Visual Object Tracking and Target System

Checklist of Deliverables

Jaime Lien
Willie Sanchez

At its completion, our system will be able to do the following:

- Feed sync signals into a television camera and receive from it a composite video signal.
- Process a composite video signal, using color comparison to locate the position of an object.
- Determine a moving object's velocity.
- Use an object's position and velocity to predict its future position.
- Aim a projectile launcher at an object's predicted future position such that, when fired, the projectile will hit the object.
- Follow an object's movement with a laser beam.

Video Track and Shoot System

Jaime Lien
Willie Sanchez

Introductory Digital Systems Laboratory

TA: Jia Fu Cen

May 12, 2004

Abstract

This system is designed to track and shoot a person or object moving in two dimensions. An NTSC camera is positioned to capture an area in front of a black backdrop. A laser pointer and a toy gun are each mounted on two stepper motors to allow rotation along two axes. The laser follows the target's motion; the gun aims at the target's predicted future position to account for the projectile flight time of its using its current position and speed.

The system is comprised of two digital subsystems: a video interface and a mechanical interface. In the former, analog video input from the camera is converted to digital and processed to determine the target's position coordinates and speed. These coordinates and speed values are then fed into the mechanical interface, which uses them to perform a linear estimation of the target's future position. The mechanical interface then drives the stepper motors to aim the laser pointer at the target's current position and the gun at its future position.

Introduction

This system is designed to track and shoot a person or object moving in two dimensions. An NTSC camera is positioned to capture an area in front of a black backdrop. A laser pointer and a toy gun are each mounted on two stepper motors to allow rotation along two axes. The laser follows the target's motion; the gun aims at the target's predicted future position to account for the projectile flight time of its using its current position and speed.

The system is comprised of two digital subsystems: a video interface and a mechanical interface. Both are run on a 10-MHz clock. In the former, analog video input from the camera is converted to digital and sampled at a resolution of 80 by 60. At each sample point, the digital luminance value is compared to a threshold luminance value. The coordinates of points corresponding to whiteness are averaged at the end of each frame to determine the center of the target. Center coordinates from two consecutive frames are then used to calculate the person or object's speed.

These coordinates and speed values are then fed into the mechanical interface, which uses them to perform a linear estimation of the target's future position. The mechanical interface then drives the stepper motors to aim the laser pointer at the target's current position and the gun at its future position.

Video Subsystem
Jaime Lien

Overview

The video subsystem is responsible for receiving the analog composite video signal from the NTSC camera and producing the target's position coordinates and speed. This system runs on a 10-MHz clock. The active video signal is sampled and converted to digital at a resolution of 80 by 60. At each of the 4800 sample points, the digital luminance value is compared to a threshold luminance value. The coordinates of points corresponding to whiteness are averaged at the end of each frame to determine the center of the target. Center coordinates from two consecutive frames are then used to calculate the person or object's speed. The subsystem can be reset by the user via a push switch. During a reset, the system outputs position and speed values of zero.

The block diagram for the video subsystem is shown in Figure 1.

Hardware

The composite video signal is fed into the following three hardware chips.

GS4981

The GS4981, a video sync separator, is used to extract the vertical sync signal from the composite video. This signal pulses low at the end of each frame.

LM311

The LM311 is a comparator that outputs a digital high or low if an input analog signal is above or below an input threshold voltage. Video horizontal and vertical retraces are hidden by surrounding the sync signals with a blanking period of pure black. Hence, a blanking signal signifies the beginning and end of each line of active video. Setting the LM311 threshold to 2.2 volts extracts the blanking signal from the composite video.

AD775

This analog-to-digital converter samples the active video at 5 MHz in order to obtain digital luminance values. The reference ladder is set from 2.5 to 5 volts, biasing the converter to the active video voltage range. Because the luminance threshold is set in the middle of the active video, only the most significant output bit is used.

Module Descriptions

GS_LM

The GS_LM module uses the synchronized vertical sync and blanking signals to determine the beginning of each frame and line. The output framestart and linestart signals pulse high for one clock cycle at their respective events.

FSM and counters

The FSM coordinates the sampling of the active video; the luminance comparison; and the operation of the position, center, and speed calculators. Its state transition diagram is shown in Figure 2.

The sampling depends on inputs from two counters. The vertical counter sends a venable signal that allows the FSM to use every fourth line of video, or 60 lines per frame. Venable is also used to reset the horizontal counter, which sends a high henable signal to the FSM after every six cycles of a 10-MHz clock, or 80 times per line. Together, venable and henable produce a resolution of 80 by 60.

The henable signal causes the FSM to accept the luminance value from the A/D converter. If the luminance for a given sample is determined to be white (i.e. the AD775's most significant output bit is 1), the FSM allows the center calculator to accumulate the position coordinates corresponding to the sample. It then increments the position calculator and waits for the next henable.

After the last sample of the line, the position calculator returns an xdone signal to the FSM, causing it to wait for the next venable before once again accepting henable. Finally, after the last sample of the frame, the position calculator sends a ydone signal that makes the FSM reset the vcounter and wait for the next frame.

Position calculator

The position calculator keeps track of the coordinates corresponding to each video sample. Starting at (x=0, y=59), the position calculator increments x at each xinc signal from the FSM and decrements y at each yinc. The end of each line, (79, y), causes the position calculator to output an xdone signal. Similarly, at the end of the frame, (79, 0), the position calculator outputs a ydone signal.

