The Electromechanical Arcade

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 scheduling strategies for the electromechanical arcade. This inexpensive experiment is designed to be a testbed for the implementation and evaluation of networked decentralized scheduling 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 electromechanical arcade was designed to provide a testbed for distributed networked scheduling strategies. Compared to past experiments, it provides opportunities to study cooperation between two control systems.

2 dSPACE: Hardware and Software

We use dSPACE hardware and software for the arcade experiment. The dSPACE software is based on Matlab/Simulink. To develop the block diagrams in Simulink for the arcade we use several processes that (i) acquire the data and store this information in some global variables, (ii) make decisions concerning the control, and (iii) 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. Below, we describe the apparatus, highlight the challenges it presents, and we explain the main Simulink blocks that can be used to implement complex control strategies.

3 Experimental Apparatus and Challenges

This experiment is composed of two main devices: guns and targets as shown in Figure 1. Each of them is provided with a laser and a photodetector. There are in total eight "pop-up" targets that "appear" (indicated by its laser being on) and "disappear" (laser off) at frequencies that are independent of each other (driven by simple timing circuits that can be adjusted by hand to provide different frequencies). The guns can detect a target appearance if its is pointed directly at it (it has no "peripheral vision") via a photodetector mounted on the gun. The two guns are each mounted on the shaft of a motor. Each of these motors has a quadrature encoder that provides the current angular position of the motor. We use a PID controller to point any gun to the target that needs to be shot and these PID loops are well tuned so their performance will not impact the scheduling strategies. The photodetectors of all targets (8 digital inputs), the laser and photodetector of each gun (2 digital inputs and 2 digital outputs), the motor encoder, the output of each PID controller (1 analog output each) are connected to one DS1104 dSPACE card. All the lasers located at the

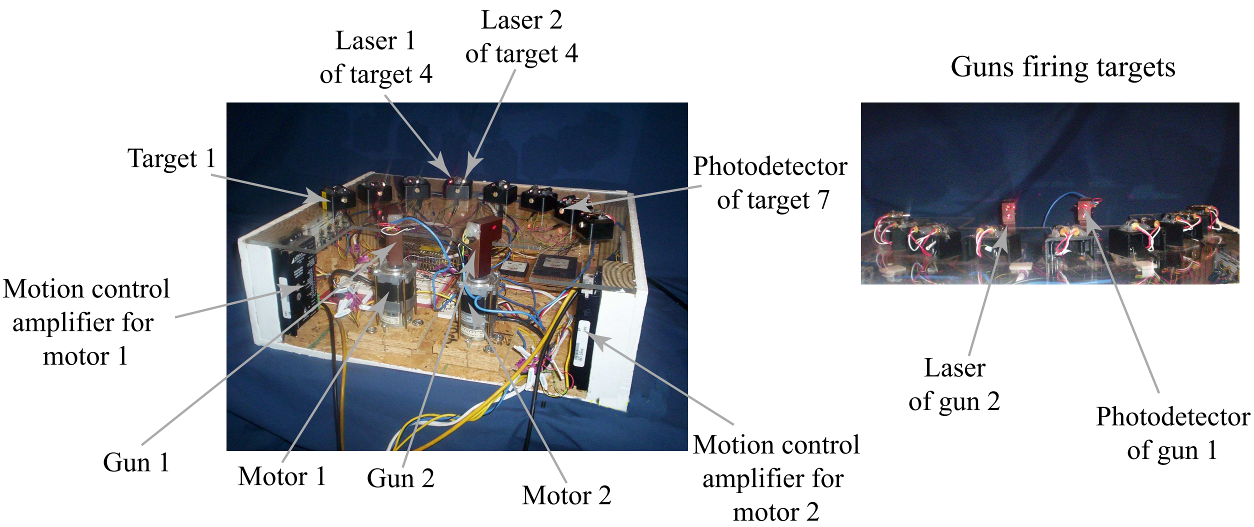


Figure 1: Electromechanical arcade experiment.

targets point to the gun photodetector and if one gun is pointing to a target when it appears, this gun can shoot (turn its laser on) at that target, which triggers the corresponding photodetector of the shot target. When the photodetector of a target is triggered, the gun considers that specific target as "hit" (it gets a point for hitting it) and then the gun will look for the appearance of another target (the target depends on the scheduling strategies implemented by the students). The analogy with arcade games should be clear.

We assume that the guns do not know any information about the rate of appearance of all targets (but one could invent strategies for estimating appearance sequences); however, the guns do know a priori the position of all targets, and the guns can communicate to each other their decisions about the targets that they are currently processing or pursuing. The challenges for this experiment are as follows:

1. To schedule in real-time a sequence of shootings so as to maximize the number of points the team gets. Since target detecting and shooting requires movement of the guns a good schedule will typically minimize the motion of the guns in maximizing point gain. Feedback is required to overcome, for instance, uncertainty about when targets appear (i.e., open-loop precomputed schedules will not be effective).

2. To cooperatively schedule the shooting of the guns in the presence of an imperfect communication network that allows communication between the two guns. While the network could be the internet and a computer could be dedicated to each gun it can also be simulated within one computer. Communication imperfections such as random but bounded delays, bandwidth constraints or message misordering could be considered.

We consider then an "environment" that is highly uncertain (e.g., uncertain target appearance times) *and* where we have imperfect communications that make it difficult for the two guns to coordinate their actions. Due to the presence of so much uncertainty it is generally not possible to accurately predict far into the future, and hence generally not useful to employ optimization approaches to develop long sequences of planned operations either off- or on-line. Note that this decentralized scheduling problem can be thought of as a type of resource allocation strategy. We have implemented in the same computer a routine that simulates a communication network between the two guns where random but bounded communication delays can be generated every time the two guns need to communicate to each other.

3.1 Experiment Layout

The housing of the experiment contains all devices and encloses the wiring. The base and sides of the housing are constructed out of chipboard and have dimensions of 17.25 $\times$ 9 $\times$ 6.5 inches. A thin layer of Plexiglas is used as a lid to make the circuitry of the experiment visible from the top. Figure 2 shows the location of the circuitry used in the experiment, which is grouped in parts designated by the letters A, B, C, and D in the figure.

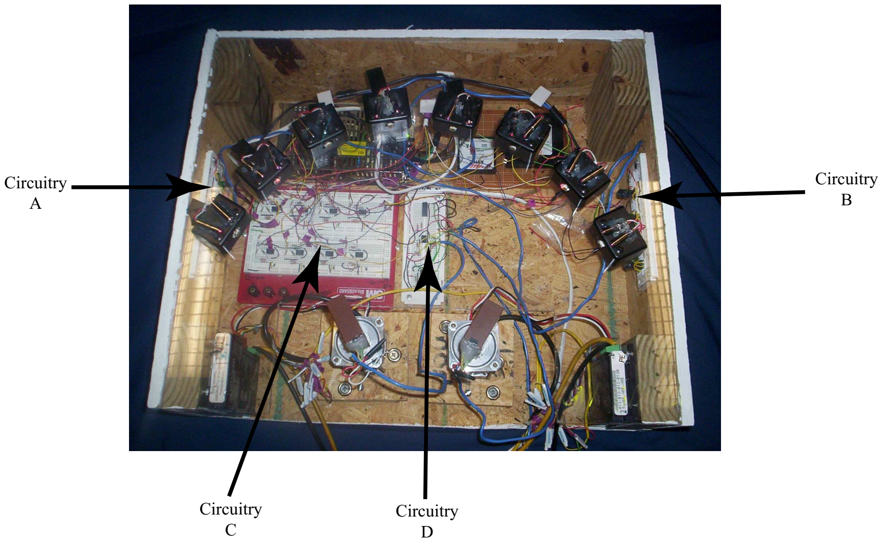


Figure 2: Circuitry of electromechanical arcade experiment.

