Cycling Timing System :

Mid-course Design Review

Submitted in partial fulfillment of

ECE Senior Design Project
(ECE 415)
Department of Electrical and Computer Engineering
Marcus Hall
University of Massachusetts Amherst
Amherst MA 01003

by

Patrick Bell
Emilio Gaudette
Eric Johnson

Submitted 14 December 2003

Project Advisor: Ramakrishna Janaswamy

Abstract

This paper discusses a proposed design for an electronic system used to time the finish of a cycling race by implementing radio frequency identification and precision location. After the problem has been introduced, our specific design solution will be discussed.

1. Introduction

We propose to improve upon the current technology for the timing, and more specifically, the ordering of the riders in a cycling competition. Via electronic means, we plan to develop a system that allows real time determination of finish time and finish place. Our major goals are to not only design a low-cost solution, but to achieve an overall greater timing precision than the current technology allows for. Our proposed design solution is to place a custom radio frequency transmitter on each bike that would uniquely identify that bike. We would then deploy two antennas near the finish line and connect them to a single receiver (which we will design). The receiver will be used to identify each bike as well as capture the exact time each rider crossed the finish line. A laptop computer will be connected to the receiver to download and process the raw data and provide both intermediate and final results, in real time.

2. Background

The order of finish of a bike race is determined by the front edge of the front wheel. The current standard is to use a video recording system to capture images of the riders as they cross the finishing line and subsequently determine the order of finish from that data. At the highest level of cycling, a digital camera with optical sensors, a narrow capture window, and extremely high refresh rate are used to relay a composite image (Figure 2.1) of the finish line to a connected PC where the image is stored for review. Using specialized software, the order of finished is manually determined (i.e. a human must identify each cyclist by their corresponding bib number on their jersey). In wealthier races, a $25,000 system (which employs magnetic signal transfer) is used to augment the video imaging system. However, this system can not fully replace the video system due to its lack of precision. This process can take a team of professionals more than 5 minutes to order a full field of about 200 riders, even with the magnetic signal transfer system. At lower levels of cycling, where the overwhelming majority of races take place, an ordinary VHS camera is used. This makes the process not only less accurate, but ordering a field of 50-60 riders (a common field size for the collegiate level races) can take the timing team hours.

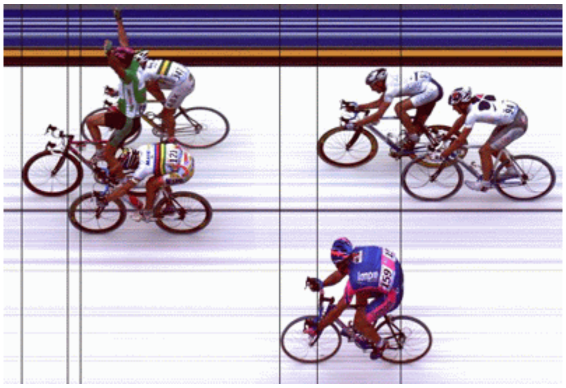


Figure 2.1

3. Design Criteria

The biggest criteria is that the system be able to furnish results in real time (with some negligible delay). A system that did not implement this feature would provide no advantage over the video capture systems. The cost of the system is also important, not only because of our project's budget, but also because the races that would benefit the most from such a system have relatively small budgets. The last, and most obvious, major design criteria is the accuracy of the system.

4. Design Issues, Considerations, and Assumed Values

There are many constraints on the problem that must be met by any design in consideration. The biggest constraint being the inability to guarantee line of sight in any direction from the front wheel of each bike. Depending on the terrain of the finish, the racers could finish at speeds ranging from 5 mph for an uphill finish, to 55 mph for a flat finish or downhill. It is not only possible, but expected that most, if not all, of the riders will finish in close proximity to each other due to tactical drafting. The width of the road is also variable. The radius of the front wheel, however, is fixed under international cycling regulations and we will assume that tire pressure and thickness are negligible in comparison with the error the remainder of our system will induce. The following is a list of estimated worst case values that, unless otherwise specified, are used in all calculations:

width of the road = 15 m
finishing speed = 60 mph = 26.82m/s
necessary processing time ≈ 700 us
derived accuracy (meters) from processing time at finishing speed = 0.02 m
assume no more than 256 unique transmitters are needed

5. Implementation Details

We will make use of the fact that all front wheels will have the same radius by placing a radio frequency transmitter on the fork of each bike, close to where the to the front wheel's hub is attached (Figure 5.0.1). Each transmitter will use a different carrier frequency which will be used to identify and locate each bike as it nears the finish line. We will now define an imaginary line, parallel to the finish line and one wheel radius before it (Figure 5.0.2). In order to determine when the front edge of the front wheel crosses the finish line, we must simply determine when the transmitter crosses the imaginary line.