Center calculator

The center calculator is an accumulator that takes in position coordinates corresponding to white video samples. It includes a counter that keeps track of the number of points accumulated. At the ydone signal, the accumulated sum is divided by the number of points used to determine an average of all the x-coordinates and y-coordinates accepted, i.e. the center of the target. Xcenter and ycenter are passed to the speed calculator and output to the mechanical interface.

Speed calculator

At each ydone signal, the speed calculator calculates the positional difference between the present center and the past center and divides this difference by the frame rate of 60 Hz. The speed is scaled by a factor of 1/10 to decrease the number of output bits needed. The final result is output to the mechanical interface.

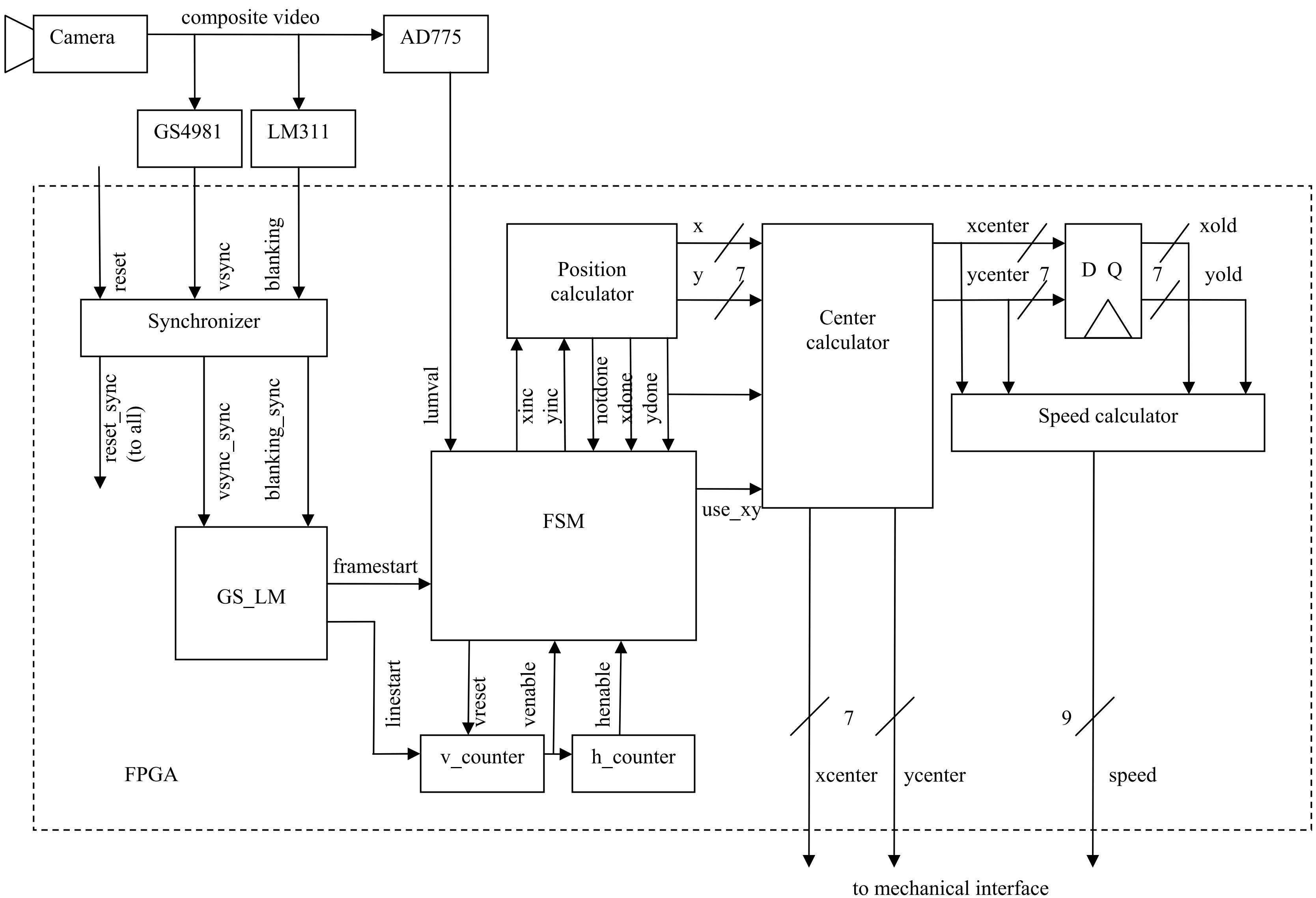


Figure 1: Block diagram of video subsystem

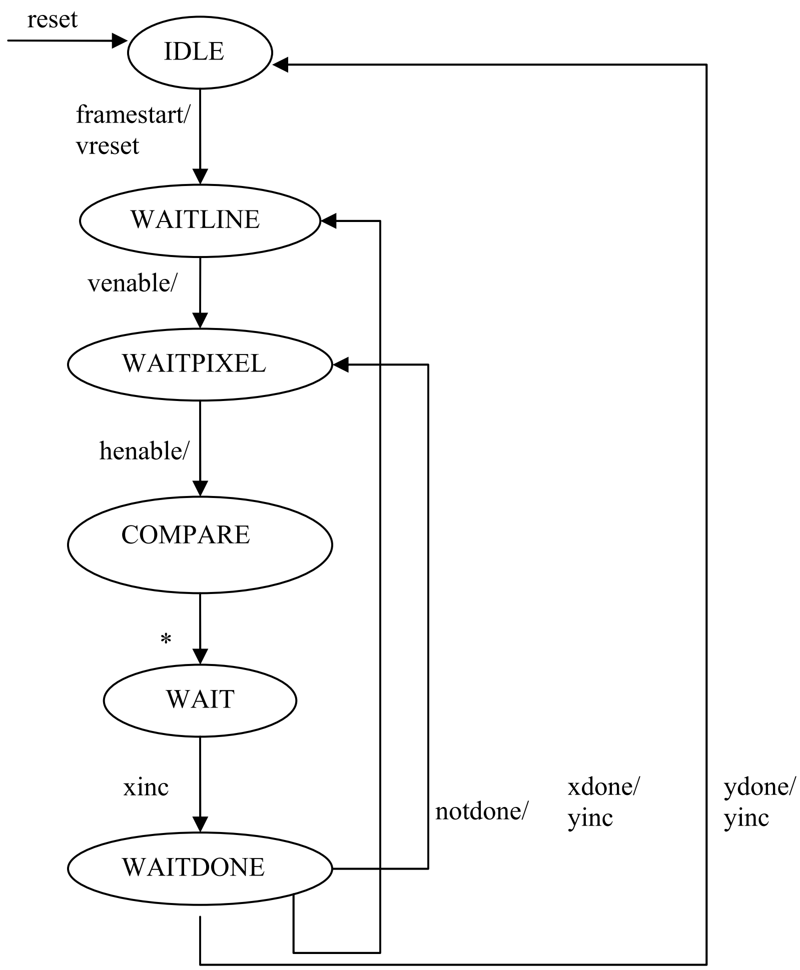


Figure 2: FSM state transition diagram

Testing

Testing the video interface occurred in four major stages. The preliminary stage involved the Verilog code was tested using ModelSim testbenches and Max-PlusII simulation. Please see Appendix for testbench waveforms. Then the system was incrementally implemented in hardware and tested via the oscilloscope and logic analyzer. First, GS_LM ouputs were compared to the composite video signal to ensure the functionality of framestart and linestart. Next, the position calculations were tested by removing the center and speed calculators from the top module. Finally, the full system was implemented and tested.

The most important result of testing was the addition of the LM311 chip. The system originally used a horizontal sync signal from the GS4981 instead of a blanking signal to determine linestart. However, GS_LM testing revealed timing issues with the GS4981's horizontal sync output, prompting a switch from horizontal sync to the more reliable blanking signal.

Mechanical interface
Willie Sanchez

Overview

The purpose of the mechanical system of this project is to interface with the real-world and track the motion of a target of specified colored as described in the video processing section. The physical system is constructed out of Lego parts onto which the stepper motors, laser, and missile projector are mounted. The mechanical subsystem consists of two divisions. The first division engages in real-time tracking of a moving target. The second division uses calculated estimated prediction values to aim the missile launcher at the targets future position n time instants ahead.

The movement of each division is controlled by two stepper motors corresponding to the x-y plane, giving the overall system a two dimensional capability. See Figure 3 for overall mechanical subsystem block diagram.

Module Descriptions

Clocking