3.2 Guns

There are two guns in the experiment. Each gun is mounted on the shaft of a motor and provided with one laser (Radio Shack # 277-1101) and one photodetector (Radio Shack # 276-1657). A fabric composite material is used for the guns which includes drilled holes to the size of the shaft that keeps the guns in place with setscrews having a rubber tip to prevent movement and damage to the shaft. Two holes were drilled through the top in order to fit the laser and detector in a manner that two targets lasers would be unable to shoot at each individual gun detector at the same time. It also allowed the positions of the laser and detector to remain unchanged with respect to the horizontal axis. The vertical position can be altered with the set screws upon initial calibration.

Precautions: There are several precautions that must be taken before running the experiment. First, the fabric composite material that contains the guns must be mounted on the shaft of the motors after the program is running in the DS1104 card; otherwise, the motor will move in a rapid manner when the run button is hit in the GUI. This could cause damage to the wiring of the laser and photodetectors of the guns. Second, if the fabric composite material is already mounted on the motors and you want to compile the Simulink program again, then you have to take out the fabric composite material first before compiling the program; otherwise, the wiring could also be damaged.

3.3 Targets

The targets consist of eight photodetectors and sixteen lasers (two lasers per target). Black plastic boxes were screwed into the Plexiglas in an arc shape so that each target is at a significantly different angle to the motor. Holes were made in the black boxes to insert the photodetectors while the lasers were fastened to the top of the black boxes at angles that were positioned to shoot directly at its corresponding gun detector.

3.4 Lasers

As mentioned above, lasers are present in the targets and guns. The frequency and duty cycle of the target lasers are generated by mean of timers. The on-off gun laser signal is generated in dSPACE.

3.4.1 Lasers in Targets

The LM555 timer sets up the frequency and duration of the target lasers. Figure 3(a) shows the LM555 circuit diagram while Figure 3(b) shows its circuit timing diagram. Both the frequency f and duty cycle $Dcyc$ can be manually adjusted by changing the parameters in the following equations

$$f = \frac{1.44}{(R_A + 2R_B)C} = \frac{1}{t_1 + t_2}$$

$$Dcyc = \frac{R_B}{R_A + 2R_B} = \frac{t_1}{t_1 + t_2}$$

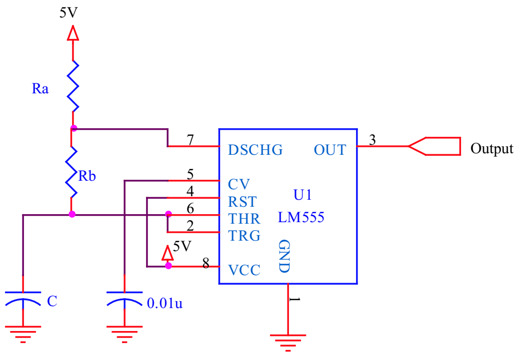
The output of the LM555 timers is connected to the positive terminal of the lasers as shown in Figure 4. Table 1 shows the key components found in the LM555 timer for each target.

3.4.2 Lasers in Guns

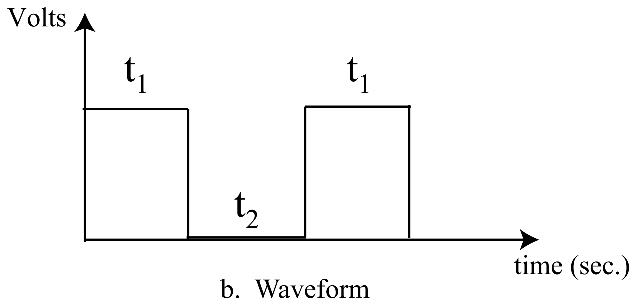
The lasers connected to the motors are controlled by an on-off signal generated in dSPACE. These outputs can be located at sub-D connector P1B-11 and P1B-44 for motors 1 and 2 respectively. Figure 5 shows the connections between dSPACE and the gun laser drivers DS2003.

3.5 Photodetectors

These devices transform the light coming from the lasers to voltage signals by using voltage dividers. When the light intake is low, the voltage divider outputs a high value that is converted by the dSPACE card to 1. When the light intake is high, the voltage divider outputs a low value that is converted to 0.



a. Circuit connection



b. Waveform

Figure 3: LM555 circuit and timing diagram.

3.5.1 Photodetectors in Targets

The dSPACE digital inputs used to monitor the status of the target photodetectors 1-8, as well as the circuitry associated with the photodetectors, are shown in Figure 6. All these signals are used by the dPSACE program in order to detect whether a target has been shot by any gun.

3.5.2 Photodetectors in Guns

The dSPACE digital inputs used to monitor the status of gun photodetectors 1 and 2 in the sub-D connector are P1A-11 and P1A-44, respectively. Figure 7 shows all these connections. These signals are used by the dSPACE program in order to detect the appearance of a target when either gun is current pointing at it.

3.6 Motors

Two inner rotor type 4 pole servo motors (Shinano Kenshi # LA052-040E4N02) are attached to the base of the housing using four screws per motor. The two motors are centered ten inches apart, allowing equal spacing for shooting and each of them is provided with an encoder. The angular position of the motors is obtained from its built-in encoder, which has a resolution of 500 pulses per revolution. The specifications of both the motor and the encoder are shown in Tables 2 and 3

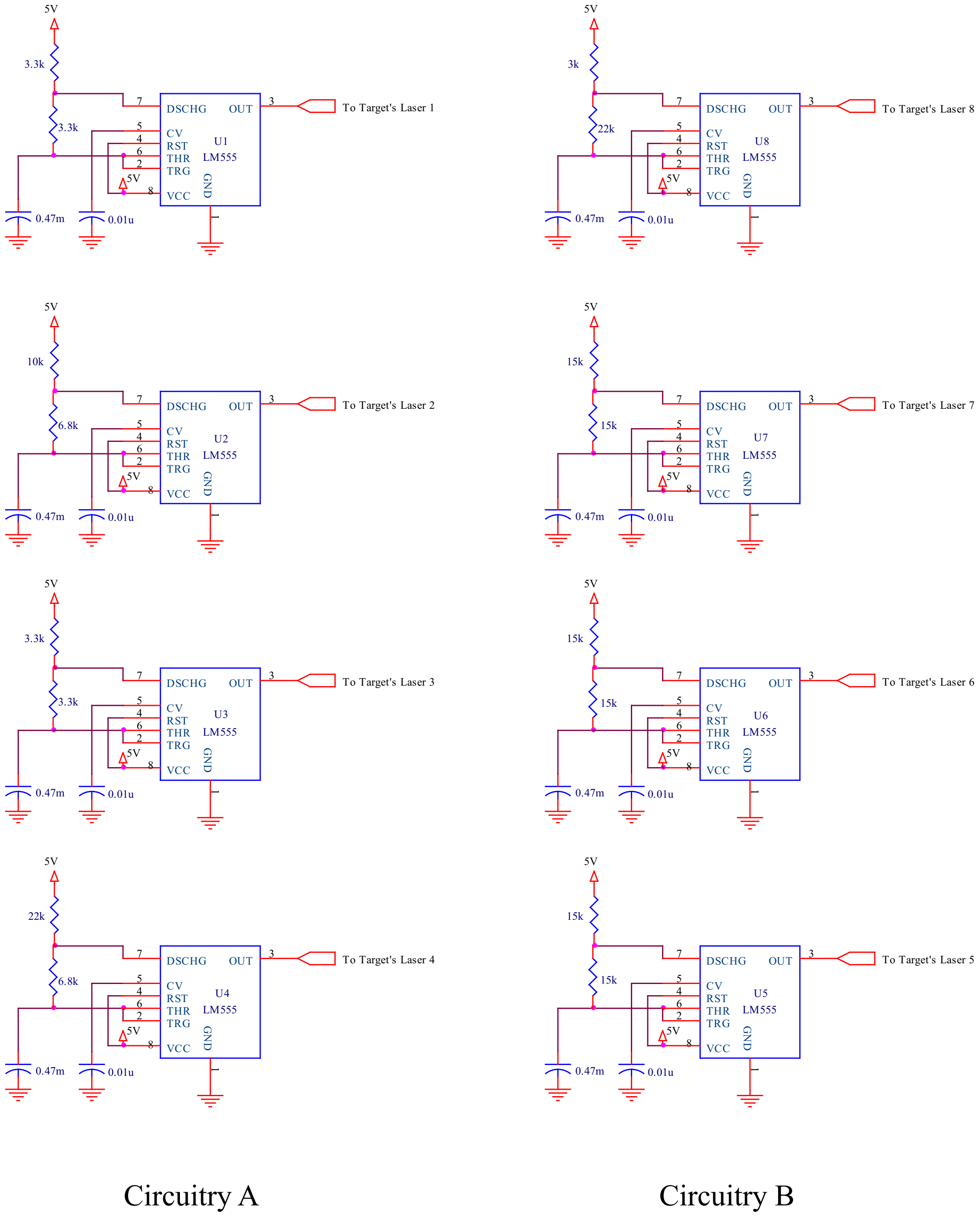


Figure 4: Connection among LM555 timers and photodetector lasers.

