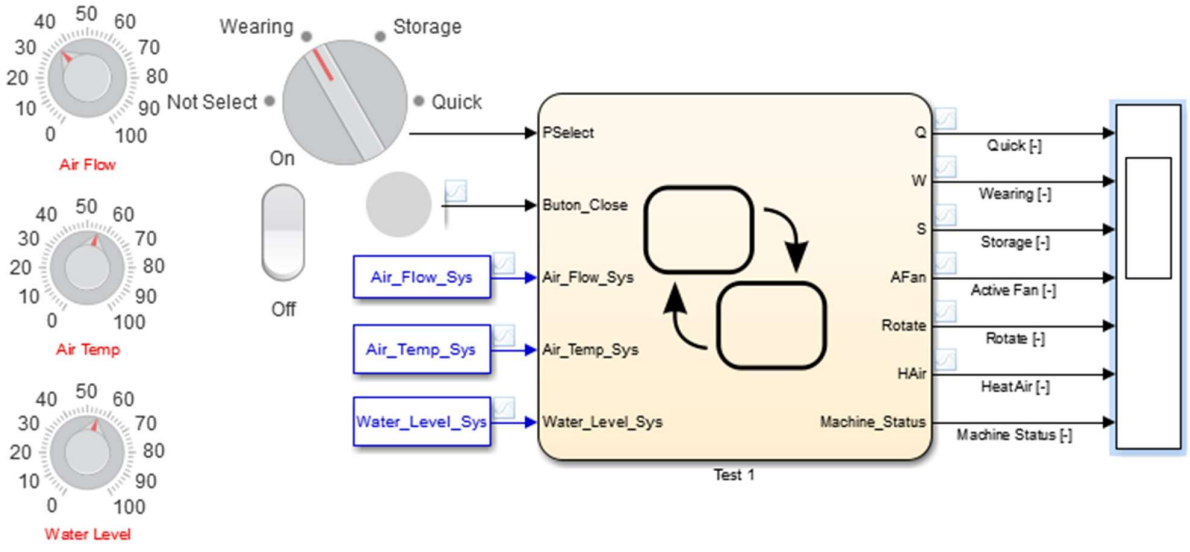
DRYING MACHINE

Embedded System Design and Modeling

Student: Burduhosu Ana



JANUARY 18, 2021

Inputs:

- Program Select = PSelect;
- Cancel Button = Cancel;
- WaterLevel = Water_Level_Sys;
- AirFlow = Air_Flow_Sys;
- AirTemperature = Air_Temp_Sys;
- Time – use block ‘Clock’.

Outputs:

- ActivateFan = Afan;
- Rotate;
- HeatAir = Hair;
- Machine Status = Machine_Status;
- Time_W – time when the program is in working state;
- Time_Fault – time when the program is in fault state.

I start from the IDLE state, this state is like an initialization (starting point). If the condition "[(PSelect == 1 || PSelect == 2 || PSelect == 3) && Water_Level_Sys> = WLevel_Low]" is true, I go to WORKING. From WORKING to NOT_WORKING, I go when I detect an error, otherwise I go to the FINISH state after the program ends (2 hours or 1.5h, depending on the selected program) and stay in IDLE mode until I set a new PSelect value.

This machine has 3 programs: Wearing, Storage and Quick with different time programming. So, we have 3 possibilities for the **PSelect** button/input, 4 with no select program. For the state Working, I have another 3 states for the programs and another for Rotate state.

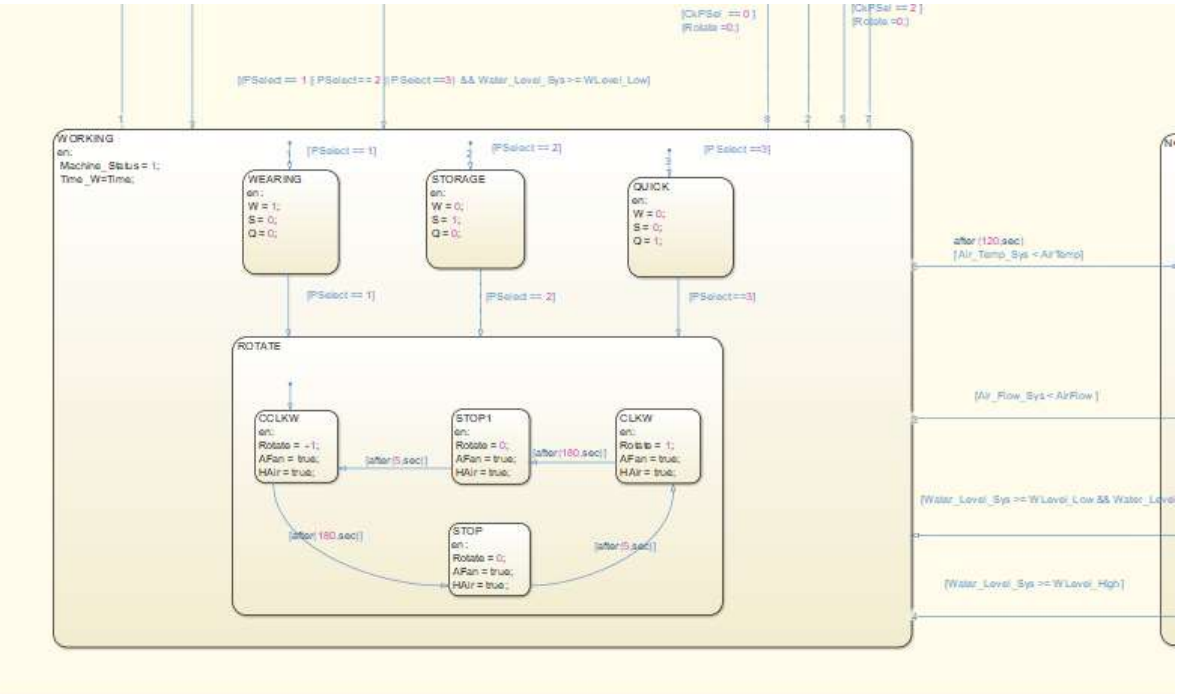


Fig. State Working

e.g. [PSelect == 1], this condition for Wearing state.

