The Balls-in-Tubes Experiment

Alvaro E. Gil, Nicanor Quijano, and Kevin M. Passino
Dept. Electrical and Computer Engineering
The Ohio State University
2015 Neil Avenue, Columbus, OH 43210

April 5, 2004

Abstract

This document provides a general guide for students who want to implement real-time control strategies for the balls-in-tubes experiment. This inexpensive experiment is designed to be a testbed for the implementation and evaluation of distributed dynamic resource allocation strategies. In this document, we describe the apparatus, its main components, the challenges that we need to face, and we show how to interface dSPACE with the experiment.

Contents

1 Introduction

The ubiquitous presence of networked computing is significantly impacting the field of control systems. There are already many “distributed control systems” (DCS) in industry and significant current research on networked multi-agent systems. These networked control systems are typically decentralized, large-scale, may be hierarchical, and are often quite complicated from a dynamical systems perspective. They typically have a blend of significant nonlinearities, nondeterministic behavior, random delays, constrained information flow (e.g., only via the topology defined by the communication network), high-dimensionality, etc. Since their main purpose is control of a dynamical system they contain many (if not all) of the challenges typically found in control systems (e.g., disturbance rejection, tracking, robustness), and additional challenges due to the presence of a computer network that synergistically interacts with the dynamical system. They represent a significant departure from typical, say classical DC motor control or inverted pendulum control problems, and demand many of the same tools/skills and more such as expertise in software engineering, object-oriented programming, or real-time operating systems. Moreover, they demand that more attention be given to a number of other nontraditional control objectives, including dynamic resource allocation, scheduling of tasks, and control over large networks, than in the past.

The balls-in-tubes experiment was designed to provide a testbed for distributed networked feedback control systems. Compared to past experiments, it provides interesting challenges to perform dynamic resource allocation.

2 dSPACE: Hardware and Software

We use dSPACE hardware and software for the balls-in-tubes experiment experiment described in this document. The dSPACE software is based on Matlab/Simulink. To develop the block diagrams in Simulink for the balls-in-tubes experiment we use several processes. First, we will acquire the data, and store this information in some global variables that will be used in the next process. Second, we develop several subsystems (i.e., the name used in Simulink to designate each of the elements that a complex block diagram has to make everything more compact and easy to read) that will be in charge of making decisions concerning the control, or other tasks such as to update variables to be used in the last process. Third, we will update the digital/analog outputs.

In the combined dSPACE-Matlab package we have Simulink and the graphical user interface (GUI) that is provided in dSPACE. In Simulink we develop the controller and all the necessary functions to run the experiment. Once we have the code, that they call the “model,” we compile it and following some steps that are transparent to the user, we obtain a file that will run the code in real time, and provide the ability to set up a user interface. This GUI in dSPACE can be viewed as a diagnostic tool, since we can change some variables, and we can see in real time some of the variables defined by the user in the model. The students can find a tutorial introduction to dSPACE at http://www.ece.osu.edu/˜passino/dSPACEtutorial.doc.pdf.

3 Experimental Apparatus and Challenges

This experiment was designed to be an inexpensive testbed for dynamic resource allocation strategies. Below, we describe the elements of the experiment, its challenges, and we show how to interface both the sensors and actuators to the dSPACE card. Figure 1 shows the balls-in-tubes experiment. There are four tubes, each of which holds a ball inside, a fan at the bottom to lift the ball, and a sensor at the top to sense the ball’s height. For each tube there is a box that the fan pressurizes. You can think of this box as a stiff balloon that is "blown up" by the fan, but which has an outlet hole used to blow air into the tube. The tubes are connected at the fan inlets via an input manifold which has an inlet at the bottom as indicated. Also, there is an output manifold at the top of the tubes with an outlet as shown. The presence of the manifolds is a key part of the experiment. These manifolds force the sharing of air at the input, or constrain its flow at the output, so that there is significant coupling between the four tubes. Characteristics of the coupling can be adjusted by, for instance, making both the inlet and outlet have different opening sizes or by placing solid objects in the manifold to obstruct airflow to some fans. For a range of inlet sizes, if one fan succeeds at lifting the ball to near the top, it can only do this at the expense of other balls dropping. This feature leads to the need for "resource allocation" where here the resource is the air that elevates the balls. Flow characteristics in the manifolds are very complicated due to, for instance, air turbulence in the manifolds and pressurized box. Finally, note that the experiment was designed to be easily extended to more tubes, different arrangement patterns, and to have different manifolds and hence interaction effects. There are a number of control objectives and challenges that can be studied for this experiment, beyond the obvious isolated balancing of a