respectively. The motors are driven by the Advanced Motion Controls amplifiers (BE12A6) which supply a 3-phase power to the motors. The amplifier specifications are shown in Table 4.

The difference between the angular position obtained from the encoder and the setpoint is used as the contoller's input. We use PID controllers to point each gun to the desired target. The PID output of motors 1 and 2 can be found in the dSPACE sub-D connector P1A-31 and P1B-31

Table 1: Components of the LM555 timers.

Target	$R_A(k\Omega)$	$R_B(k\Omega)$	$C(\mu F)$	$Period(sec.)$	$DutyCycle(\%)$
1	3.3	3.3	470	3.22	33.33
2	10	6.8	470	7.69	28.31
3	3.3	3.3	470	3.22	33.33
4	22	6.8	470	11.60	19.10
5	15	15	470	14.66	33.33
6	15	15	470	14.66	33.33
7	15	15	470	14.66	33.33
8	3.3	22	470	15.41	46.51

Table 2: Motor specifications.

Rated Power (W)	40
Rated Speed (rpm)	3000
Rated Voltage (Vdc)	75
Instantaneous Max. Torque (N x cm)	38.2
Max. Speed (rpm)	5000
Weight (Kgf)	0.6

Table 3: Encoder specifications.

Power Supply Voltage (Vdc)	5
Consumption Current (mA)	100
Resolution (pulse/rotation)	500
Operating Temperature (^{o}C)	0-40

Table 4: Advanced Motion Control amplifier specifications.

Power Supply Voltage (Vdc)	20-60
Peak Current (A, 2 sec. max.)	$\pm$ 12
Max. continuous current (A)	$\pm$ 6
Power Dissipation (W)	10
Size (inches)	$5.09 \times 2.98 \times 0.99$
Weight (oz.)	10

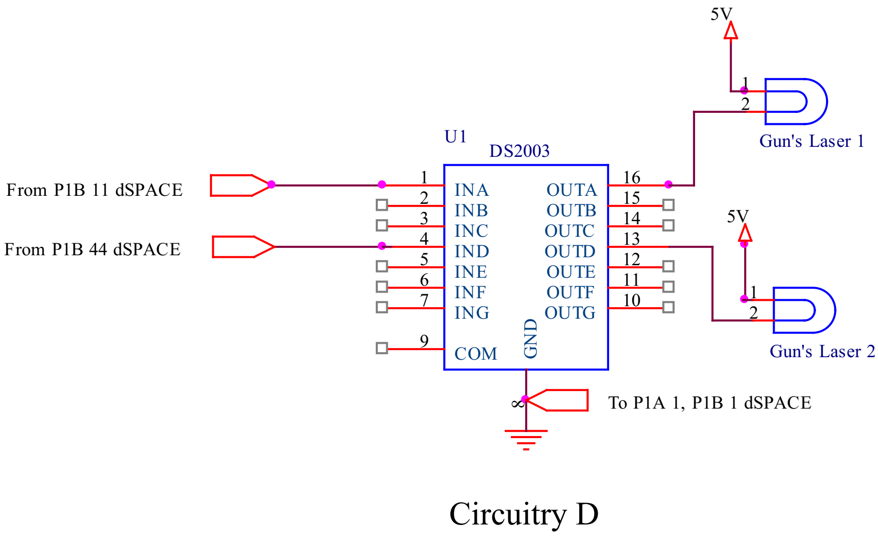


Figure 5: Connection between dSPACE and gun laser drivers.

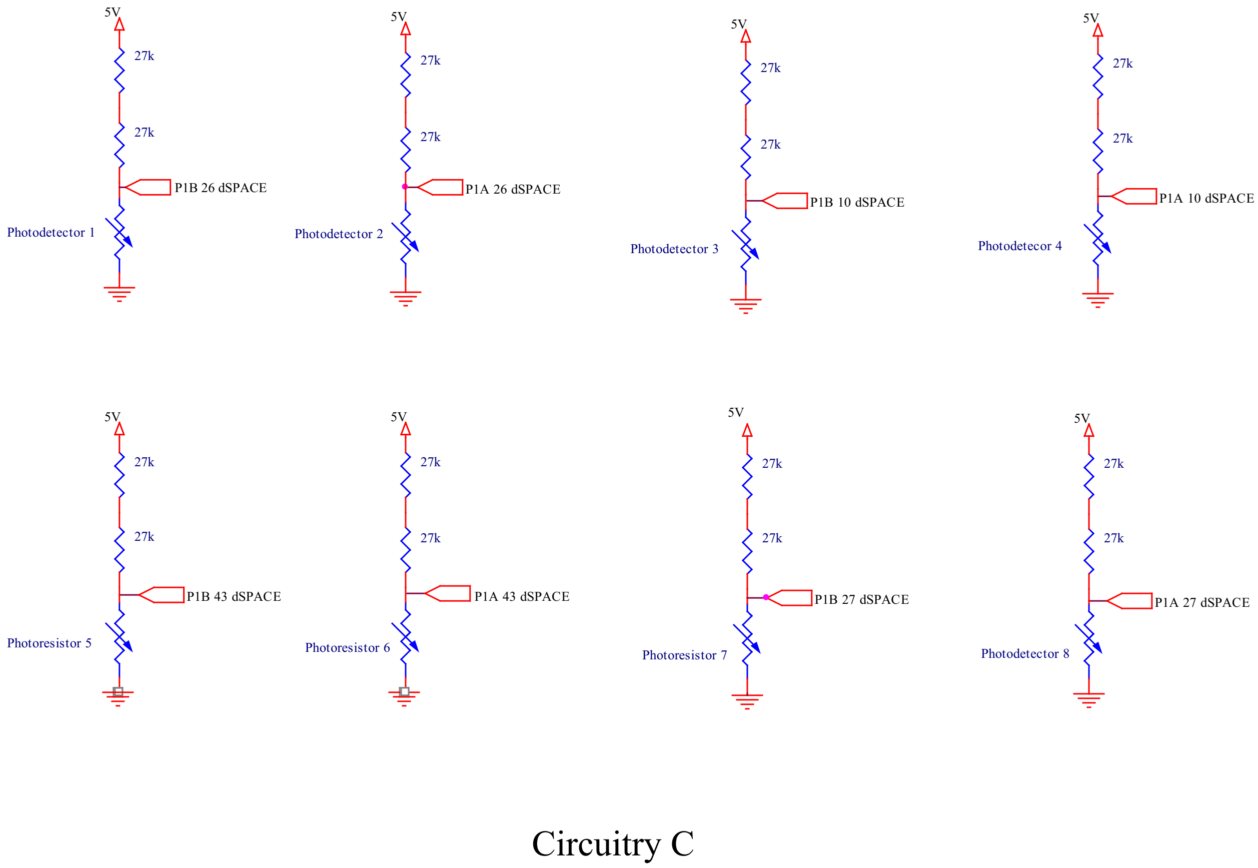


Figure 6: Connection between dSPACE digital input and photodetector circuit.

