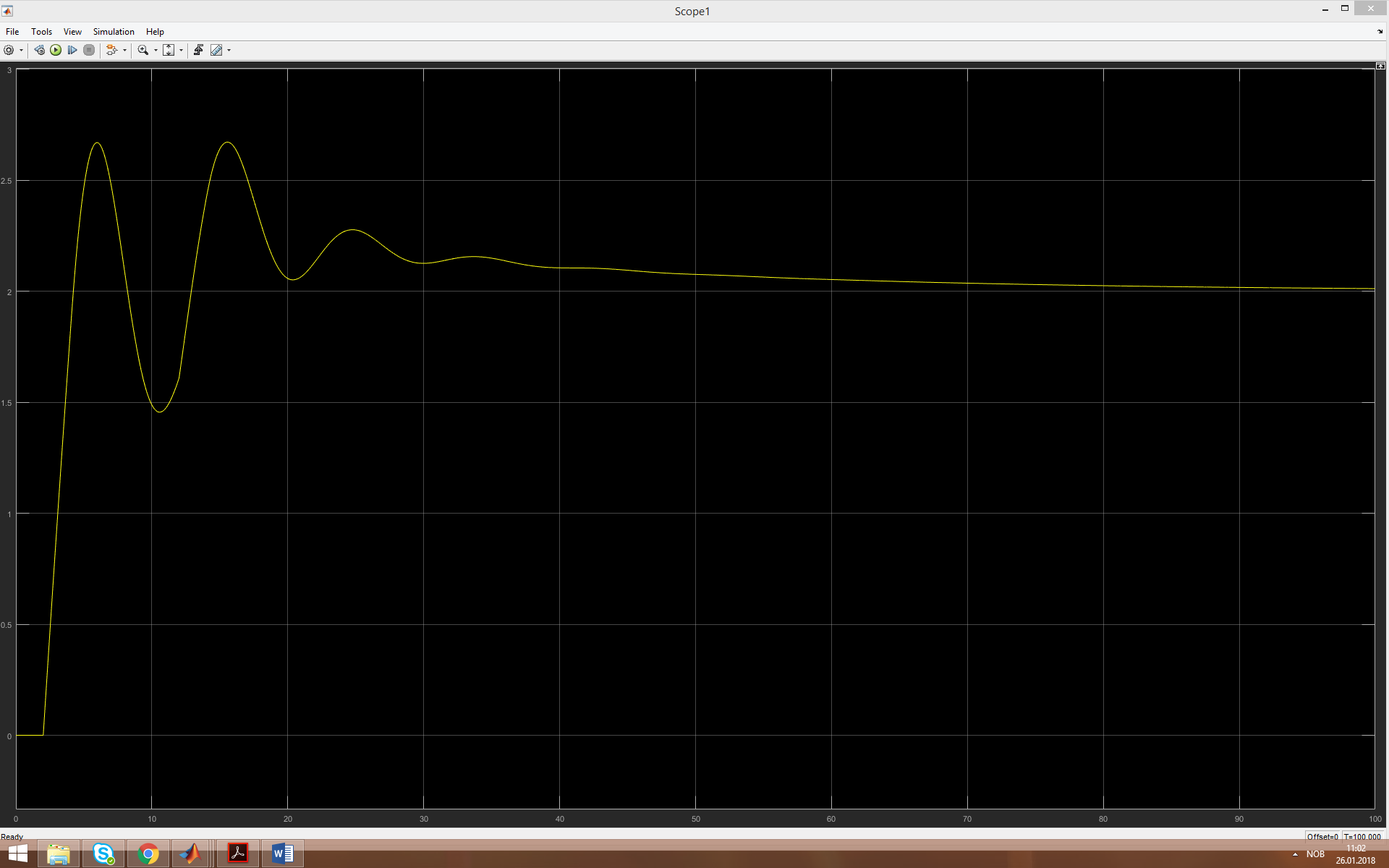


IAE=15.8469

1. As Problem 2b, but now a PID controller.

According to formula, Kpu = 0.6\*1.6 = 0.96, Ti=0.5\*pu=0.04, Td=0.125\*pu=0.01

The response is:



IAE=18.3500

1. Probably, you will see here that the IAE index with the PID controller is less than with the PI controller, indicating that PID is favourable. Despite this, what is the reason why PI often often is preferred to PID in a practical system? Illustrate with a simulation!

It seems stranger that in my simulation, the PI workd better than PID.

The PI controller seems have better robust than PID against noise, I tested the IAE of PI and PID under different measurement noise, the results are:

For PID controller:

Noise 0.01, IAE 18.4426

Nosie 0.05, IAE 19.1909

Noise 0.1, IAE 20.0335

For PI

Noise 0.01, IAE 16.5369

Noise 0.05, IAE 17.8033

Noise 0.1, IAE 18.8816