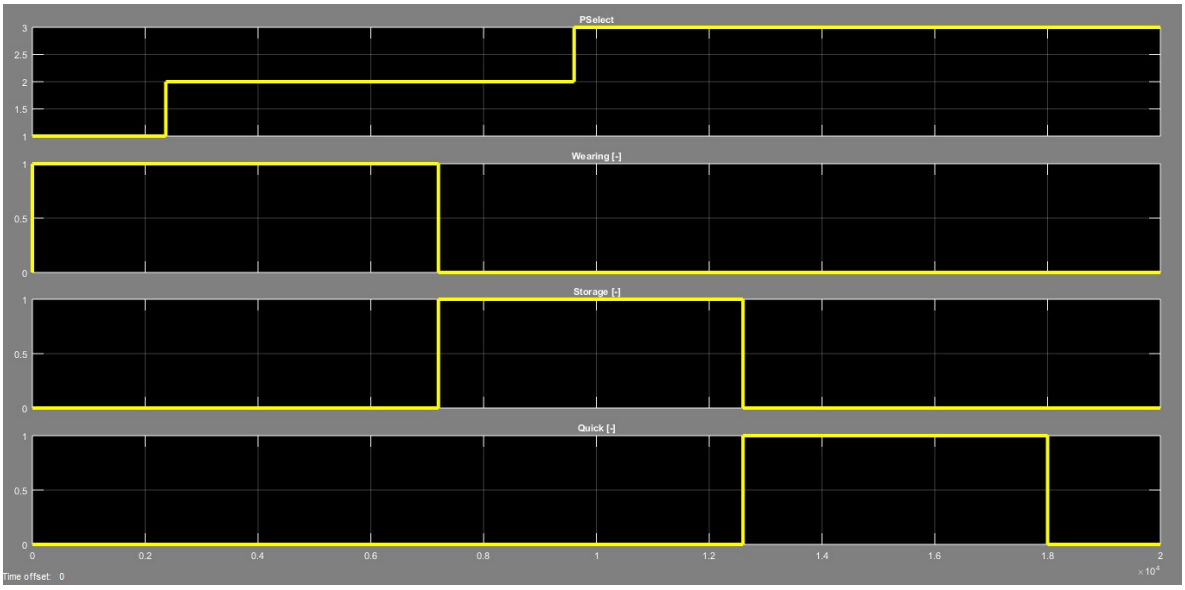


Fig. Example no. 1

Rotate state make the drum of the machine to rotate every 3 minutes, stop for 5 sec and change the rotation direction. To implement this, I use "*after*" and four states who have next variables:

- Rotate, -1 for counterclockwise, 0 for stop and 1 for clockwise.
- AFan and HAir, which are always true when the program works.