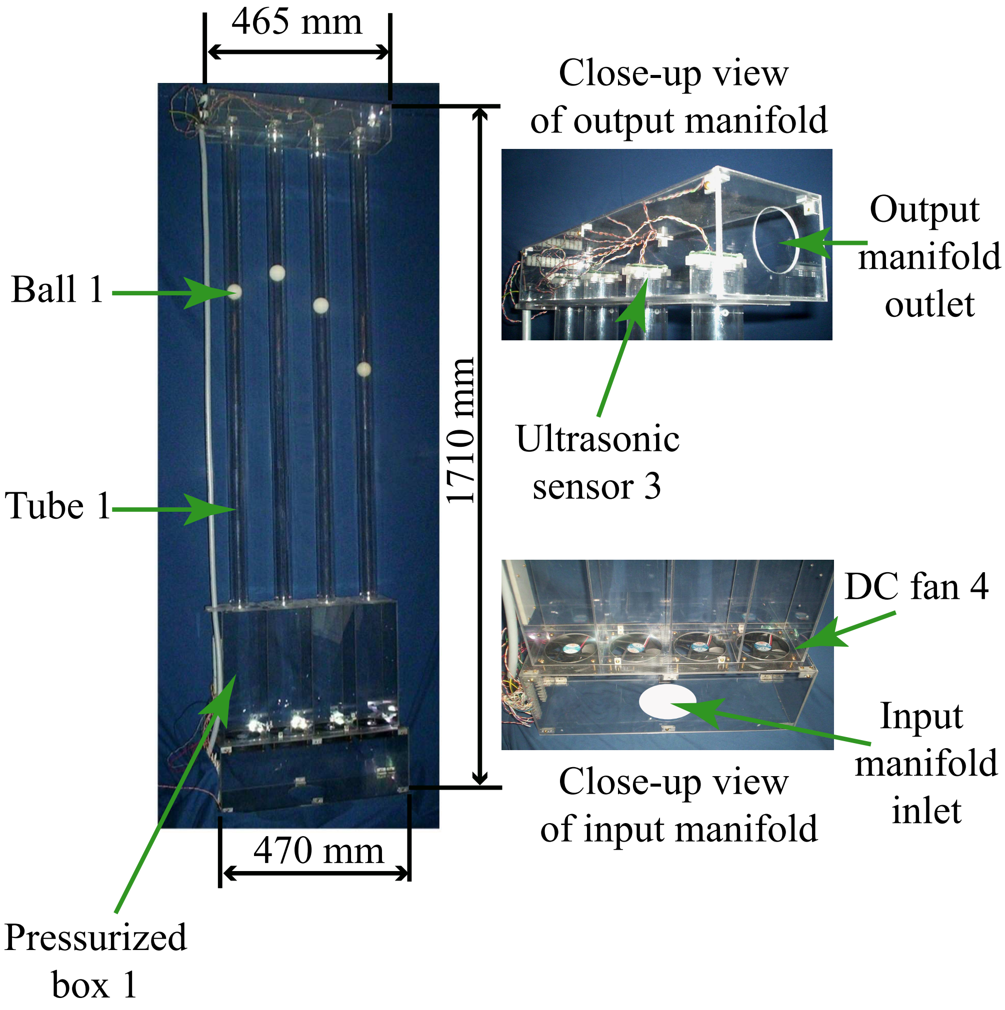


Figure 1: Balls-in-tubes experiment (tubes 1-4, numbered left to right).