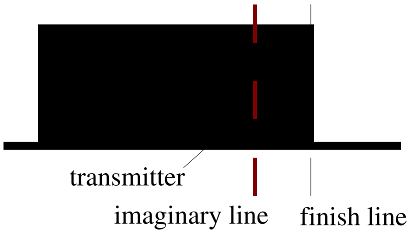


Figure 5.0.1

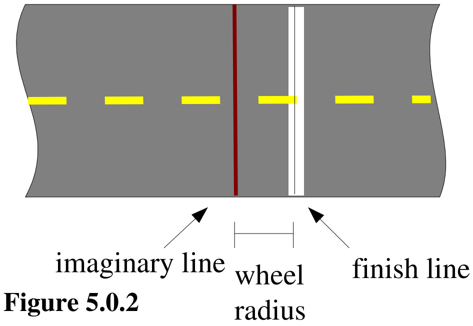


Figure 5.0.2

To determine, if a transmitter is at the imaginary line, we plan to use two omni-directional antennas, positioned vertically on the side of the road (Figure 5.0.3).

By looking at various phase differences in the signals received at each antenna, we will be able to determine exactly when each transmitter is within some small constant distance from the center line. The specific details on how this detection will be accomplished follow in the next section.

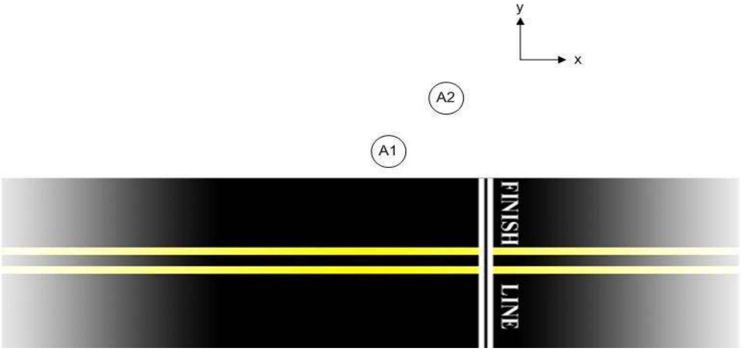


Figure 5.0.3

5.1 Analog Receiver

We intend to use the antenna setup shown in Figure 5.0.3, with antennas A1 and A2 arranged as shown. We will detect the phase difference between the signals received at A1 and A2. This phase difference (ϕ_d) can be used to determine the position of the bike transponder (shown as a blue dot) relative to the antennas.

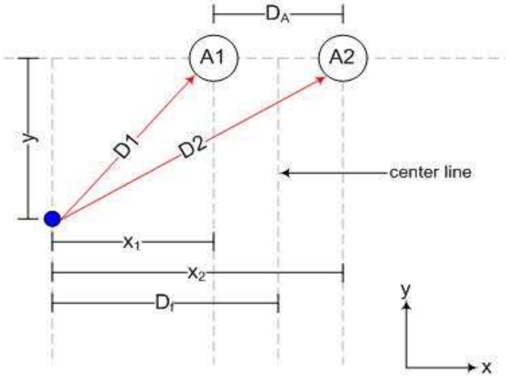


Figure 5.1.2

Figure 5.1.2 shows the pair of antennas (A1 and A2) and from said figure we get the following equations for the difference in distance, E.

$$E = D2 - D1 = \sqrt{(x_2^2 + y^2)} - \sqrt{(x_1^2 + y^2)}$$

$$x_1 = D_f - \frac{D_A}{2}$$

$$x_2 = D_f + \frac{D_A}{2}$$

$$E = \sqrt{\left(D_f + \frac{D_A}{2}\right)^2 + y^2} - \sqrt{\left(D_f - \frac{D_A}{2}\right)^2 + y^2} \qquad (1)$$