Although the input clock of the mechanical subsystem is the same as the video processing subsystem, 10MHz, two clock dividers are used internally to bring down system clock frequency. The first divider brings the input 10 MHz signal to 300 Hz to allow ample time for accommodating for large number of motor steps, or impulses from target. The second divider established a 2Hz T sampling pulse that samples the position and speed vectors coming from the video processing subsystem. The reasons for these further discussed shortly are a matter of future work.

Follower

The follower is responsible for taking in 14-bits of registered position vectors and translating those to outputs corresponding to the number of steps for the respective x or y stepper motor to take, and the direction, either clockwise or counter-clockwise, in which these steps are to be taken. The follower does these calculations twice per second as it needs to wait for the 2 Hz enable signal coming out of the 2 Hz clock divider. These slow frequencies are chosen for functionality of the motor.

Predictor Calculator

This is a combinational block that continuously takes in the position and speed vectors and calculates to a first-order linear approximation the future position of the object n time instants ahead, where n is depends on the distance of the targets z-distance from the missile launcher. We assume zero acceleration, which allows our expression for calculation to be:

$$x[\Delta n] \approx x[n] + v_x[n]\,\Delta n$$
and
$$y[\Delta n] \approx y[n] + v_y[n]\,\Delta n$$

In our particular implementation, we chose n = 1 in order to save hardware and maintain the linearity of our model. By choosing n = 1, our expressions above just become addition operations.

Predictor

This module is identical to the follower module with the exception that it takes as inputs the outputs of the predictor calculator module. This controls the missile launcher and aims it one time unit ahead of the actual position.

Motor FSMs

There are 4 motor FSMs controlling the two motors in each of the divisions of the mechanical system. They are identical except for the x-y inputs, which depend on whether they are controlling the follower or the predictor divisions. The follower FSMs take in the outputs of the follower module, namely the step direction and the number of steps to be taken, and output a sequence of pulses to the LM18293 chip that directly drives the motors. See Figure 4. Similarly, the predictor FSMs take as inputs the outputs of the predictor module.

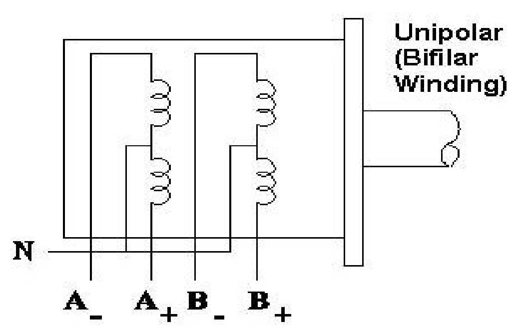
Each FSM has ten states. The stepping sequence of the motors is designed to consist of half-steps rather than full-steps. This design allowed for finer resolution per step of the motors, which in turn allowed higher accuracy on the tracking. After a reset state, the FSMs wait for a delayed version of the 2 Hz signal in order to sample the number steps and direction to be taken from the follower or predictor modules. The 2 Hz signal that goes to the FSMs is the 2 Hz signal from the clock divider module delayed by two clock cycles. These enable signals come from both the follower and predictor module stages and ensure that the signals indicating the number of steps to be taken and the direction in which these steps should go are stable when fed to the appropriate FSM.