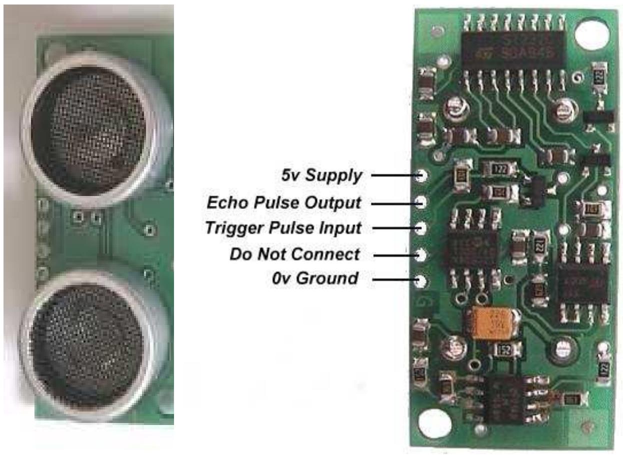
single ball in a tube:

1. Balancing the balls inside the tubes, trying to allocate air pressure to keep all the balls at fixed positions or alternatively, a uniform height but maximally elevated.
2. Balancing and reallocation dynamics in the presence of disturbances such as changes in manifold inlet sizes or flow obstructions in a manifold. Also, effects of using decentralized and networked decision making in the presence of an imperfect communication network could be studied.

The actuators that we selected are DC fans commonly found inside computers. In total, there is one digital input (DI) and one digital output (DO) for each sensor, and one digital output for each fan for a total of 12 digital input-output lines that we connect to a DS1104 dSPACE card.

3.1 Height Sensors

To sense the ball height for each tube we use a Devantech SRF04 ultrasonic sensor shown in Figure 2(a). Figure 2(b) depicts the sensor connections. Table 1 shows the specifications of the Devantech SRF04 sensor (http://www.robot-electronics.co.uk/htm/srf04tech.htm). The raw data obtained from the sensor is very noisy. For example, the spikes that we get from it correspond to errors greater than 10 centimeters. We developed a filter to smooth the sensor output. For our sampling rate $T_s = 100\mu$ sec., after filtering we achieved a resolution of ± 1 centimeter. The ultrasonic sensor timing diagram is shown in Figure 3. There are two digital signals (one DI and one DO) needed to obtain raw data from this sensor, which is the travel time of the ultrasound wave between the sensor and the ball. The DO is connected to the trigger input to start the ball height measurement. After the application of a signal to the trigger input, the sensor will send out an 8 cycle sonic burst at 40kHz and set its echo line high. It then waits for an echo, and when it detects it, the echo line is reset. The pulse width of the echo line is therefore proportional to the ball height and it is connected to a dSPACE DI.



a. Devantech SRF04 Ultrasonic sensor

b. Sensor connections

Figure 2: Ultrasonic sensor used in the balls-in-tubes experiment.

Table 1: Specifications of the ultrasonic sensor.

Voltage (DC)	5v
Current	30mA Typ. 50mA Max
Frequency	40KHz
Maximum Range	3 m
Minimum Range	3 cm
Weight	0.4 oz.
Size	1.75” w x 0.625” h x 0.5” d

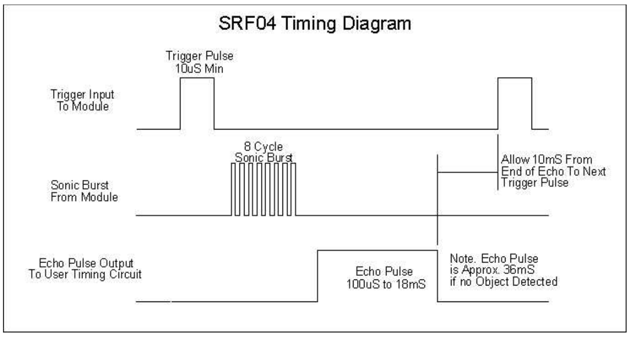


Figure 3: Timing diagram of the ultrasonic sensor.