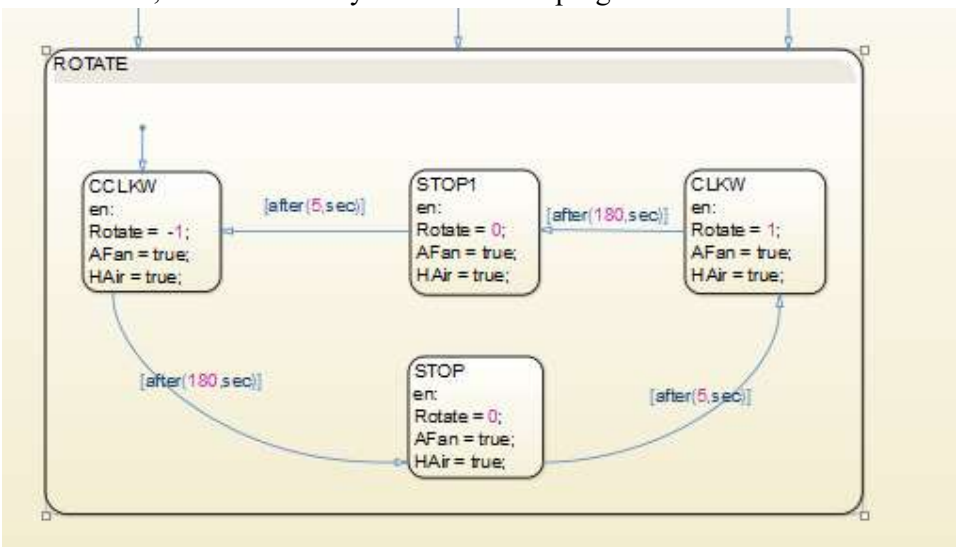


Fig. Rotate state

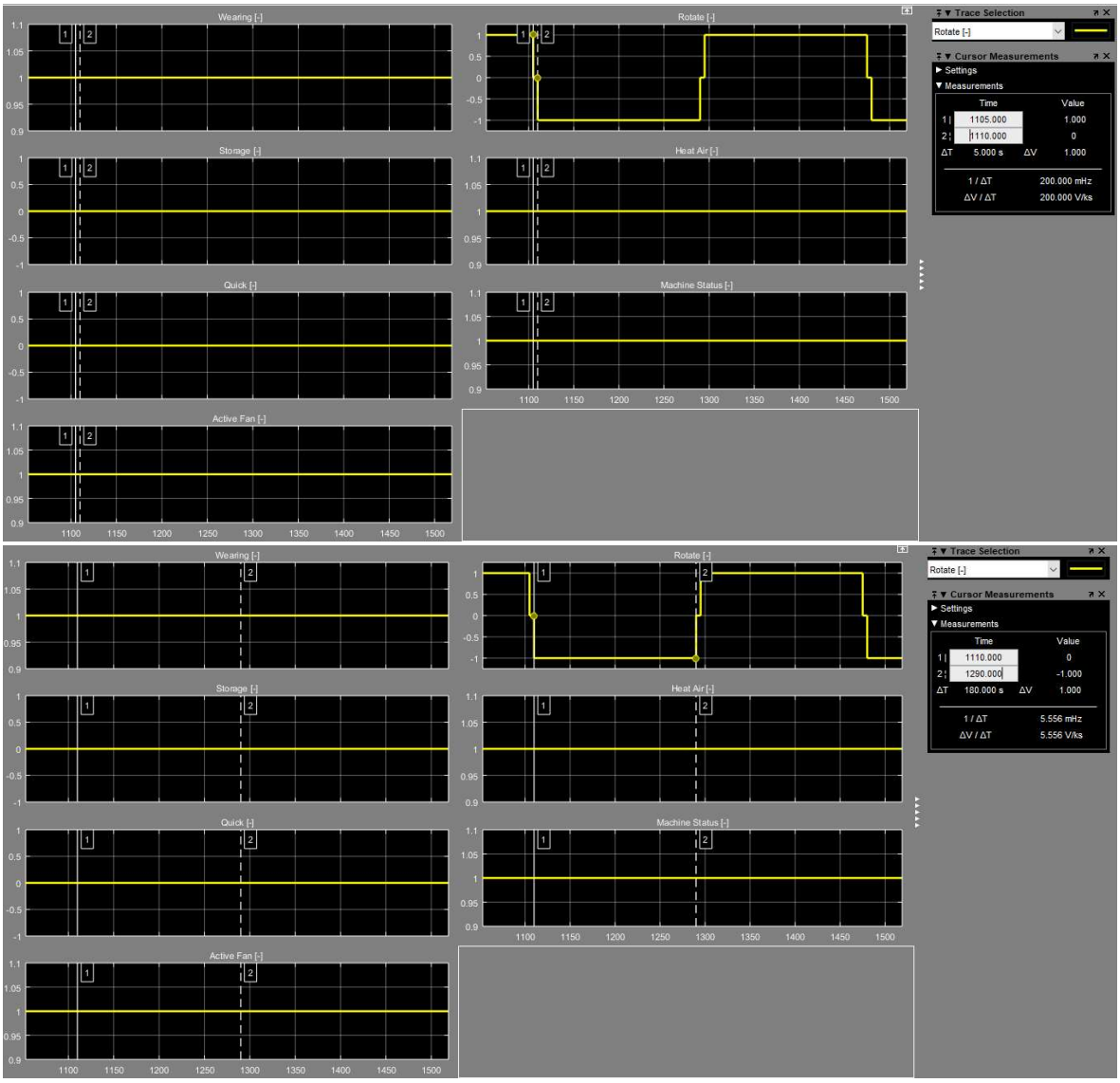
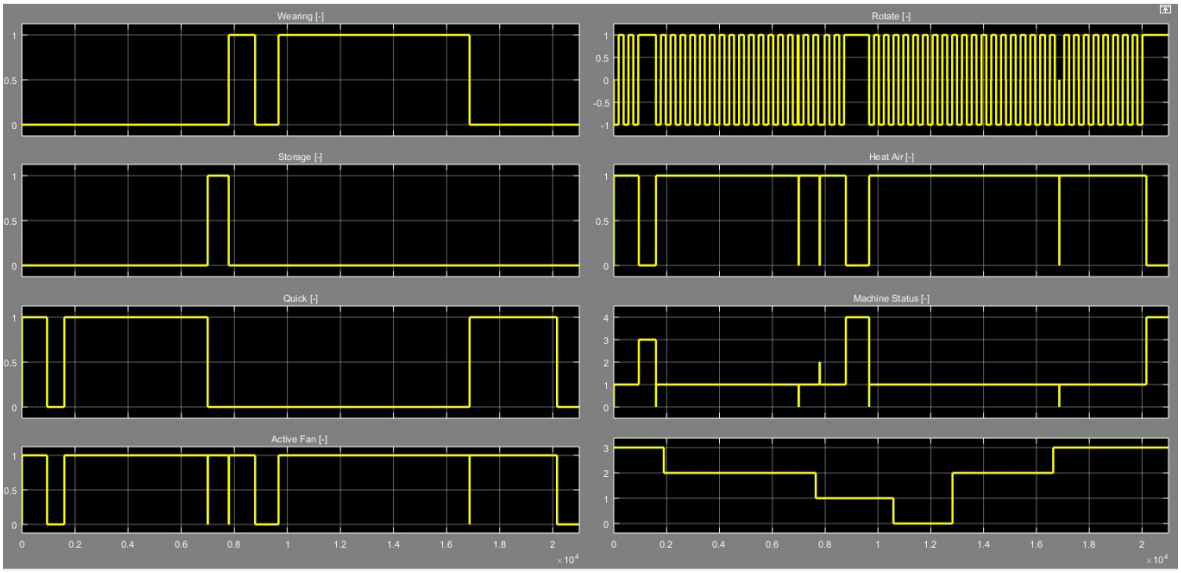


Fig. Example no.2

I set Wearing, after 2 hours the program is finish. If I don't select another program the output is always 0. To happen this, I create a new state Finish and a local variable 'CkPSel', to check the program select and do not start over again same program if time of this program pass. Also, if I select another program while running another, the first one will end first and then the other.

For the Fault Detection, are 3 error: Water Full, Filter Full and Heater Fault. I do three different state for each and I put the conditions for fault on transition.