At each sample time, the number of steps to be taken and the direction of these steps are placed into the appropriate *steps_left* and *step_dir* registers respectively. The *step_dir* signal determines whether the order in which the states are traversed is forward of backwards. This corresponds to clockwise or counter-clockwise motion of the motor. The *steps_left* value is decremented at each state transition. Once the value in the *steps_left* register is zero, the state in which this occurs is put into the *tmp_state* register and the FSM returns to the *WAIT_STEP* state for the next enable signal to arrive. In order, to maintain the correct pulse sequence fed to the motors so as not to lose track of the winding arrangements, at each sample time, the FSMs jump from the *WAIT_STEP* state to the state in the *tmp_state* register except after a reset which fixes the following state depending on the value of the *step_dir* signal. See Figure 5 for state transition diagram of FSMs.

Stepper Motor Control

The stepper motors in our system are the **Series 42M100B Stepper Motors.** These were chosen because of their 3.6º/step resolution. This means that if our target was at distance of z = 5ft., each at step would consist of .315ft. See Figure 6 in Appendix for more details.

The **Series 42M100B Stepper Motor** has unipolar (bifiliar winding) arrangement with ~75Ω/winding. See Appendix for motor documentation.



Stepper motor winding schematic

In order to drive the motors, LM18293 driver chips from National Semiconductor are used. Figure 4 shows the schematic for the interface between the driver and stepper motor and the pulse sequencing used to produce steps in the motor.

To power the motors, external power supplies should be used. Although it may be possible for the 6.111 kits to supply the necessary currents to drive the motors, this is not recommended as the motor back EMF is large may adversely affect other circuits on the kit. Initially after loading the motors with the Lego structures, driving the motors with a 12V supply did not provide the necessary torque to carry the load. To remedy this, we increased voltage to ~20V which then provided enough torque without causing harmful heat dissipation to the motors.

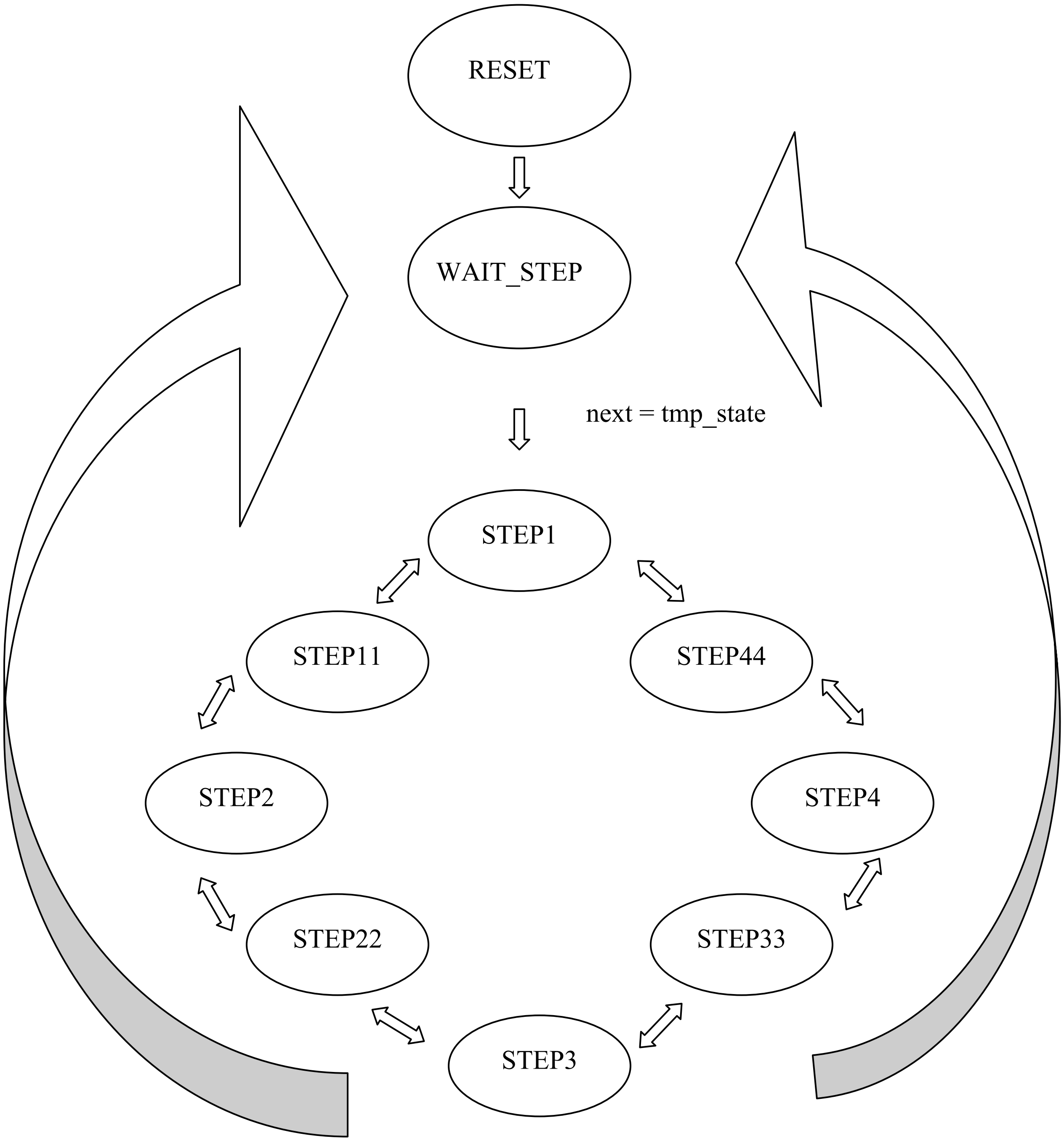
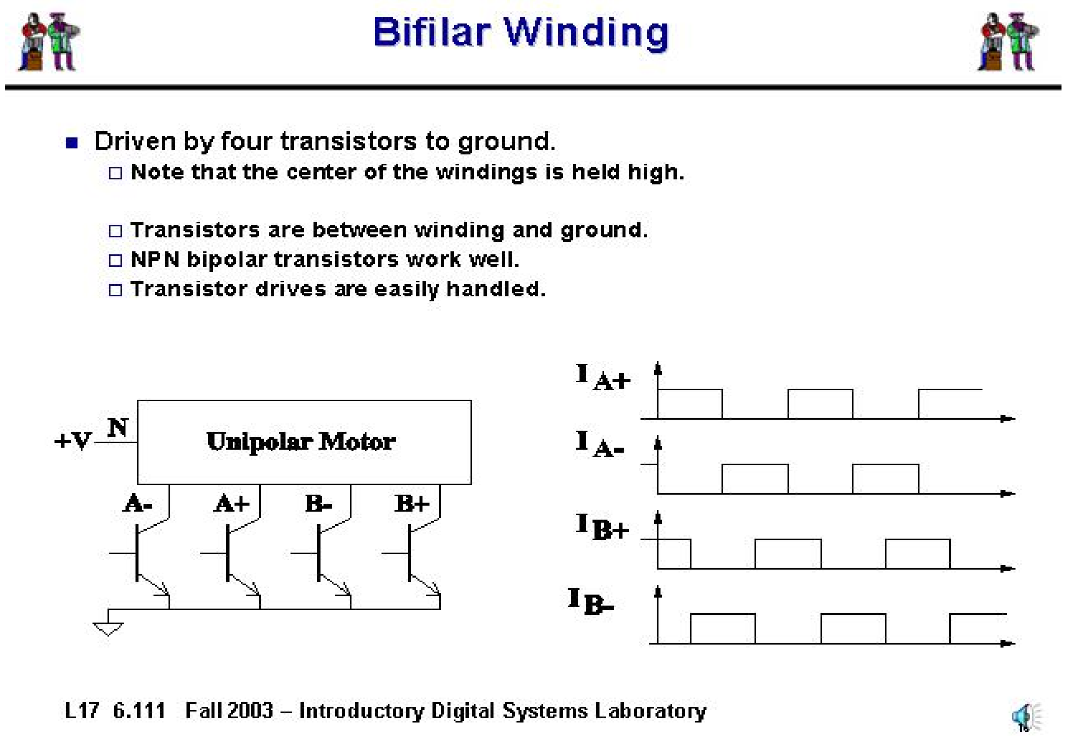