respectively. These two analog outputs are connected to pins 4 and 5 of the connector P1 of the Advanced Motion Control amplifiers whose outputs are connected to the coils of the motors 1 and

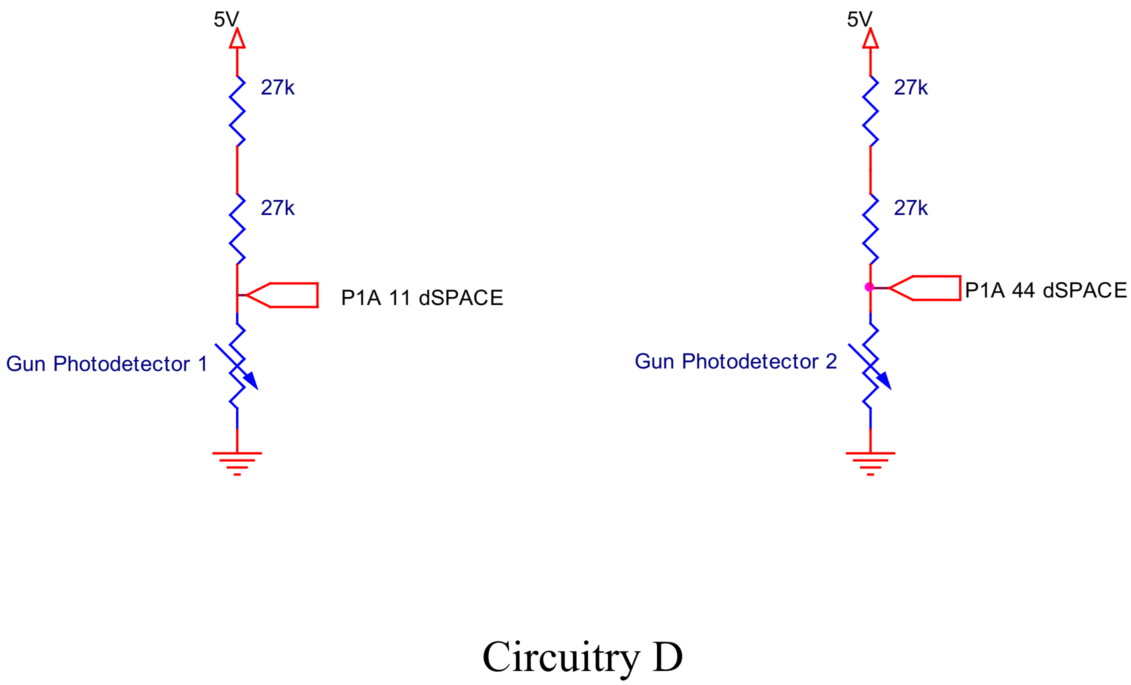


Figure 7: Connection between dSPACE and gun photodetectors.

2. Figures 8 and 9 shows all the connections among the encoder, motor, and dSPACE hardware for guns 1 and 2 respectively.

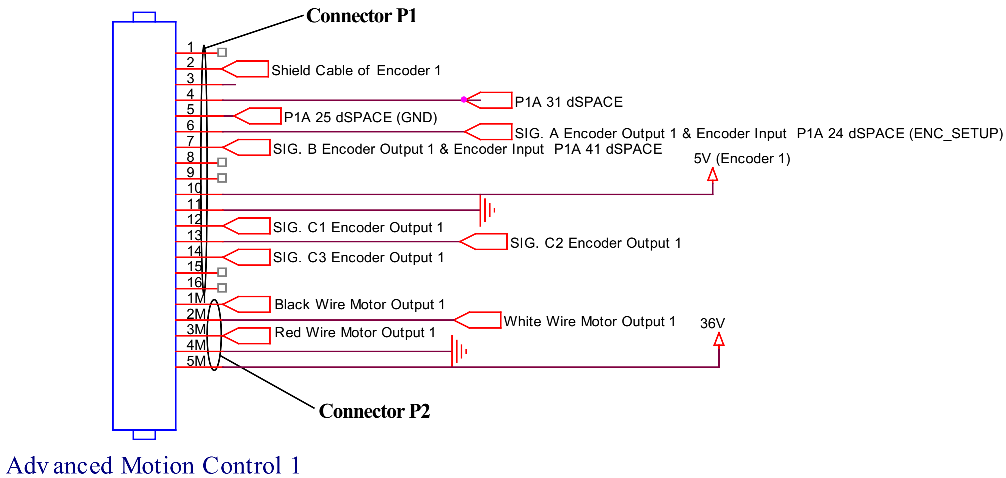


Figure 8: Connection among dSPACE analog outputs, Advanced Motion control amplifier 1, and motor 1 inputs.

3.7 Power Supply

Two power supplies are used in the experiment. The first one is manufactured by Twinfly and it supplies a 12 DC voltage at a maximum of 8.5A. This voltage is down-converted to 5V by means of

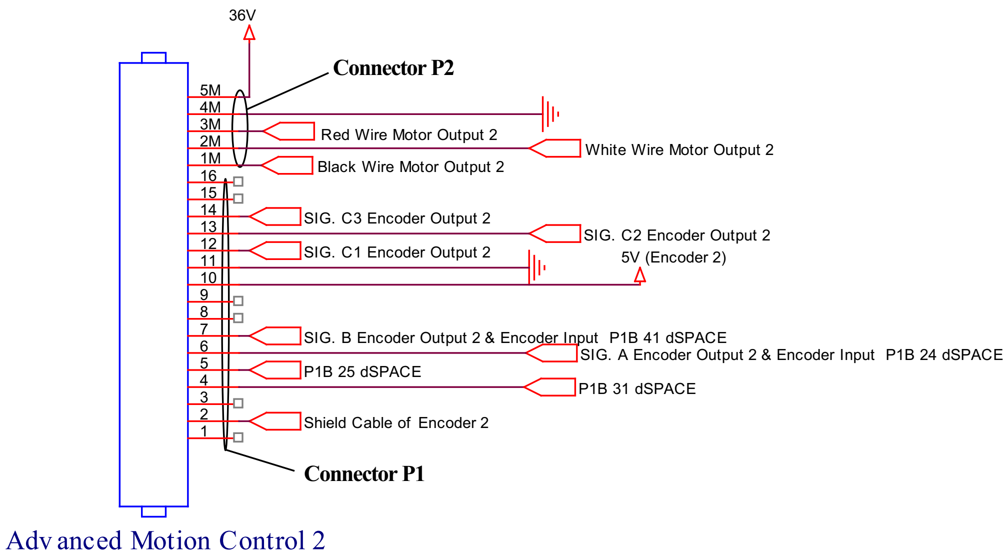


Figure 9: Connection among dSPACE analog outputs, Advanced Motion control amplifier 2, and motor 2 inputs.

the voltage regulator LM7805. These 5 volts are used to power up all the timers that drive both the target and gun lasers and all photodetectors. The other power supply is manufactured by Sorensen (DCR-40-25B) which supplies 36 DC volts to the two Advanced Motion Control amplifiers.

3.8 Travel Times Between Targets

Tables 5 and 6 show the travel times between targets for gun 1 and 2 respectively. These values are useful in the decision-making process when the control strategies to be implemented consider the minimization of the travel time of the guns between targets. Moreover, note that, in general, the travel times between targets i and j are slightly different from the ones from targets j and i, and compare the travel times for the same targets for guns 1 and 2.

Table 5: Electromechanical arcade: travel times (seconds) between targets for gun 1.