The phase difference ϕ_d, which would result from a distance error E, is:

$$\phi_d = E\left(2\frac{\pi}{\lambda}\right) \qquad (2)$$

where λ is the wavelength of the transmitted signal

$$\lambda = \frac{C}{f} \qquad (3)$$

where C is the speed of light and f is the frequency of the signal

We want ϕ_d to be no greater than $\pi/2$ (explained later) so it follows from equation (2) that we want $E \le \lambda/4$. In order to ensure that this is true, it is first important to note that when $y = 0$, equation (1) shows that $E = D_A$. This is the most that the value of E can ever be. Also, when the transponder is on the centerline of the two antennas the value of $E = 0$, which is its minimum value. Therefore if you move the transponder in an arc from the line A1-A2 to the centerline, the corresponding value of E will decrease. By using equation (2) and setting $E = D_A$ and $D_A \le \lambda/4$, we ensure that $\phi_d \le \pi/2$. If we substitute $\lambda/4$ in for D_A in equation (1), we get:

$$E=\sqrt{\left(D_f+\frac{\lambda}{8}\right)^2+y^2}-\sqrt{\left(D_f-\frac{\lambda}{8}\right)^2+y^2} \qquad (4)$$

Now, in order to show that we want $\phi_d \le \pi/2$, the method for obtaining ϕ_d must be explained. Each bike transmitter will be sending out a pure cosine wave at one of 256 different frequencies between 860MHz and 885MHz. The waves received at A1 and A2 (R1(t) and R2(t) respectively) will consist of (at worst) all 256 different bike signals (each at a different frequency) along with channel noise. First we pass the received signals through a set of parallel band pass filters that pass the frequencies from 860MHz-885MHz. These band passed signals will then be multiplied by the signal $\text{Cos}(2\pi\ 860*10^6)$, giving us signals R1(t)' and R2(t)' in the frequency range 0MHz-25MHz. In order to check the phase of a particular bike, we pass the signals into two (one for R1(t)' and another for R2(t)') parallel arrays of 256 fixed band pass filters (BPA1 and BPA2). The arrays will output the bike signals B1n(t) and B2n(t) (n=0,1…255) at their frequency Fn. B1n(t) and B2n(t) will only differ by phase. B1n(t) and B2n(t) will then be multiplied together to get:

$$E1n(t)=B1n(t)*B2n(t)$$

$$E1n(t)=A^2\ \cos(2\pi*Fn*t+\phi_1 n)*\cos(2\pi*Fn*t+\phi_2\ n)$$

$$=\frac{A^2}{2}*\cos(4\pi*Fn*t+\phi_1 n+\phi_2\ n)+\frac{A^2}{2}*\cos(\phi_1 n-\phi_2\ n)$$

where $\phi_1 n$ is the phase caused by D1 and $\phi_2 n$ is the phase caused by D2 for bike n

By passing E1n(t) through a low pass filter we end up with:

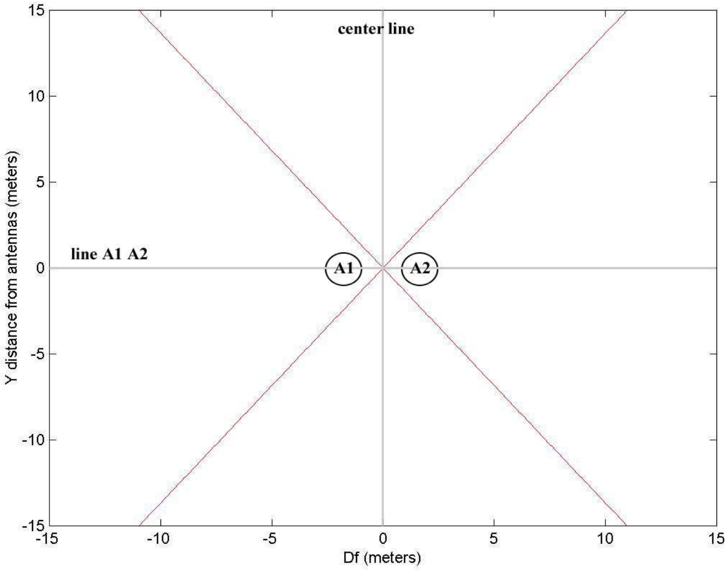
$$E2n(t)=\frac{A^2}{2}*\cos(\phi_1 n-\phi_2\ n) \qquad (5)$$