3.2 Actuators

The actuators that we selected are Dynatron DF1209BB DC fans (Figure 4) commonly found inside computers and their specifications are shown in Table 2. We use a pulse width modulation (PWM) signal as an input to the fan. The sampling period is 100 $\mu sec.$ the period for the PWM is 0.01 sec., and the duty cycle is varied in order to command the ball inside the tube either to reach a desired height or to raise as much as possible during a certain time interval. One limitation present in the actuators is that they possess an internal temperature sensor that changes the revolutions per minute (rpm) of the fan when the ambient temperature changes; this fact makes the rpm of the fan temperature-dependent. For single runs of the experiment this is not a problem since the room temperature is relatively constant in such a short time interval. Another more significant problem is that the fans are of relatively low bandwidth.

Table 2: Specifications of the Dynatron DF1209BB DC fan.

Voltage (DC)	12v
Range Voltage (DC)	10.2-13.8v
Max Current	300mA
Operating Temperature	-10C to + 65C
Weight	80 gm
Size	92mm w x 92mm h x 25.4mm d

Figure 4: Fan used in the balls-in-tubes experiment.

3.3 Electrical Connections

Here we show all the electrical connections in this experiment. Figure 5 shows a side view of the apparatus where you can see all the terminal blocks. There are two terminal blocks (i.e., TOP-1 and TOP-2) on the output manifold, three terminal blocks (i.e., BOT-1, BOT-2, and BOT-3) on the input manifold, and one terminal block on the pressurized block. Figures 6, 7, and 8 show detailed information on all the connections found in the output manifold, pressurized block, and the input manifold respectively. The nomenclature used in this case lists first where the connection comes/goes from/to, and second also lists the terminal block in parenthesis (if applicable) associated with this connection.

3.4 Power Supply

Two power supplies are used in the balls-in-tubes experiment. A 15 DC volt power supply is used to power all the fans. This power supply was designed by undergraduates in an EE 682P design project. You must use one 15 DC voltage source for powering just one fan; otherwise, an overload could occur causing a malfunctioning in the fans. The other one is a 5 DC voltage power supply Tektronix CPS250, which powers all the ultrasonic sensors.

3.5 Simulink Blocks

Here we explain the Simulink blocks that the students need to interface to the height sensors and the actuators. Next, we describe these key blocks in more detail.

3.5.1 Height Sensors

The height of the ball inside the tube is determined by means of the Simulink blocks shown in Figure 9. Notice that the logic implemented here corresponds to the specifications of the timing diagram depicted in Figure 3. The block “Pulse Generator” sends out a signal with amplitude 1, frequency 0.05 s, and duty cycle 0.02 to the trigger input of the sensor. All these values were

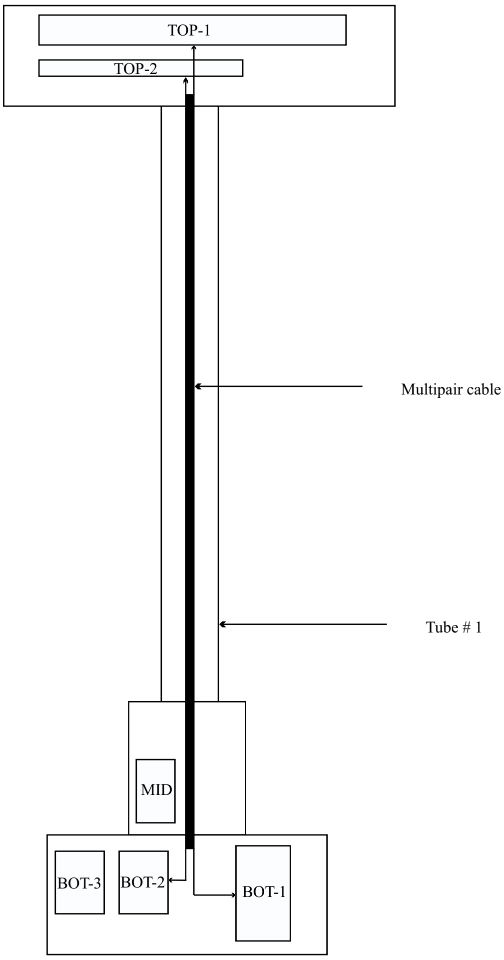


Figure 5: Side view of the terminal blocks in the balls-in-tubes experiment.

