

Centralized and Decentralized Dynamic Resource Allocation Experiments

Nicanor Quijano and Kevin M. Passino

Department of Electrical and Computer Engineering

The Ohio State University

2015 Neil Avenue, Columbus, OH 43210-1272

April 26, 2004

Contents

1	Introduction	2
2	Laboratory Procedures	2
2.1	Necessary Equipment	2
2.2	Connections	2
3	Procedures	2
3.1	Centralized Controller	3
3.2	Decentralized Controller	4
4	Post-Laboratory Exercises	4

1 Introduction

In this laboratory we will implement some of the strategies developed in [1], that is available online at http://www.ece.osu.edu/~passino/tesis_final.pdf.

2 Laboratory Procedures

2.1 Necessary Equipment

- dSPACE software.
- 1 Power supply Tektronix.
- DS1104 interface card.
- Quanser DS1104 interface board.
- 5 Analog input cables.
- 1 Digital I/O cable.
- 5 LM35 CAZ.
- 2 DS2003 (Darlington Device).
- 5 Lamps.

2.2 Connections

You are going to connect the experiment as it is shown in Figure 1. Remember that the current for each of the lamps is provided by the DS2003 that you have to connect as you did in Lab 2. Since each lamp consumes 100 mA, and the DS2003 can provide only 350 mA, you will need two of them connected such that one provides the necessary current for 3 lamps, and the other two lamps will get the current from the other one. Each of the lamps will be turned on using digital outputs.

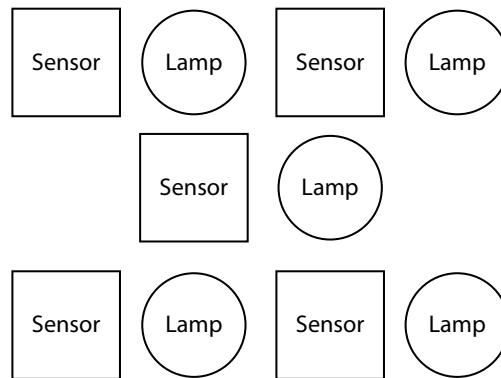


Figure 1: Physical distribution of the lamps and the sensors for this lab.

3 Procedures

As you can see in [1, 2] we have developed different strategies for a similar temperature control experiment. Now, you are going to test some of these strategies for the physical layout in Figure 1. A zone is composed of a lamp and a sensor; hence, there are five zones in Figure 1.

3.1 Centralized Controller

The task is to regulate the temperature to be uniform, but with some fixed maximally elevated value. However, there is a constraint on the number of lamps that you can turn on at a time. For this case, we think of the current to turn on each light as a resource which must be conserved. Then, with some sensing constraints we try to allocate the heat energy from the lights driven by the current to the zones that will result in achieving a uniform maximized temperature.

The first control strategy simply applies at each time instant a 100 ms duration pulse (which is the same as the sampling time) to the zone that has the lowest temperature. We are trying to allocate a “resource” (the time that the light is on on current available for heating and hence the current to the heaters).

1. Open a new Simulink model.
2. Put the necessary blocks to read the data that the sensors acquire. Remember that the output of the sensors is a voltage that is proportional to the temperature in degrees Celsius.
3. You can put a filter similar to the one that you used in Lab 2. Label the output of each filter as temp*, and this is the temperature that you will use for the controller.
4. Each lamp will be turned on/off by software using digital outputs.
5. If you want to start taking data from time 0.0 on, you have to do the following. In the Real-Time Workshop tab, instead of having *make_rti* write *make_rti SSTATE=STOP*¹. Then, when you build your layout in dSPACE you will have to put a RadioButton that will control the state of the simulation, i.e., you will start or stop the experiment by yourself (remember that 0 is equivalent to have the SSTATE=STOP, and 2 is equivalent to have the SSTATE=START).
6. The controller that you will build will take the zone with the minimum temperature and will apply a pulse of 100 ms to the lamp corresponding to that zone. In this case, you will use a sampling time of 100 ms.
7. Compile your model.
8. Build a New Experiment in dSPACE and add all opened files to it.
9. Your layout has to have at least a RadioButton to start and stop the experiment (as mentioned above), and 5 plotters where you will see the temperatures for each zone. You can also put some other elements that indicate the status of the lamps (for instance, you can use a MultiStateLed).
10. Take data for at least 300 seconds. Store the temperature and the state of the lamps.

This first controller is a centralized controller without delays. In the next part, we are going to add delays and in the post-laboratory exercises you will compare the two results.

There are two different ways to apply a delay of T_d seconds for this case. One is to apply the pulse of 100 ms to the zone that had the minimum temperature T_d seconds before. The second option is to delay the output, i.e., instead of applying continuously the pulse of 100 ms to the zone that has the minimum temperature, you will apply this pulse and then you will wait for T_d seconds before applying another pulse to the zone that has the lowest temperature. Build these two options in two different models, compile them and acquire the data for at least 300 seconds. To understand better what is happening in this case, it is suggested to vary the T_d values (e.g., you can take 0.5, 1, and 5 seconds as your delays such that you can compare the results afterwards).

¹Thanks to Vivake Asnani and David Orn Sigthorsson for this suggestion.

3.2 Decentralized Controller

In this section we assume that there is a “local” controller for each zone that can make decisions about whether to turn its heater on or off only based on the temperature in its own zone and possibly those in some “adjacent” zones, but only by obtaining such sensed temperatures from “neighbors” over a communication network that may have delays. Therefore, in this case we are going to be able to turn on more than one lamp at a time.

1. Define a topology similar to the one in [1, 2].
2. You are going to develop a decentralized controller where each of the five local controllers will turn on the lamp if its temperature value is lower than all its neighbors (“neighborhood” is defined by your topology).
3. Repeat the same steps to compile and create a new experiment.
4. Acquire data for at least 300 seconds.

To study the effects of delays in the decentralized control experiment, you have to modify the control strategy studied before. Now, instead of storing the data directly from the data acquisition, you have to delay the value acquired. Therefore, you are going to apply a pulse of 100 ms duration to the zone that had the minimum temperature in the neighborhood defined by your topology, T_d seconds before. Try to acquire data for different T_d values, and compare the results that you obtained in this case with those obtained in the centralized case.

4 Post-Laboratory Exercises

1. Explain briefly the model developed for the centralized case without delays.
2. Do you obtain after 300 seconds a uniform temperature in all the zones? Explain.
3. Explain briefly the model developed for the two delay cases on the centralized experiment.
4. What happens when you start increasing the delay?
5. Justify the choice of your topology in the decentralized case.
6. Explain briefly the model that you design for this case.
7. After 300 seconds, do you obtain the same temperature in all the zones? Why?
8. What is the effect of the delays for this case?
9. Which controller gives you the most uniform temperature in all the zones without delays? And with delays? Explain.

References

- [1] N. Quijano, “Experiments and technologies for decentralized temperature control,” Master’s thesis, The Ohio State University, 2002.
- [2] N. Quijano, A. E. Gil, and K. M. Passino, “Experiments for decentralized and networked dynamic resource allocation, scheduling, and control,” *Submitted for Publication, IEEE Control Systems Magazine*, 2004.