E2n(t) approaches 0 as $\phi_1 n - \phi_2 n$ approaches $\pi/2$ and E2n(t) approaches $A^2/2$ as $\phi_1 n - \phi_2 n$ approaches 0. In order to ensure that the $\text{Cos}(\phi_1 n - \phi_2 n)$ does not repeat any values, it is necessary to limit the phase difference, $\phi_d = \phi_1 n - \phi_2 n$, to be less than or equal to $\pi/2$. Then, the E2n(t) (which will be a DC value) becomes approximately equal to a constant, V_d, which is then passed into a voltage comparator.

$$Vd = \left|E2n(t)\right| = \left|\frac{A^{2}}{2} * \sin(\alpha n - \phi n)\right| \qquad (6)$$

By setting the comparison voltage, V_c, of the comparator we can "decide" which values of V_d will be considered in phase. So for a set V_d, we derive a constant error in distance E by applying equations (6) and (2). Using this constant E and equation (4) we can derive a boundary area where the bike is considered to be in phase by a pair of antennas. In other words, by setting E to be constant and providing values of y ranging from 0 to y_{max}, we can get the corresponding values of D_f (x component of the distance from the center of the antenna pair to the transmitter). To illustrate this we created files bike.m and runbike.m (see Appendix A) in MATLAB to produce Figure 5.1.3 that shows the line E = 0.038 meters (assuming that $A = \sqrt{2}$ volts and $\lambda = 0.3$ meters that gives us $\phi_d = 0.8$ rad/sec). The antennas A1 and A2, the center line, and the line A1 -A2 have been added for reference. The area bounded by the plotted lines represents the locations at which the bike would be considered in phase. Note that these lines are linear and that they are symmetrical about both the x-axis and y-axis (where the origin is located at the intersection of the centerline and the line A1-A2).

Figure 5.1.3



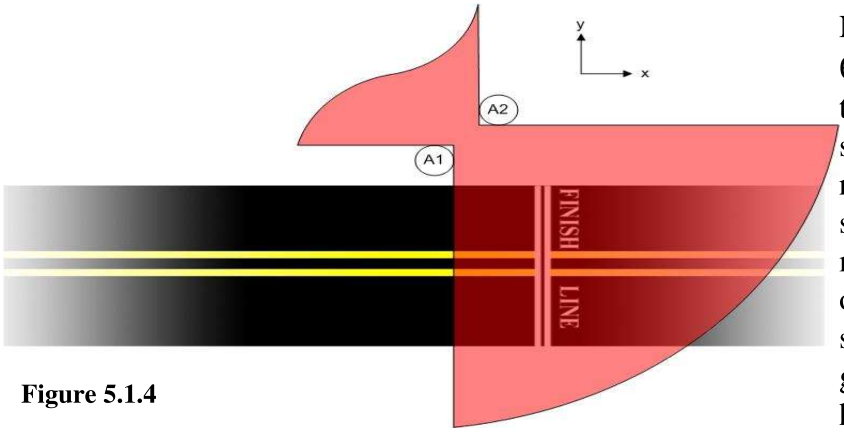


Figure 5.1.4

Now if we change the angle θ_R($\theta_R = 0$ when A1-A2 parallel to the road) of the line A1-A2 so it is no longer parallel to the road we get the following as shown in Figure 5.1.4. This means that as long as the output of the comparator is sampled often enough, we can guarantee that the bike will have only moved a certain distance: this distance is our error. Figure 5.1.5 illustrates the entire functional block diagram of the Analog Receiver module. This module produces the signals Cn (bike n inside the area) which are sent to the Digital Control Block (DCB) for further processing.