Figure 5: Motor FSMs state transition diagram



Testing and Debugging

As with any digital system, testing and debugging is a major step in the process. I ran all simulations on Max Plus II since it is a more reliable tool than ModelSim. The biggest issue I faced was bringing down the clock frequency to one where state transitions, corresponding to changing the pulse sequences driving the LM18293 chip, were slower. I began trying to produce steps at 10 MHz and quickly realized that this did not meet the rise and fall times of the LM18293, and therefore would not reliably drive the motor. I then proceeded to decrease my clock frequency to 2 MHz. This guaranteed I met the rise and fall times of the LM18293 chip; however, this still did not work. I proceeded to further reduce clock frequency and discovered that I could drive the motor anywhere beneath 500 Hz. I opted for 300 Hz trading off number of correctly taken steps versus stepping speed. Our system allowed for $2^{\wedge 7} = 128$ steps in the worst case difference between an old coordinate and a new sample (i.e. 128 – 0). Allowing this worst case difference meant that at 300 Hz, I could step sample new x/y values at ~2 Hz without interfering with the old x/y number of steps. The issue with this is that the direction of stepping may be changed, hence causing the incorrect number of steps to be taken in the wrong direction.

Once the issue of stepping speed versus highest number of steps allowed was resolved the next matter involved interfacing with the video processing system. I turns out that this was a bug that did not exist and resulted from malfunctioning equipment. We spent several hours trying to discern why the video processing system put out valid values at its terminals and be invalid at the inputs of the mechanical system. To debug this situation we tried isolating the motors and power supplies from the connecting wires to reduce magnetic noise effects and other phenomena. When this did not solve the problem we returned to more rigorous testing of each subsystem verifying proper state transitioning and corresponding signal values on the logic analyzers and oscilloscopes. After this we hypothesized that one of the analyzers must have been dysfunctional. This indeed resulted in being the case at which point we upon interfacing we were immediately functional, except for calibration.