	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6	Target 7	Target 8
Target 1	0	1.801	3.359	4.83	6.207	7.469	8.744	9.99
Target 2	1.81	0	1.506	2.975	4.356	5.616	6.890	8.135
Target 3	3.374	1.513	0	1.414	2.794	4.054	5.328	6.572
Target 4	4.832	2.971	1.412	0	1.335	2.596	3.867	5.115
Target 5	6.211	4.350	2.791	1.335	0	1.213	2.491	3.736
Target 6	7.478	5.618	4.056	2.6	1.227	0	1.224	2.470
Target 7	8.747	6.889	5.328	3.873	2.494	1.218	0	1.199
Target 8	10.003	8.142	6.581	5.127	3.748	2.472	1.212	0

Table 6: Electromechanical arcade: travel times (seconds) between targets for gun 2.

	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6	Target 7	Target 8
Target 1	0	1.443	2.922	4.488	6.069	7.798	9.706	11.937
Target 2	1.443	0	1.421	2.984	4.571	6.305	8.207	10.442
Target 3	2.919	1.426	0	1.512	3.096	4.824	6.731	8.963
Target 4	4.484	2.991	1.514	0	1.529	3.257	5.167	7.4
Target 5	6.065	4.573	3.093	1.529	0	1.679	3.587	5.82
Target 6	7.794	6.301	4.824	3.258	1.675	0	1.857	4.088
Target 7	9.714	8.22	6.744	5.178	3.593	1.866	0	2.169
Target 8	11.949	10.449	8.976	7.41	5.822	4.095	2.168	0

3.9 Simulink Blocks

Here we explain the main Simulink blocks that you will need to initialize the experiment, to interface to all photodetectors, to shoot the gun lasers once a target has been detected, and to control the guns' angular positions. There are five main blocks in the programs as shown in Figure 10. The "memory allocation" block contains all the global variables defined in the program.

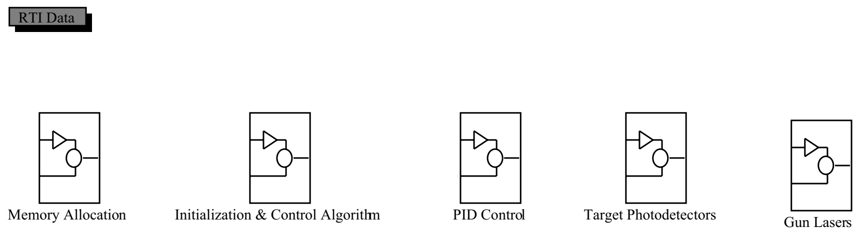


Figure 10: Main blocks in the experiment.

The "initialization and control algorithm" block has the initialization of the gun angular positions and the main program. The "PID control" block sets the reference inputs for the motor angular positions, reads the actual angular position of the motors, generates the controller output, and determines when the guns reach the desired position. The "target photodetectors" block scans the digital inputs of the target photodetectors. The "gun lasers" block turns the lasers on and off. Next, we describe the key blocks in more detail.

3.9.1 Initialization and Control Algorithm

Figure 11 shows the blocks contained in the initialization and control algorithm. The "Mode" variable has to be defined in the dSPACE GUI so it can take 0 or 1 values. If its value is zero, then it will enter the "initialization" block; otherwise, it will enter to the "regular" block.

The initialization block is used for setting the angle references for both guns. Figure 12 shows the initialization of the gun angles. This turns both gun lasers on and assigns 1 to the variables "t1" and "t2," which indicates that both guns start at an angular position equal to 120.5° and 120° respectively (more details on this will be explained below). These angular positions serve as

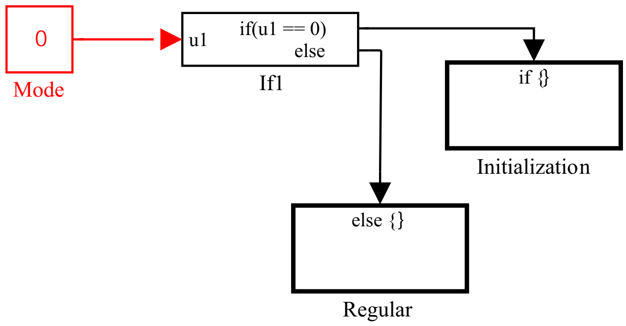


Figure 11: Blocks contained in the initialization and control algorithm.

references for future desired angular positions so that you need to point both gun lasers at the photodetector of target 1 at this time. Figure 13 shows the logic to check whether it is time to

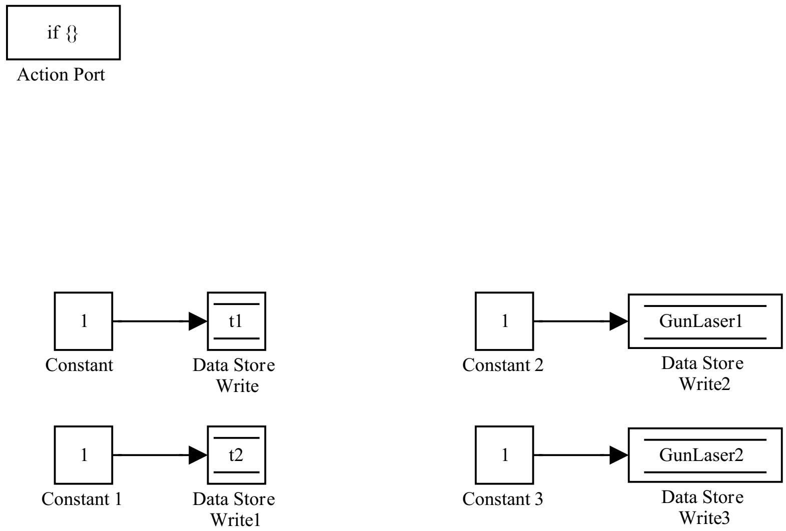


Figure 12: Initialization of the experiment.

make a new decision. If so, then the variable “DecisionFlag1” is equal to 1 and the block “Choose t1” will select the new target to process next. You will implement your own scheduling algorithms in this block, where the results of your solutions will dynamically change the values of the variables t1 and t2 to point the guns at the desired targets. Figure 14 shows one option of how the guns can choose different targets at every decision time; however, the students still need to implement their own strategies in all the blocks “Apply Policy for Gun 1 if T2=X.” Here, the basic idea is that gun 1 checks first which target gun 2 is pursuing or processing. Then gun 1 chooses any target except the one already being pursued or processed by gun 2. Once a gun has chosen a new target to process next, it will wait for the appearance of the target that is pointing at. If so, it will turn its laser on; otherwise, it will wait for the target appearance. Once the gun shoots at the target, the program checks if the target was shot successfully and if that is the case, the gun will gain a point and it will move to another target's location (more on this will be explained below).

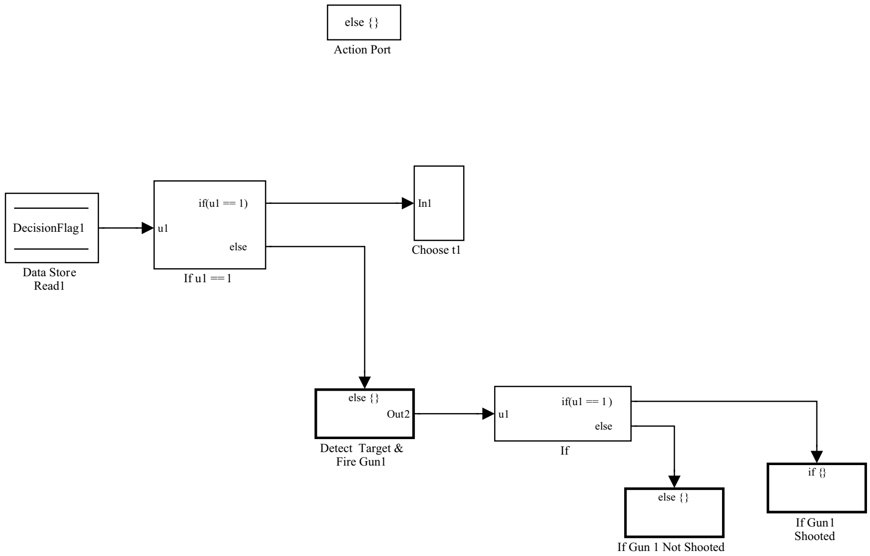


Figure 13: Regular block of the experiment.