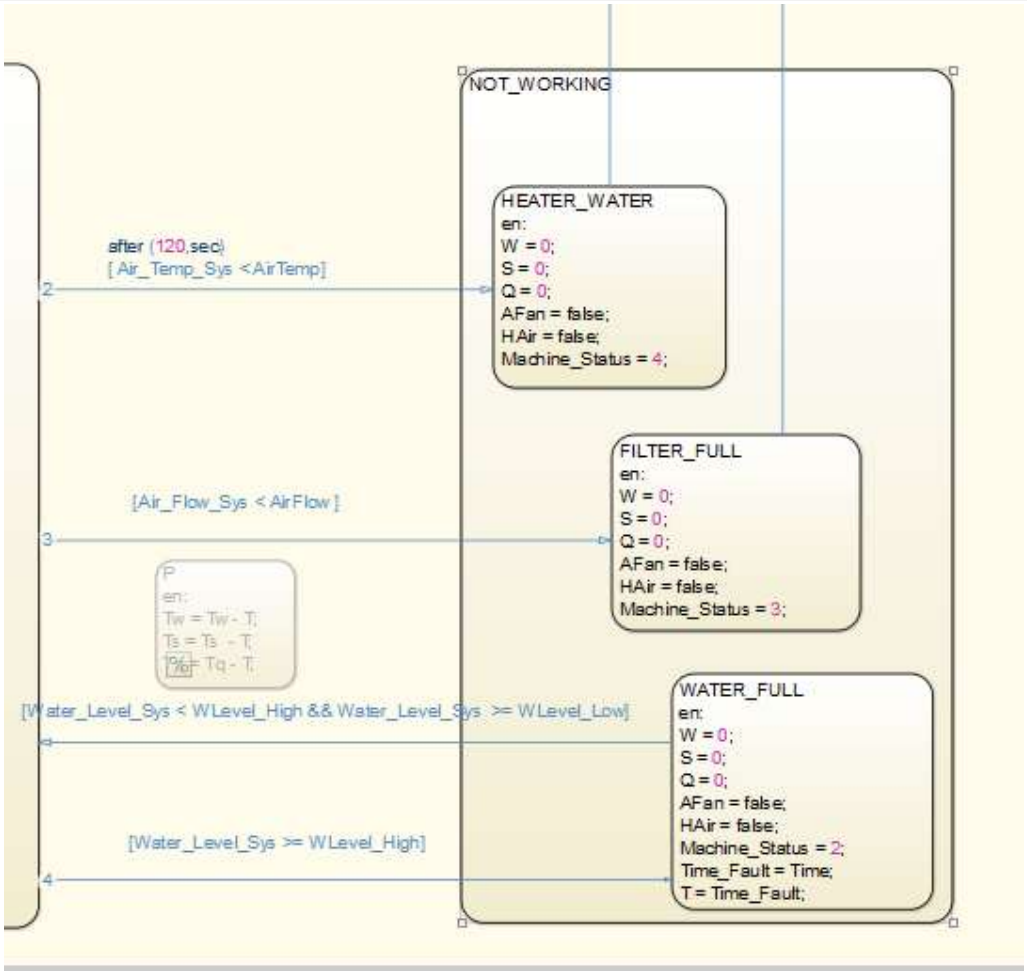


Fig. State Not Working – Fault Detection

To be easier and if the values of the parameters are change some time, I do a script, .m file in Matlab and I initialize variables. So, the Air_Temp_Sys, Air_Flow_Sys, Water_Level_Sys

are the values from system, values chose by me (inputs). The W_Level_High, AirFlow and AirTemp are the values when the program failed or is out of range.

e.g. after(120, sec)[Air_Temp_Sys < AirTemp]

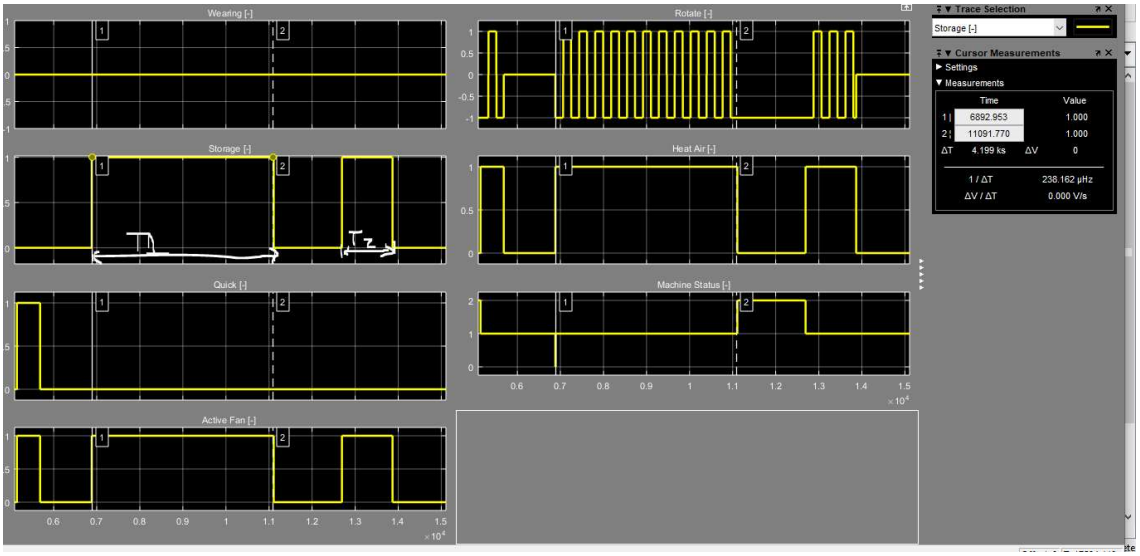
For the input Cancel, I do another state and with two condition [Buton_Cancel == 0] for off and [Buton_Cancel == 1] for on.

For req. Water Level reaches 90, stop and set status to WATER FULL:

- do not start until Water Level is below 10;
- afterwards, continue from when the program was paused.

I put a condition at the start, Water_Level_Sys >= WLevel_Low and for continue after program was paused, I try to find the time for the WORKING State and time for the NOT_WORKING state.

I put a Clock block; this is the input *Time*. Then I make local variables for start time, *Tstart*, fault time, *T* and time for state Working, *Time_Work*.



T1 + T2 =~ 5400 – Time for Storage Program

Tstart = Time; this is time when I go in IDLE state, is important when I finish for example program 2, I have a new Tstart, after that I go to WORKING state and add Ts (time for storage program) and I decrease T (time when I go in fault mode – Water Level).

Ts= (Tstart + Ts) - T;

Test case 1 – Normal Condition:

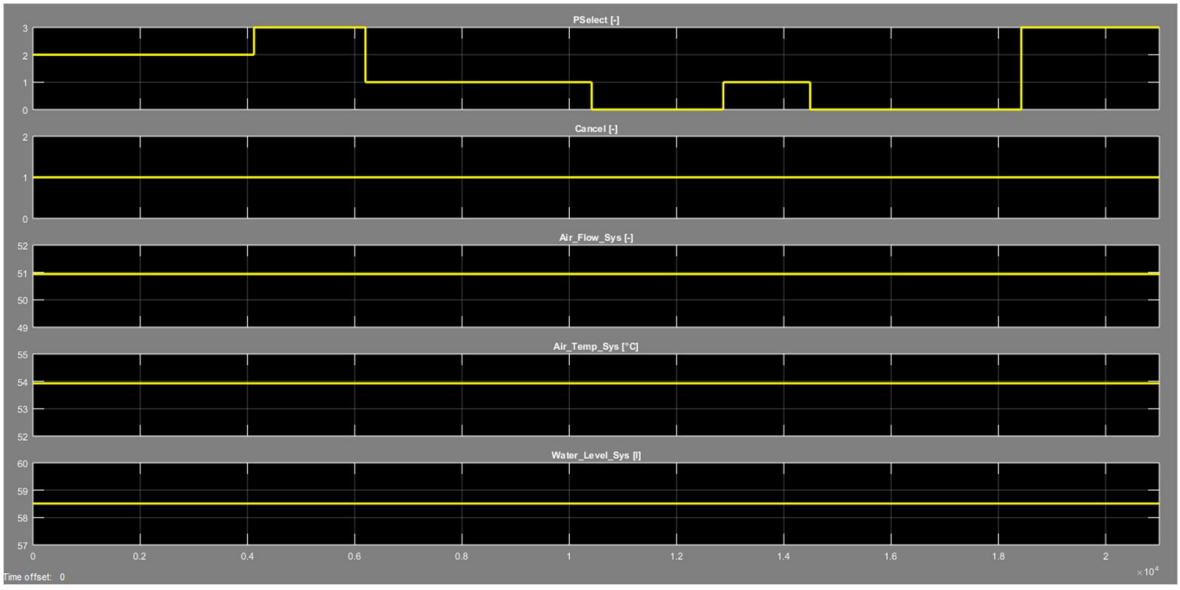


Fig. Inputs of Drying Machine

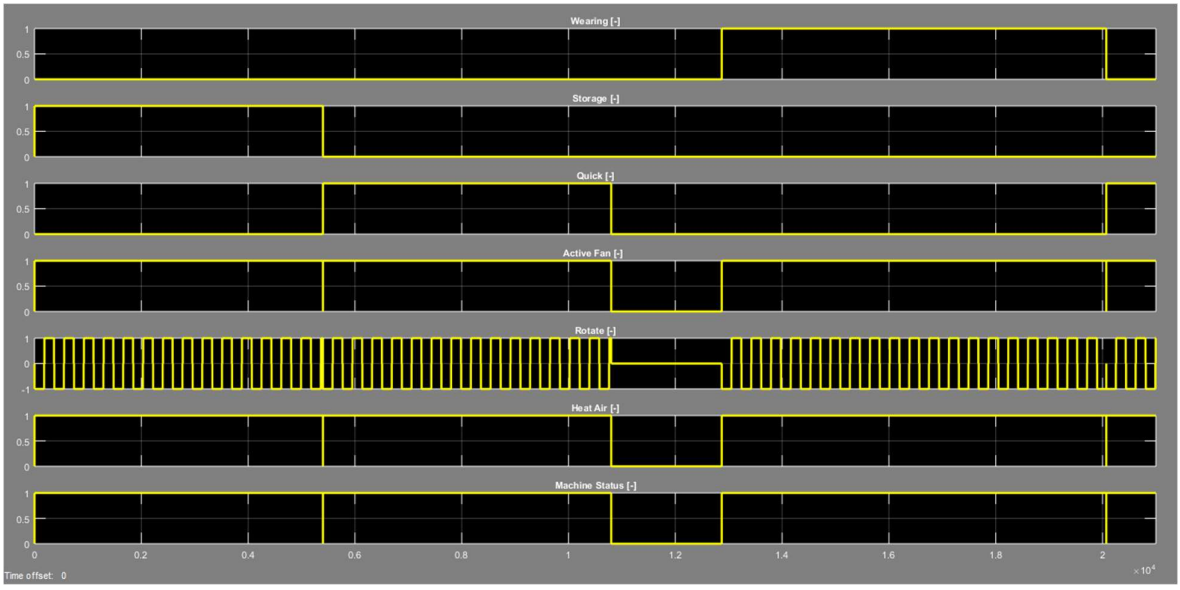


Fig. Outputs of Drying Machine

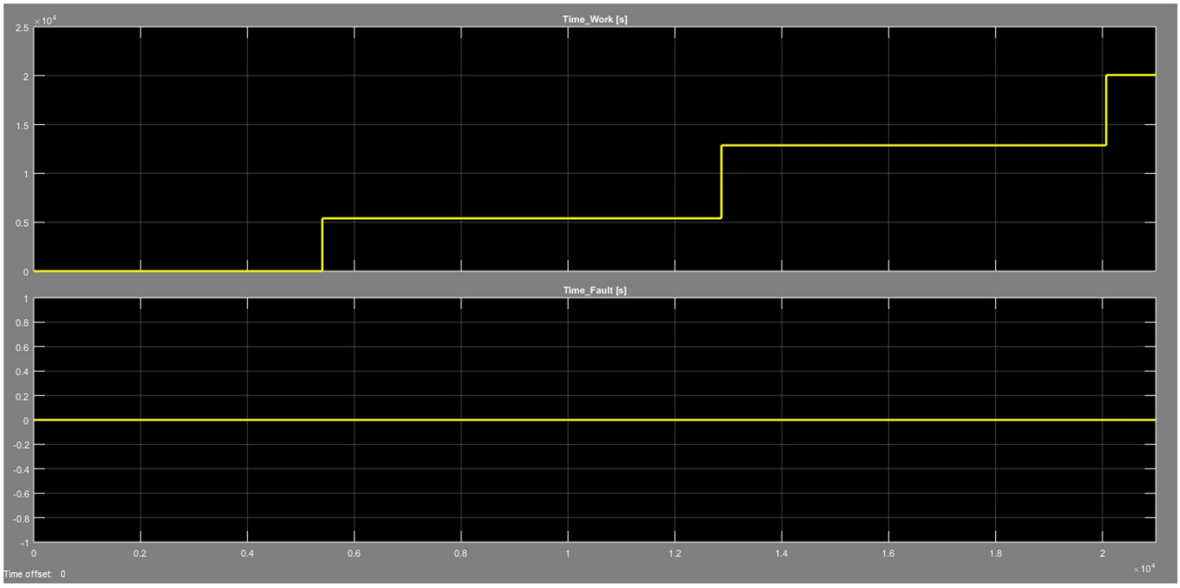


Fig. Outputs of Drying Machine – Time

Test case 2 – Fault Status:

- I change the Air Flow and the Air Temp to see what happened with the outputs;