determined from the timing diagram of the ultrasonic sensor. The block “EchoLine” is the digital input used for the echo line of the ultrasonic sensor. The travel time of the sound wave is the difference between the negative edges of the trigger input signal and the echo line of the ultrasonic sensor. We then use this travel time divided by two and the sound velocity (i.e., $331.1\frac{m}{s}$) to obtain the height of the ball. The block “Discard Values” avoids having negative values at the variable “Travel time of sound wave” at every time.

We have to cope with three limitations presented by this sensor: changes of the sound velocity with respect to the ambient temperature, the sensor noise, and the resolution of the sensor. If the temperature in the laboratory changes, then the absolute value of the ultrasonic sensor changes also, which makes the height measurement temperature-dependent. The raw data obtained from the sensor is very noisy so we have to filter these data in order to reject undesirable values. We consider that a signal is “noisy” here when the difference between two consecutive ultrasonic sensor readings is greater than 10 centimeters. Hence we have two variables: the actual measurement that could be corrupted, and the filtered measurement. If the measurement does not have noise, we update the second variable (i.e., we assign the same value that the actual measurement has), otherwise we will not, and the value will be the same as the previous iteration. The resolution of the sensor depends on the sampling time used in the experiment. In this particular case, we chose

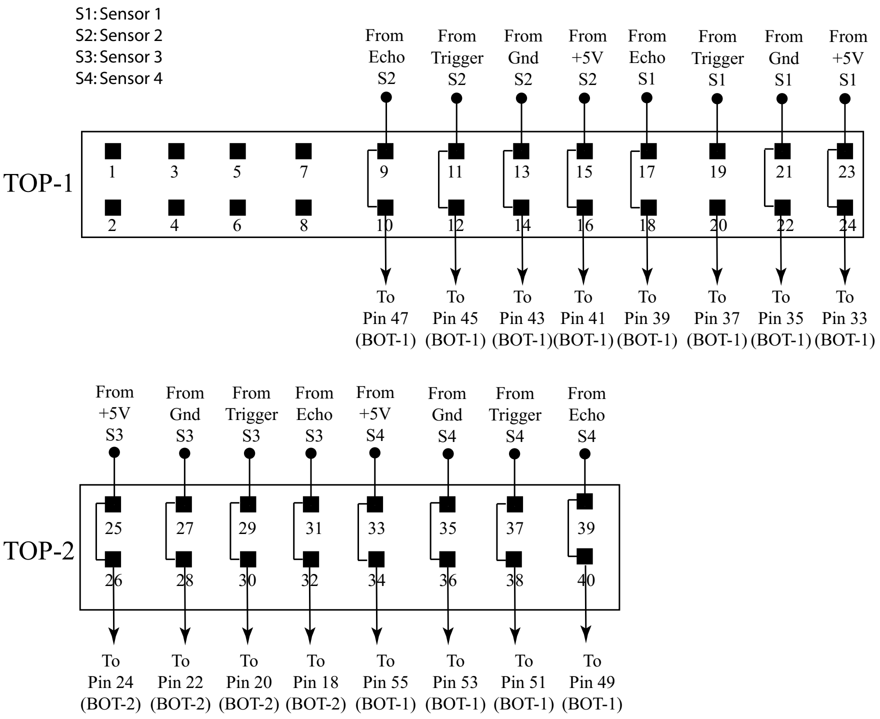


Figure 6: Electrical connections at the terminal blocks in the output manifold.