3.9.2 PID Control

The blocks contained in this part will be used for controlling the gun angular position. Figure 15 shows the angles to which the guns need to point at in order to detect any of the eight targets. The values contained in variables t1 and t2 will determine the angular position to which gun1 and gun2 will point at. This is the current setpoint that is passed to the block "MotorXCon," which is used by the PID controller shown in Figure 16 as the variable "Angle1." A rate limiter is used to slow down the speed of the motor during setpoint changes to prevent any slippage of the guns on the motor shafts and to allow for easy viewing of the operation of experiment. Each of these motors has a quadrature encoder that provides the current angular position of the motor. The actual angle and the setpoint values are used to compute the angular position error, and the PID controller uses this value to control the angular position of the guns. When the absolute value of the error is less or equal to 0.6 the flag "enableFlag1" is set to 1, which means that the gun is already pointing at the desired target. Note that if t1=3 in the block "PID Control," then gun1 will set the reference of the target 3 angular position to 79.5 (see Figure 15). This reference will be passed to the blocks defined in Figure 16 and the PID controller will point this gun at that reference angle sometime later. Furthermore, there is logic that will indicate when the gun has reached the target position by setting the flag "enableFlag1/enableFlag2" to 1 for motor1/motor2.

3.9.3 Detection and Shooting at Targets

Once a gun has chosen a target to process next, the variable "DecisionFlag1" is reset and the algorithm enters the block "Detect Target & Fire Gun1" (see Figure 13). In this particular case, the gun will wait for the appearance of the chosen target. Figure 17 shows how a gun detects a target appearance if it is pointed directly at it via a photodetector mounted on the gun. If the signal on the variable "enableFlag1" is greater than or equal to 0.5, the block propagates the signal read by the gun photodetector "GunDetector 1;" otherwise, it propagates the third input (turns the laser off in the block "DisableGun1"). Notice that the gun photodetector only looks for an appearance when the gun has reached the desired angular position (i.e., variable "enableFlag1" has been set to 1 in the PID control block). Once the target appearance has occurred, the gun laser is automatically shot. Figure 18 shows how this is done in dSPACE using the variables "GunLaser1" and "GunLaser2" (these variables are contained in the block "gun lasers").

3.9.4 Shot Target

When a gun shoots at a target, the corresponding target photodetector must be checked to see whether it was successfully shot. Figure 19 shows the logic to determine how to do that. First, this routine finds out which target the gun is pointing at (block switch case 1). Second, it checks if the corresponding photodetector was triggered (i.e., block ifTargetx). Figure 20 shows how the target photodetector signals are associated to digital inputs in dSPACE. Figure 21 shows the logic of how to verify whether a target (target 1 in this case) was successfully shot respectively. If TargetSensor 1 is equal to zero, then the student can execute the desired action in the block "If Action Subsystem;" otherwise, the algorithm keeps looking at TargetSensor 1 until it becomes equal to zero. Finally, when the target is successfully shot, the block ifTargetX will output 1 and the block "Set DecFlag 1" (Figure 22) will be executed, which increments the gun points by 1, and the gun will then be ready to make a new decision by setting the variable "DecisionFlag1."

3.10 Steps to Start and Stop the Experiment

This section describes the steps necessary to either start or stop the experiment in the proper manner. They must be followed to avoid any damage caused to the guns.

3.10.1 How to Start the Experiment

Here we provide a recipe about how to sequentially execute several steps in order to guarantee the proper functioning of the electromechanical arcade. Recall that the moment in which you attach the guns to the shaft of the motor is very critical so these steps make sure that this takes place at the right time.

★ **Warning:** Do not ever look directly into the lasers as this can damage your eyes.

1. **Guns Unmounted from Motor Shafts:** At this time both guns must be unmounted from the motor shafts.

2. **Reset Variable Mode in dSPACE GUI:** Make sure that this variable is equal to zero before running the experiment. See Section 3.9.1 for details.

3. **Run the Experiment:** Run the experiment using the dSPACE GUI.

4. **Power Supplies ON:** Before connecting the Sorensen power supply to the experiment, make sure its output voltage is set to 36 volts. Having checked that you can turn this power supply on. On the other hand, connect the cord of the Twinfly power supply to the AC outlet. The shaft of the motors must be fixed at this time.

5. **Mount Guns:** Proceed to mount both guns on their respective motor shafts. You must align each laser beams to the photodetector of target 1. Once the laser beams are hitting the

photodector of target 1, proceed to fix the guns to the motor shafts by means of the setscrews located in the fabric composite material.

6. **Run your control strategy:** At this moment you are ready to test your scheduling strategy. You can do this by setting the variable Mode to 1 in the dSPACE GUI.

   ★ **Warning:** If you want to stop the experiment, you must follow the steps below.

3.10.2 How to Stop the Experiment

These are the steps that must be taken to stop the experiment. These steps avoid the potential damage that could be caused to the gun wires if these are not followed in the right order. These steps must be followed when the program needs to be stopped when it is already running or when you need to recompile your Simulink file.

1. **Reset Variable Mode in dSPACE GUI:** Set this variable to zero. The guns will point at target 1.
2. **Unmount Guns:** Proceed to take the guns off the motor shafts.
3. **Experiment Ready:** Now you can do what you need to do with the software. You can recompile your file or stop the experiment in the dSPACE GUI.

★ To start the experiment again follow the steps in the above subsection.

Acknowledgements: We would like to thank William Thalgott and Emmet Evans for helping with some construction details for the electromechanical arcade. The arcade was originally constructed by a group of undergraduates in an EE 682P design project and it was later improved by Kristina Guspan and Dawn M. Kin.