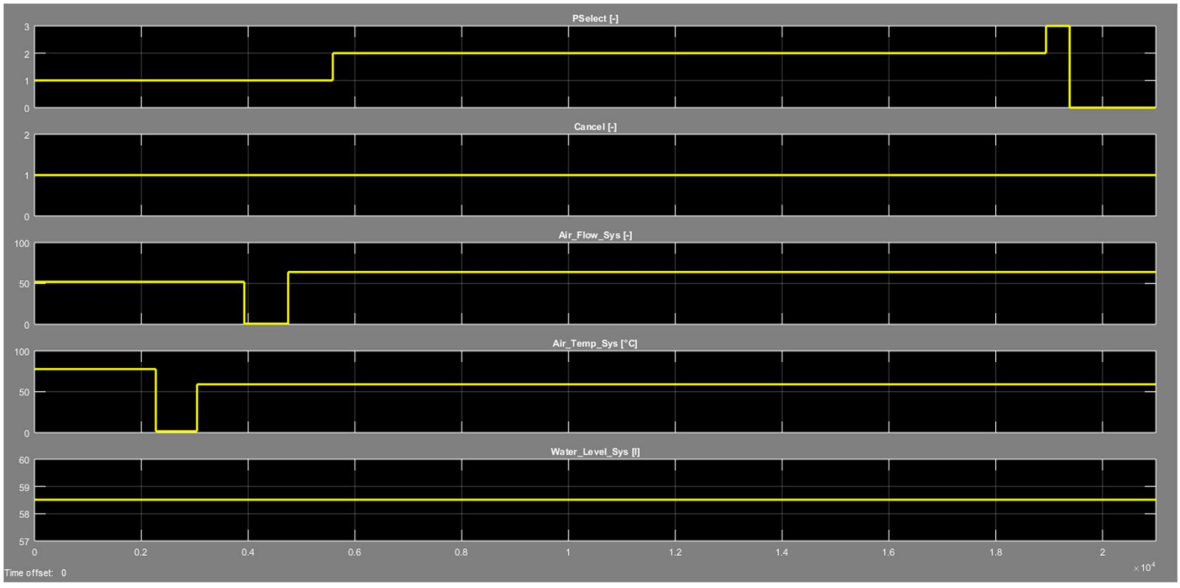


Fig. Inputs of Drying Machine

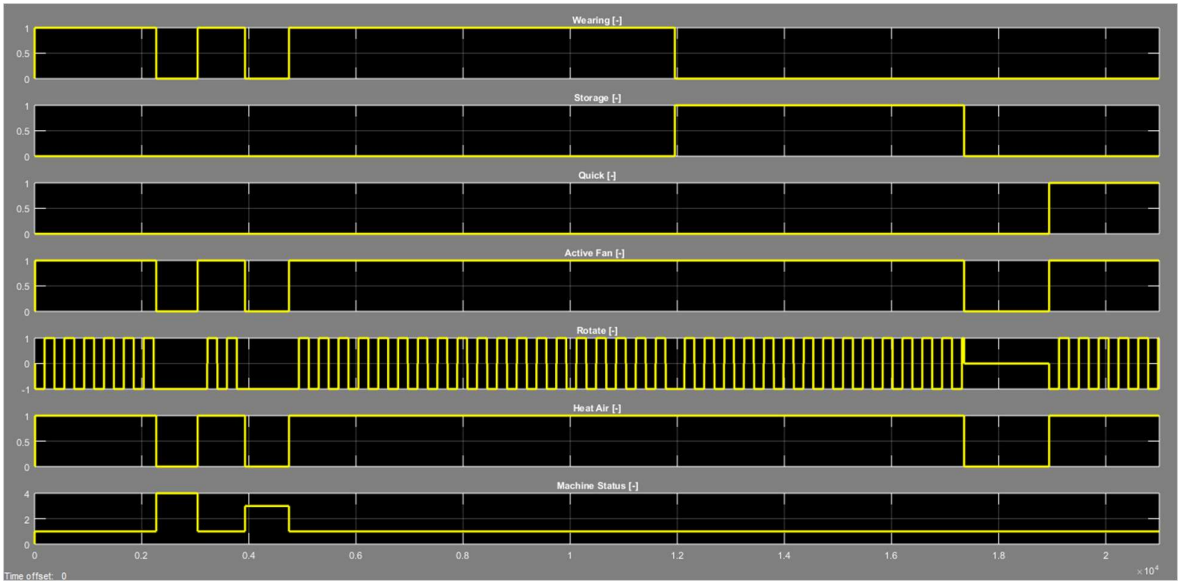


Fig. Outputs of Drying Machine

Test case 3 – Fault Status:

- I change the Air Temp and Water Level;
- After 120 sec detect the error Heater Fault, the program works for 120 sec.