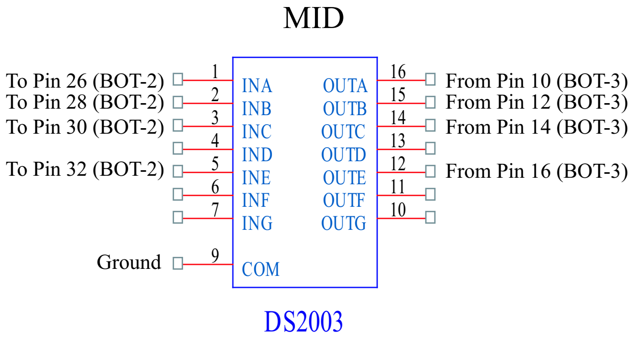


Figure 7: Electrical connections at the terminal block in the pressurized box.

a sampling time $T_s = 100\mu$sec. and achieved a resolution of the sensor equal to ±1 centimeter.

3.5.2 Actuators

The actuators are used to supply the air needed to lift the ball inside the tube. The amount of air supplied for lifting a ball is proportional to the voltage applied to the fan, which is proportional to the duty cycle of the PWM signal. Figure 10 shows the blocks necessary to generate a PWM signal. What you need to do here is to assign a value between 0 and 1 (representing from 0 to 100 % duty cycle) to the memory block “A1” depending on the voltage that you want to apply to a fan. This duty cycle value will depend on the controller design that you will use for the ball-in-tubes experiment.

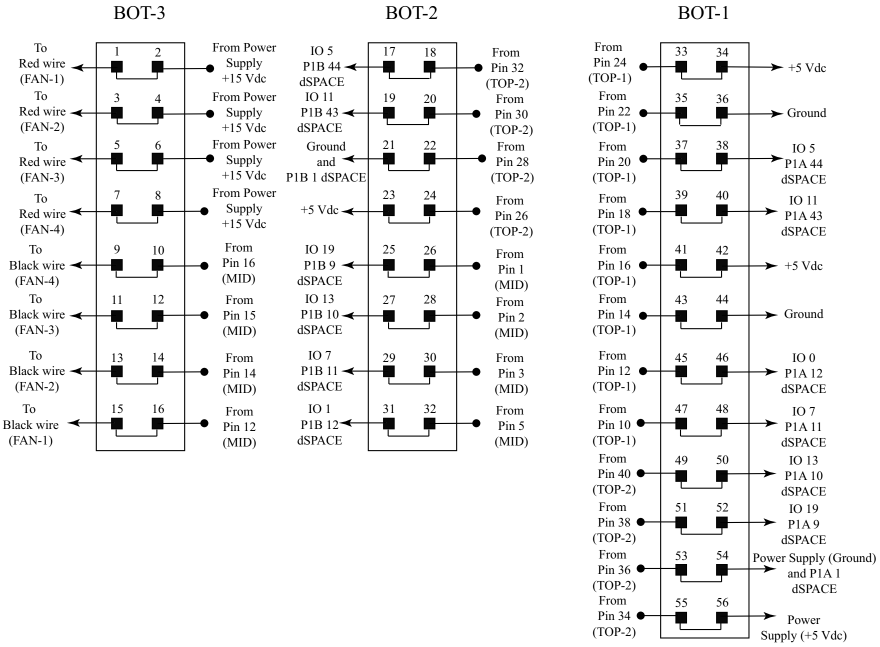


Figure 8: Electrical connections at the terminal blocks in the input manifold.