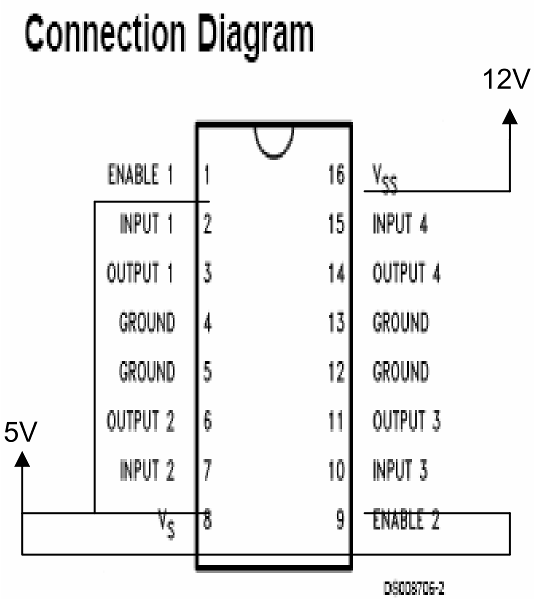
Conclusions and Future work

At the project deadline, the system was fully functional. However, due to mechanical issues, only the laser was implemented. In the future, we would like to use stronger motors. In future work of this project, I think it would be advisable to increase stepping speed while maintaining robustness to rapid jumps from the target. It has occurred to me that increasing the stepping speed and sampling times, while reducing the jumping threshold, may have also worked since faster sampling would only put a cap on the maximum speed the object could move at before the number of steps overflows.

This was an extremely exciting experience and fulfilling upon completion and functionality. We learned several things from this project. Most importantly, interfacing between digital systems is never a trivial task and when needing to meet a deadline where interfacing to systems is required, plan to devote a lot of time on interfacing.

Appendix

LM18293*
Four Channel Push-Pull Driver



Enable 1 activates outputs 1 & 2
Enable 2 activates outputs 3 & 4

TABLE 1. Input/Output Truth Table

V_E (**)	V_I (Each Channel)	V_O
H	H	H
H	L	L
L	H	X (*)
L	L	X (*)

(*) High output impedance.
(**) Relative to the pertinent channel.