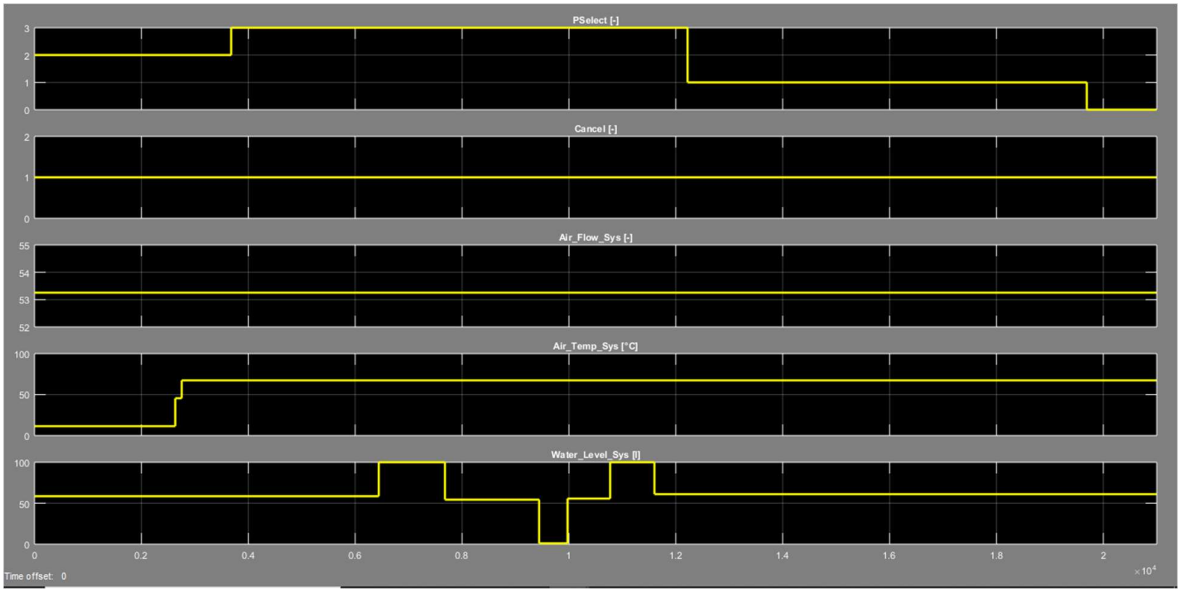


Fig. Inputs of Drying Machine

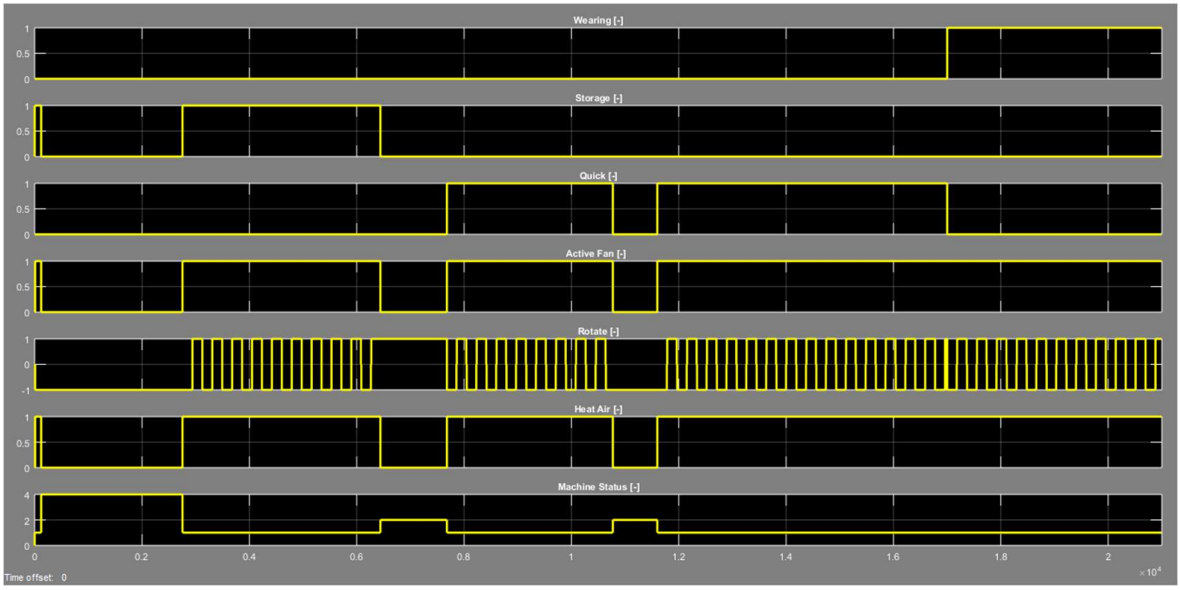


Fig. Outputs of Drying Machine

Test case 4 – Fault Status:

- I modify the button Cancel and Water Level;