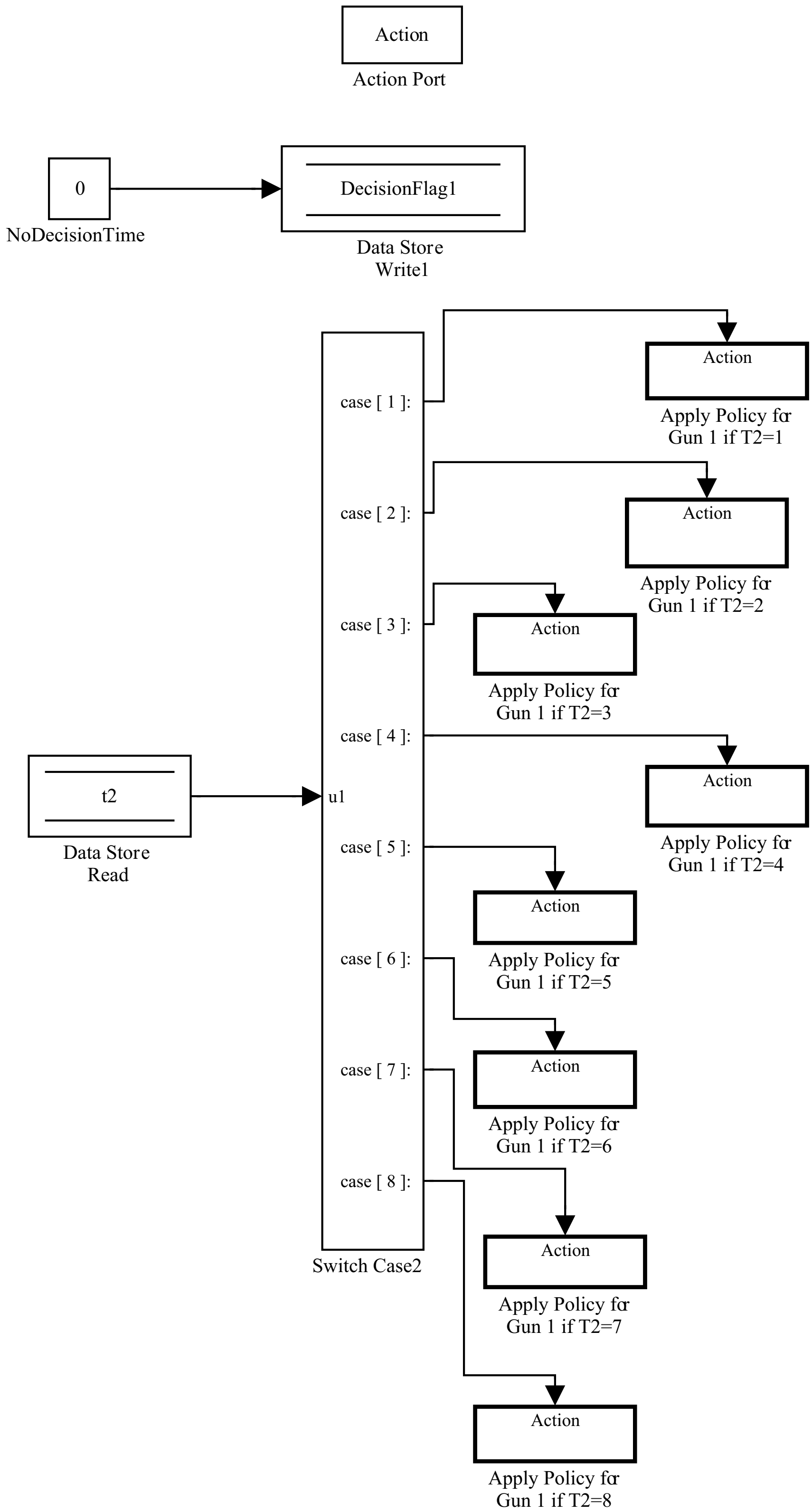


Figure 14: Scheduling block of the experiment.

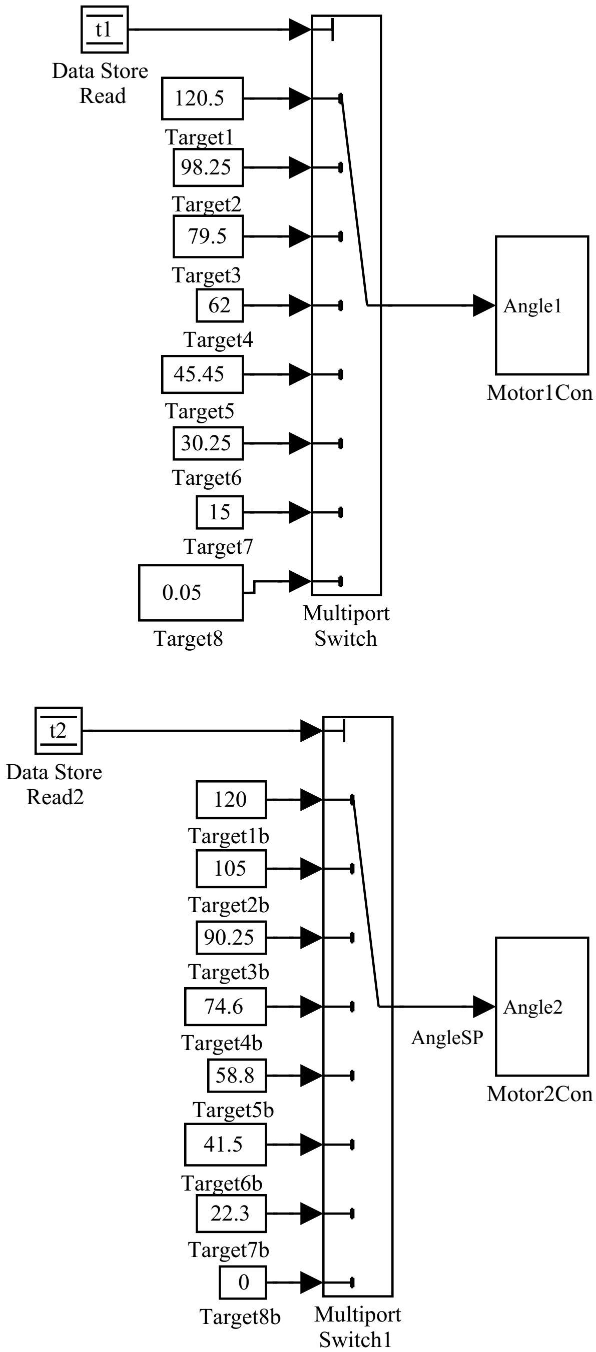


Figure 15: Set-point of guns.

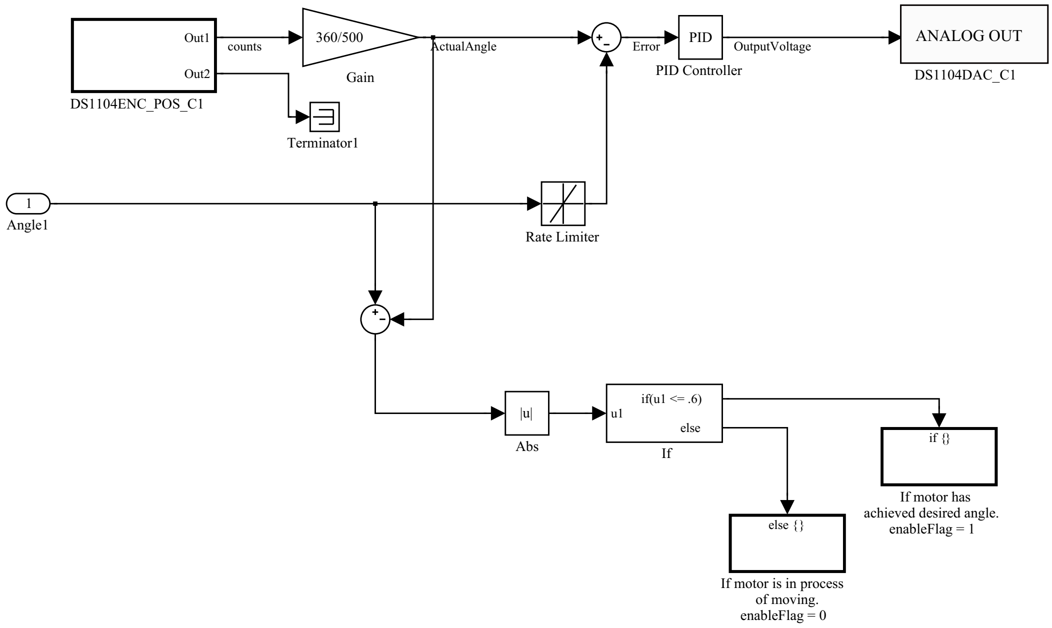
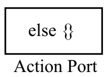


Figure 16: PID control block of the experiment.



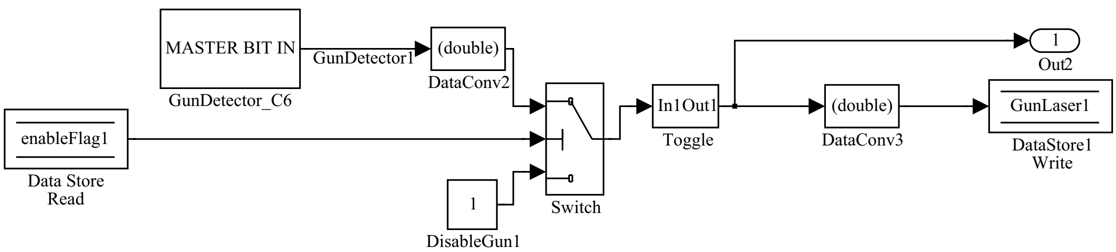


Figure 17: Detection of target appearance and shooting of laser.

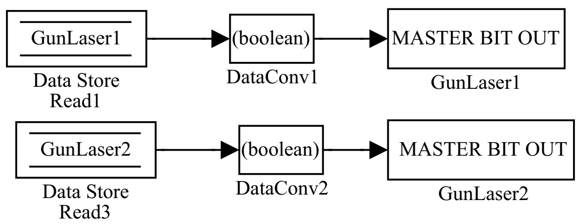


Figure 18: Shooting of guns.