* Courtesy of ***National Semiconductor***

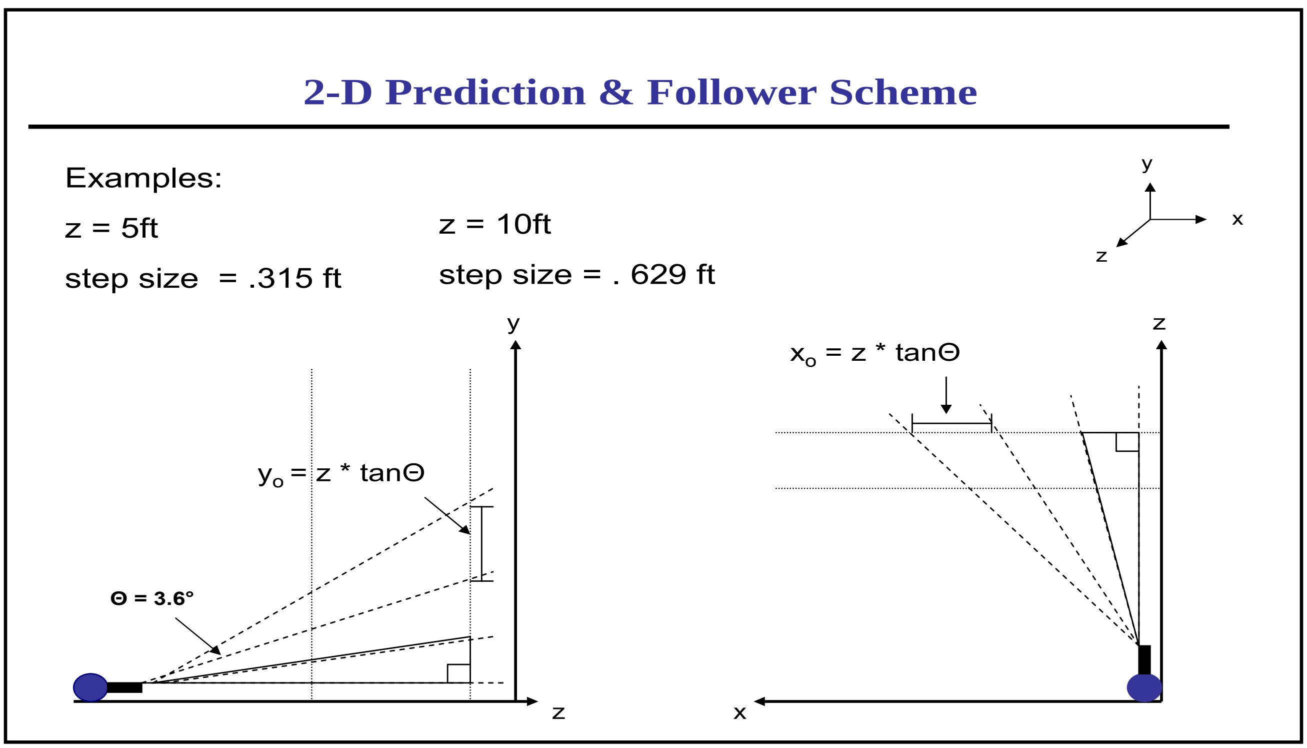
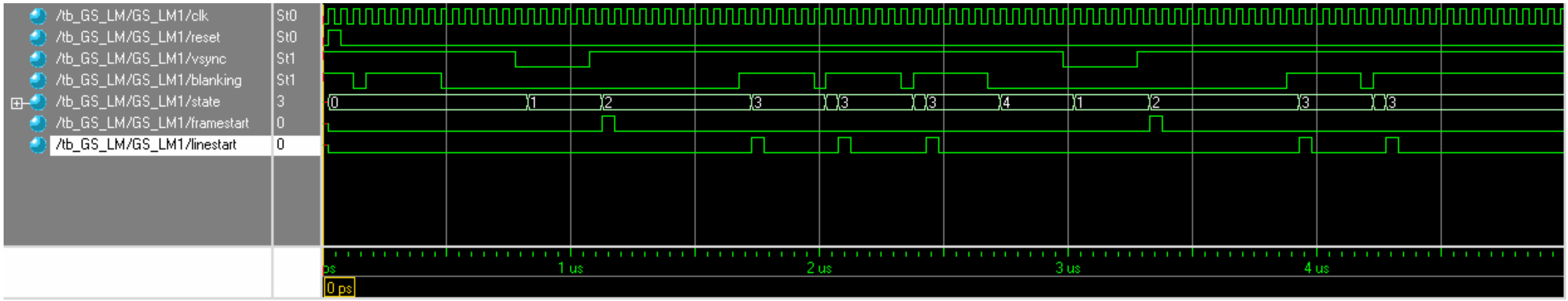
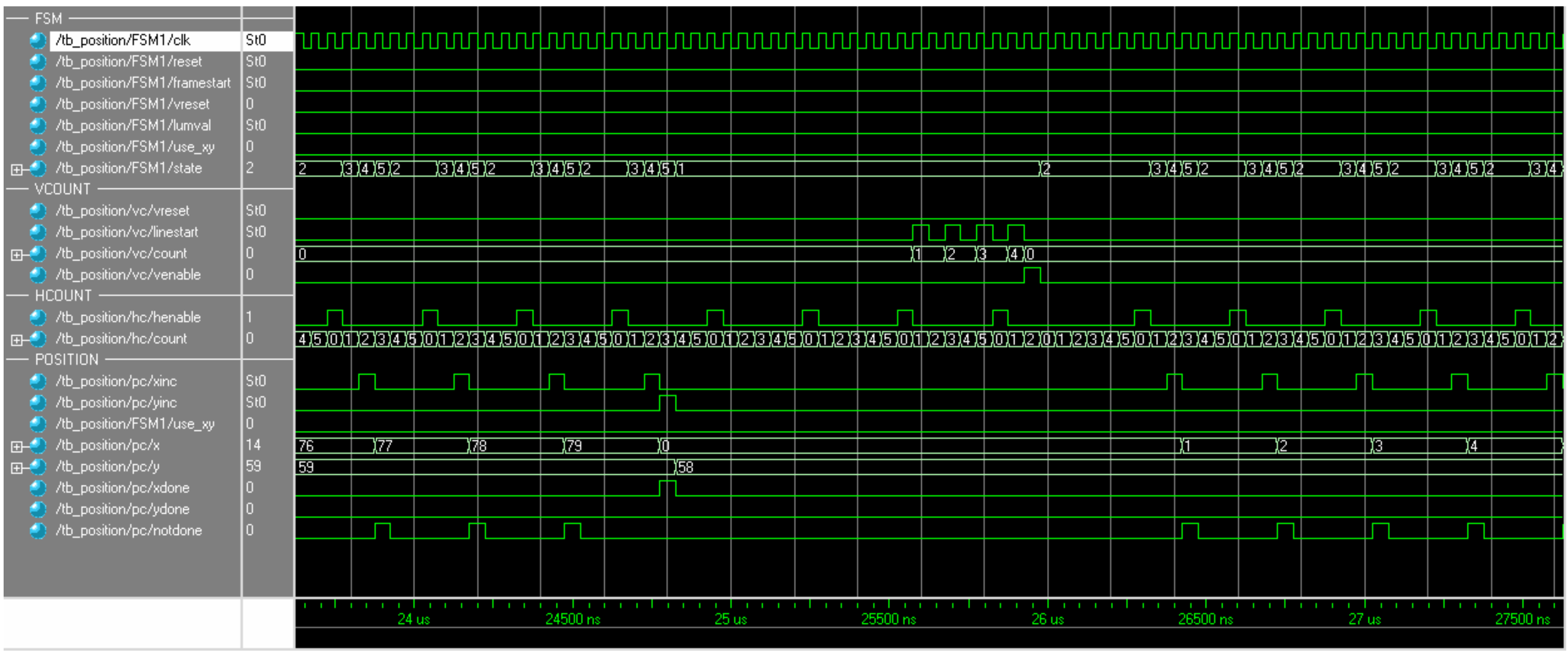


Figure 6: Physical setup of laser follower and/or missile launcher.



GS_LM testbench



Position testbench

Series 42M100B Stepper motors

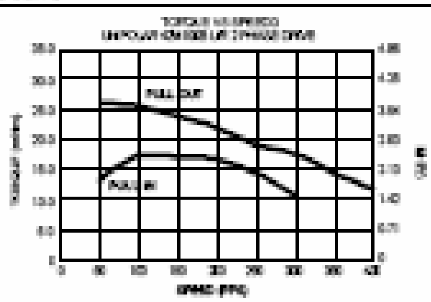
Features:

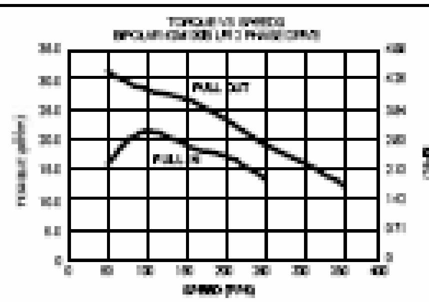
- 3.6° step angle
- Permanently lubricated sintered bearings
- Compact size