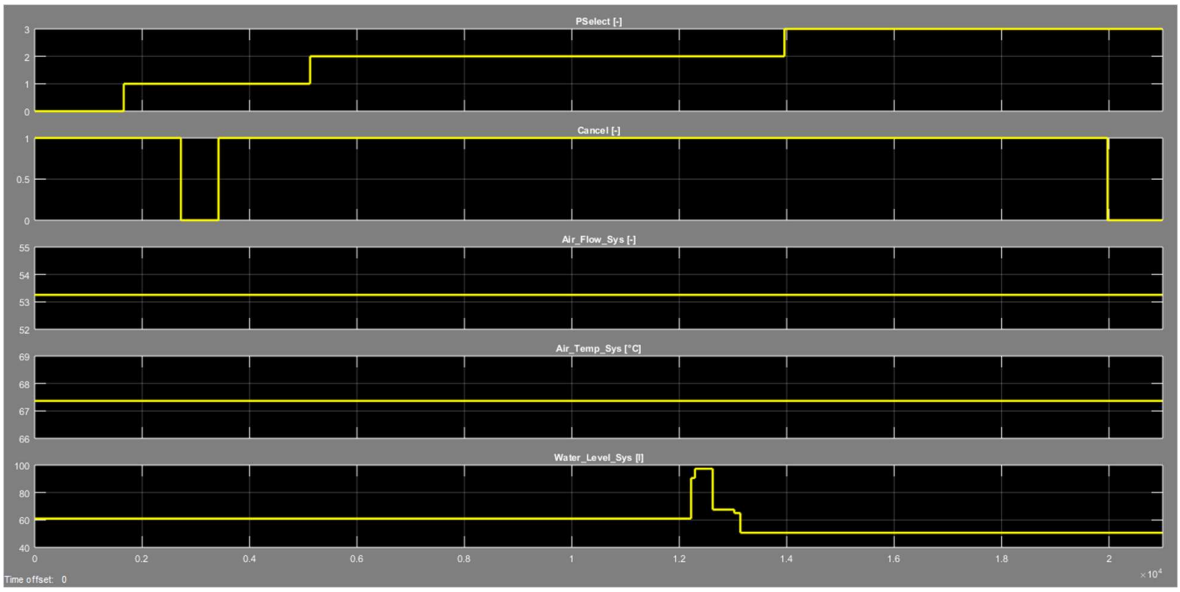


Fig. Inputs of Drying Machine

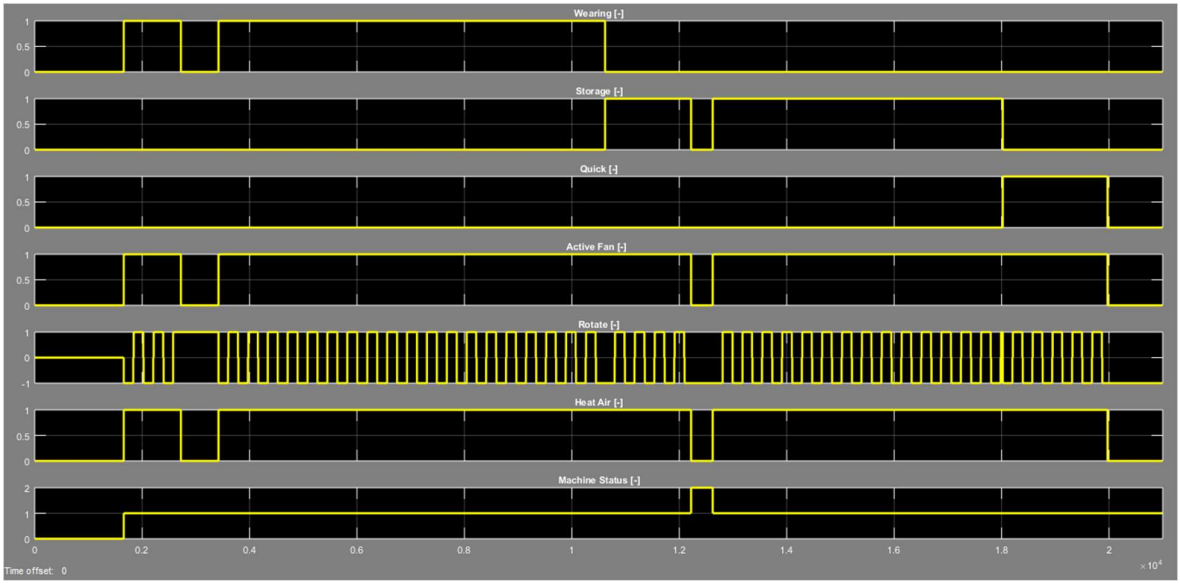


Fig. Outputs of Drying Machine

Another requirement says that the values for WaterLevel, AirFlow and AirTemperature should be between 0 and 100. To do this select the min. and max. from Model Explorer -> Select variable -> General -> Limit range, or do a script who displays a message when you are out of range.

Test case 5 – Fault Status:

- **Water Level Error,** after I pause the program because water level is bigger than 90, continue the program.

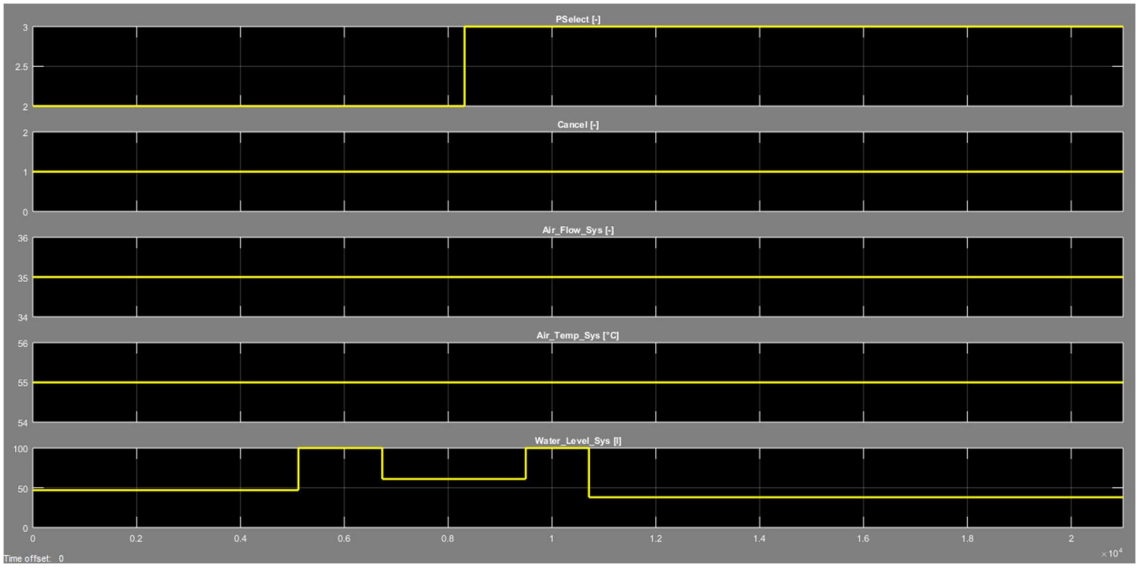


Fig. Inputs of Drying Machine

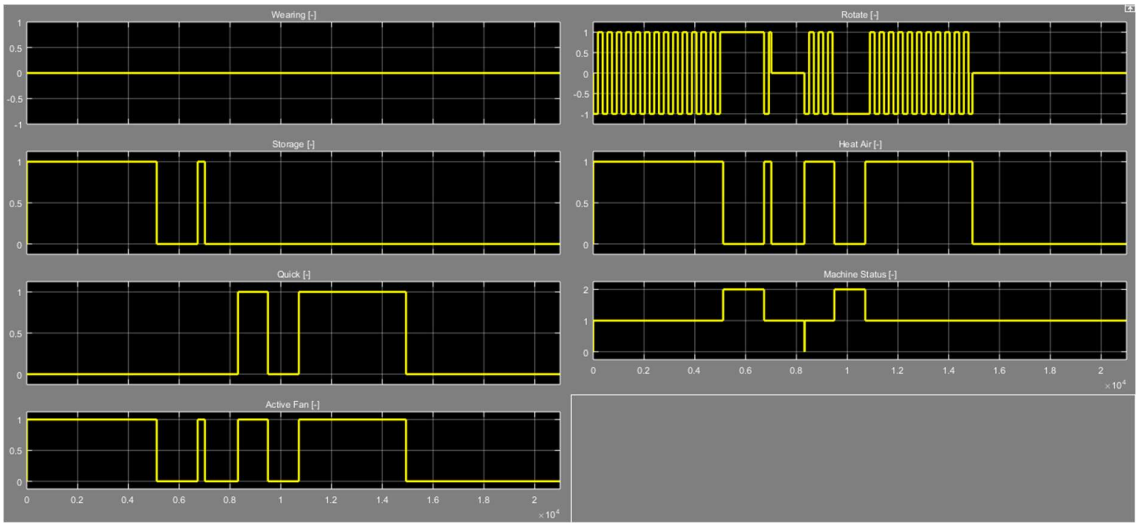


Fig. Outputs of Drying Machine