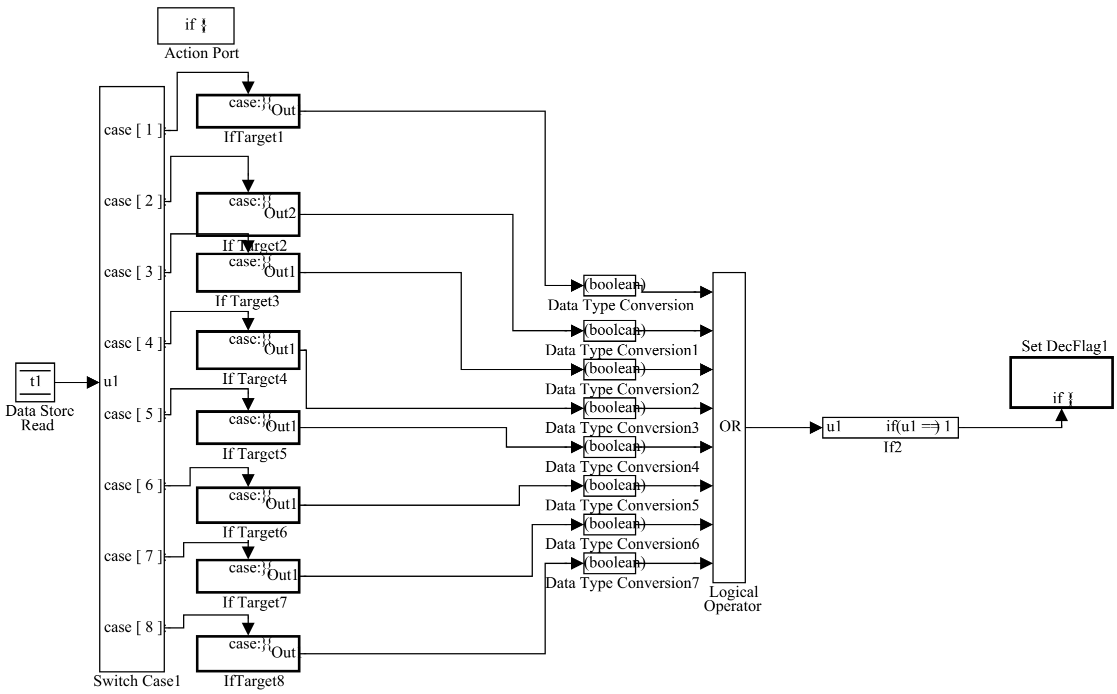


Figure 19: Determination of the target that was shot .

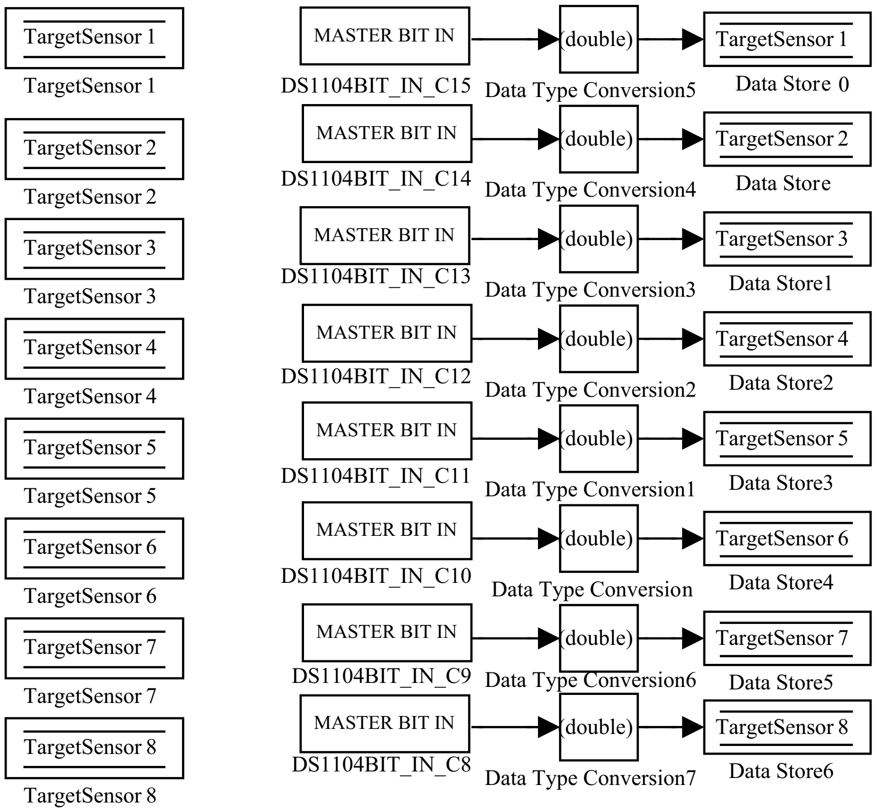


Figure 20: Target photodetectors connected to dSPACE.

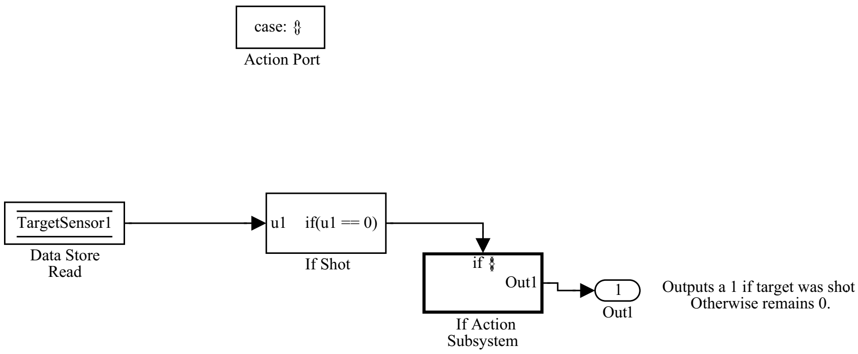


Figure 21: Logic for verifying a target shot.

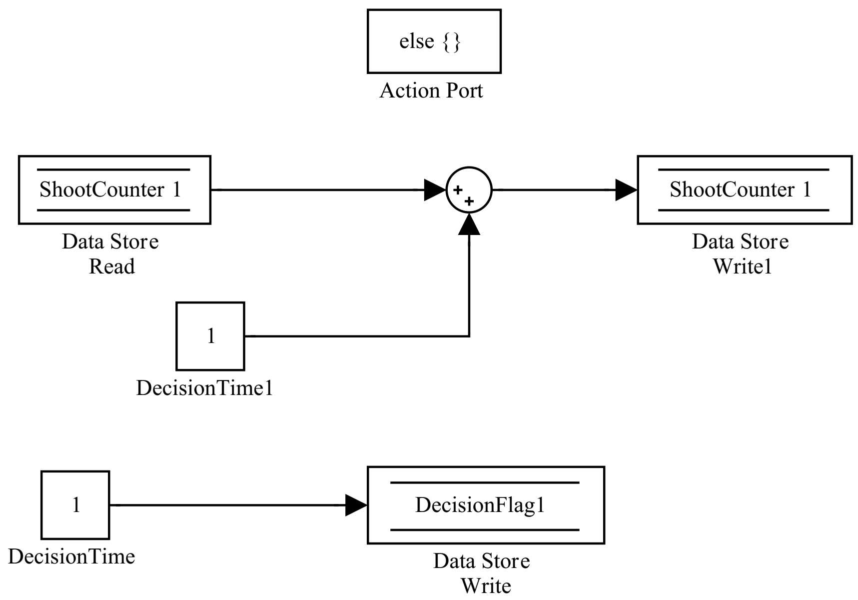


Figure 22: Point obtained by the guns.