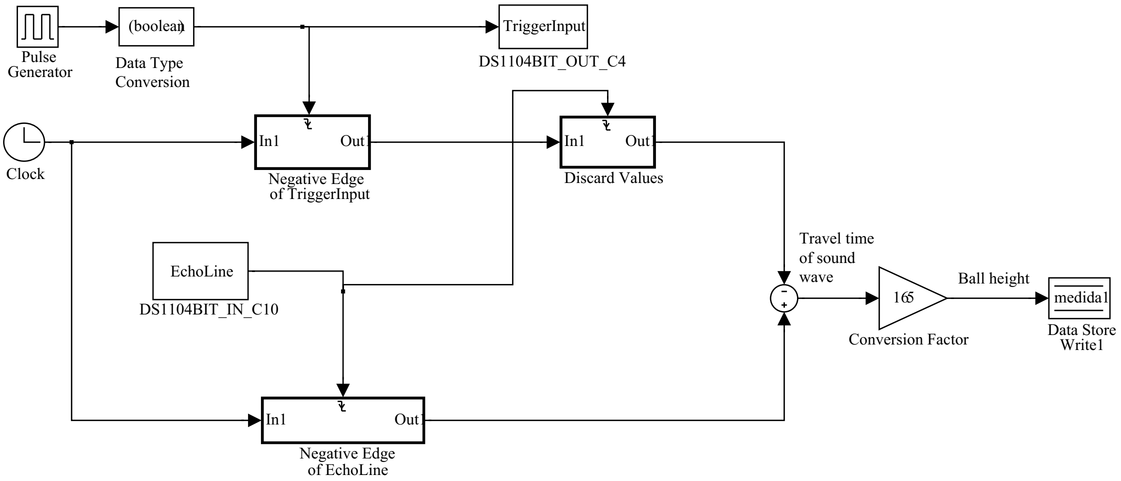


Figure 9: Simulink blocks used to determine the ball height.

3.5.3 Controllers

Two control strategies will be discussed here to give you an idea of how controllers operate for this plant. First, a conventional controller can be designed to float the ball inside each tube at a desired height. For this case, you need to define for each tube a variable in Simulink for the desired height, take the difference between the desired height and the actual one, and this value will be the input to the designed controller. Other additional inputs can be considered depending on whether you want to design a proportional-integral or proportional-integral-derivative controller. Second, you can try to balance the balls around a certain common height. In this case, the peak-to-peak oscillations of each ball height are allowed to be relatively large, but we want the time averages of

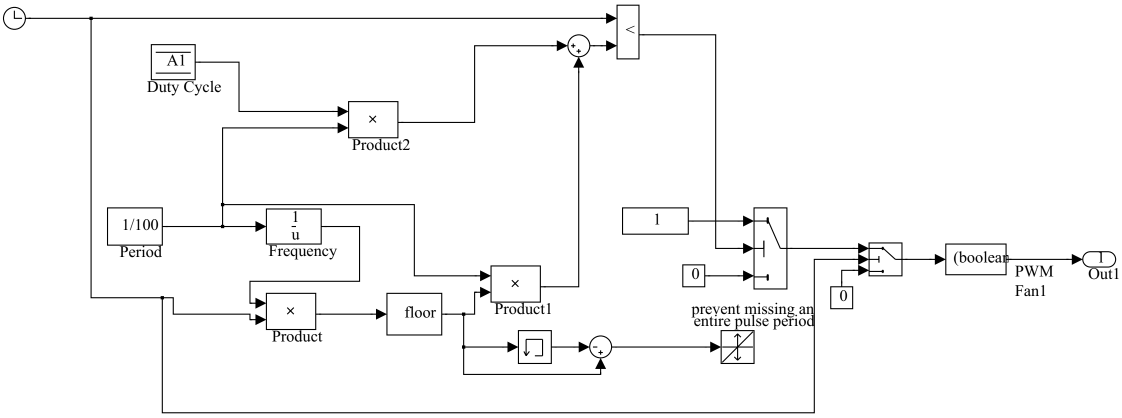


Figure 10: Simulink blocks used to generate a PWM signal.

the heights of each of the four balls to be the same. This goal can be achieved by allowing only one fan enough percentage of PWM duty cycle to raise its ball at a time; hence we think of this as juggling the balls. Such juggling is a type of allocation of the air resource from the common input manifold to each ball. The allocation strategy must be designed to persistently seek to minimize the differences between the average ball heights in spite of air turbulence, inter-tube coupling via the manifolds, fan bandwidth constraints, and significant sensor noise. The dynamics of allocation rely critically on feedback information on ball heights, and the distributed decision-making that implements the allocation strategy.

For the details of the design and operation of two resource allocation strategies for this experiment see [1].

References

[1] N. Quijano, A. E. Gil, and K. M. Passino, “Experiments for decentralized and networked dynamic resource allocation, scheduling, and control,” *Submitted to IEEE Control Systems Magazine*, 2003.