Motor data				
Motor order number	Unipolar		Bipolar	
	1738-42M100B1U	1738-42M100B2U	1738-42M100B1B	1738-42M100B2B
DC operating voltage	5	12	5	12
Resistance per winding Ω	12.5	75	12.5	75
Inductance per winding mH	6.6	37.7	11.3	62.1
Holding torque mNm *	45.2		49.4	
Detent torque mNm	5.0			
Rotor moment of inertia g x m²	12.5 x 10[illegible]			
Step angle	3.6° ± 0.25°			
Steps Per Revolution	100			
Insulation Resistance At 500Vdc	100 M Ω			
Leadwire Type	AWG #28			
Bearing Type	Bronze sleeve			
Operating Temperature	100°C, MAX			
Ambient Temperature Range	Operation: -20°C to 70°C Storage: -40°C to 85°C			
Dielectric Strength	650 ±50 Vrms, 60 Hz for 1 to 2 Seconds			

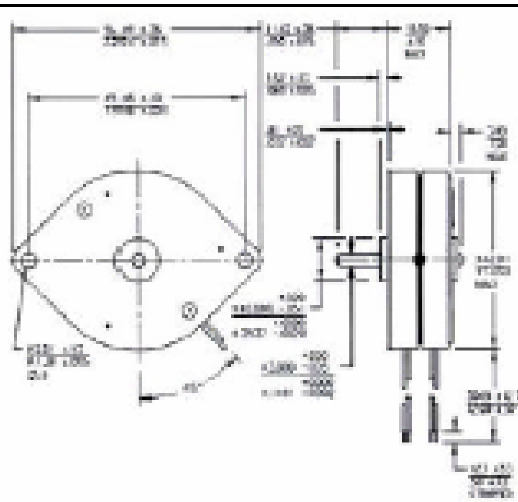
* Measured with 2 phases energised

Performance curve





Dimensional Drawing



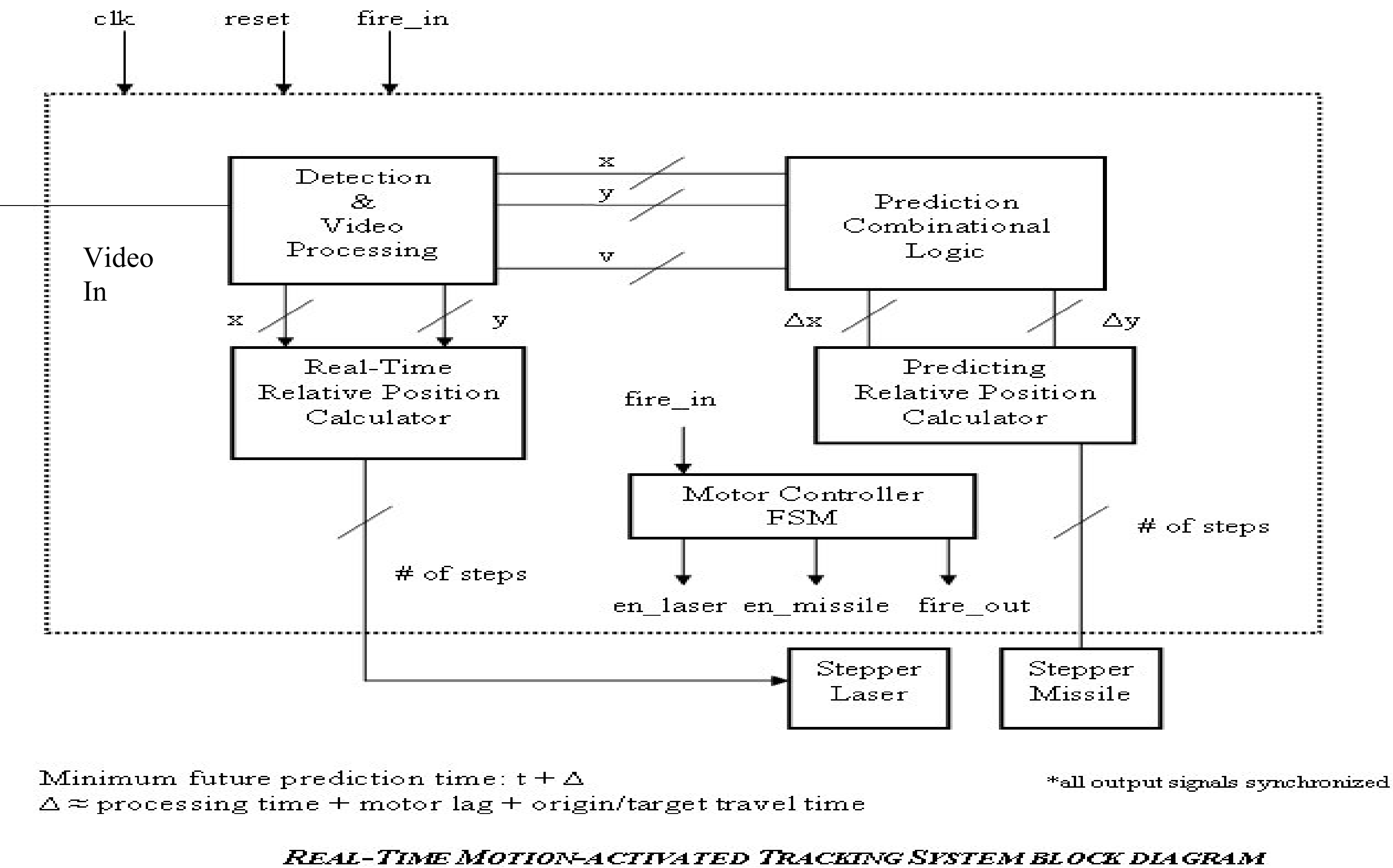


Figure 3: Mechanical subsystem