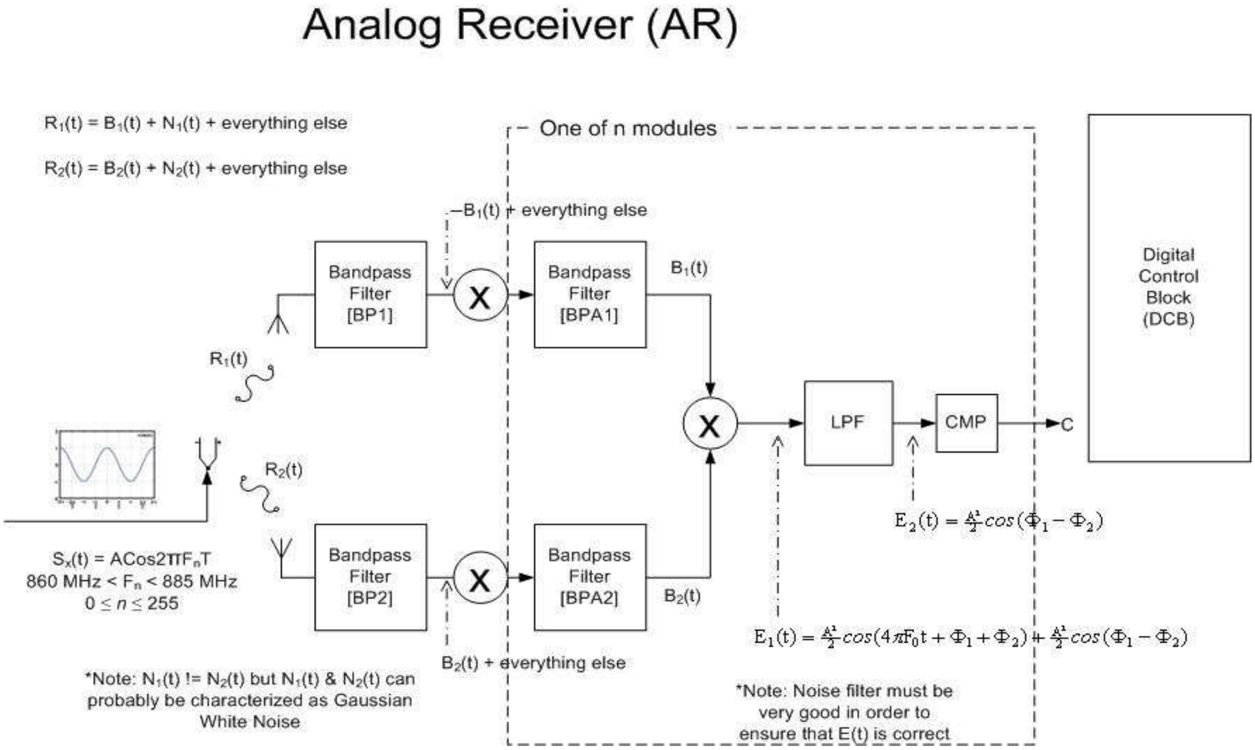


Figure 5.1.5

The specific accuracy for our system is dependent on the amount of time it takes for the entire system to process the data for all 256 bikes. We would like our accuracy to be better than 2 cm at the maximum finishing speed. The follow values

V = speed at which bike is moving while in range of A1 and A2 (m/s)=26.82 m/s
Ac_t = accuracy in time of the detector (s)
Ac_d = accuracy in distance of the detector (m) = 0.02m
T_{bpf} = time that a band pass filter takes to produce correct output (ie delay caused by a bpf)
T_{proc} = time to process data
λ = wavelength = $(3*10^8 m/s)*(1/900MHz) = 0.333m$
D_A = distance between A1 and A3 = $\lambda/4 = 0.0833m = 8.33cm$
Fn = frequency of bike n (n=0,1…255) = 860+n*0.09765 MHz
R_T = transmission radius of bike transmitters = 15m

$$T_{bpf} + T_{proc} \leq Ac_t$$
$$T_{proc} \leq Ac_t - T_{bpf}$$

Since we have already stated that we want the Ac_d = 2cm = 0.02m and that V=26.82 m/s then :

$$Ac_t = Ac_d / V = 0.02 / 26.82 = 0.000746s = 746us$$
$$T_{proc} \leq 746us - T_{bpf}$$

This makes it clear that the time we have to process the data is less than 746us.

5.2 Digital Control Block

The digital control block (figure 5.2.1) has undergone a number of major revisions since the beginning of this semester. The initial design consisted of a majority of registers and the latest design is centered around a PIC microcontroller. These revisions are based on changes in the Analog Receiver, as well as simplifications and improvements in our initial schematic.

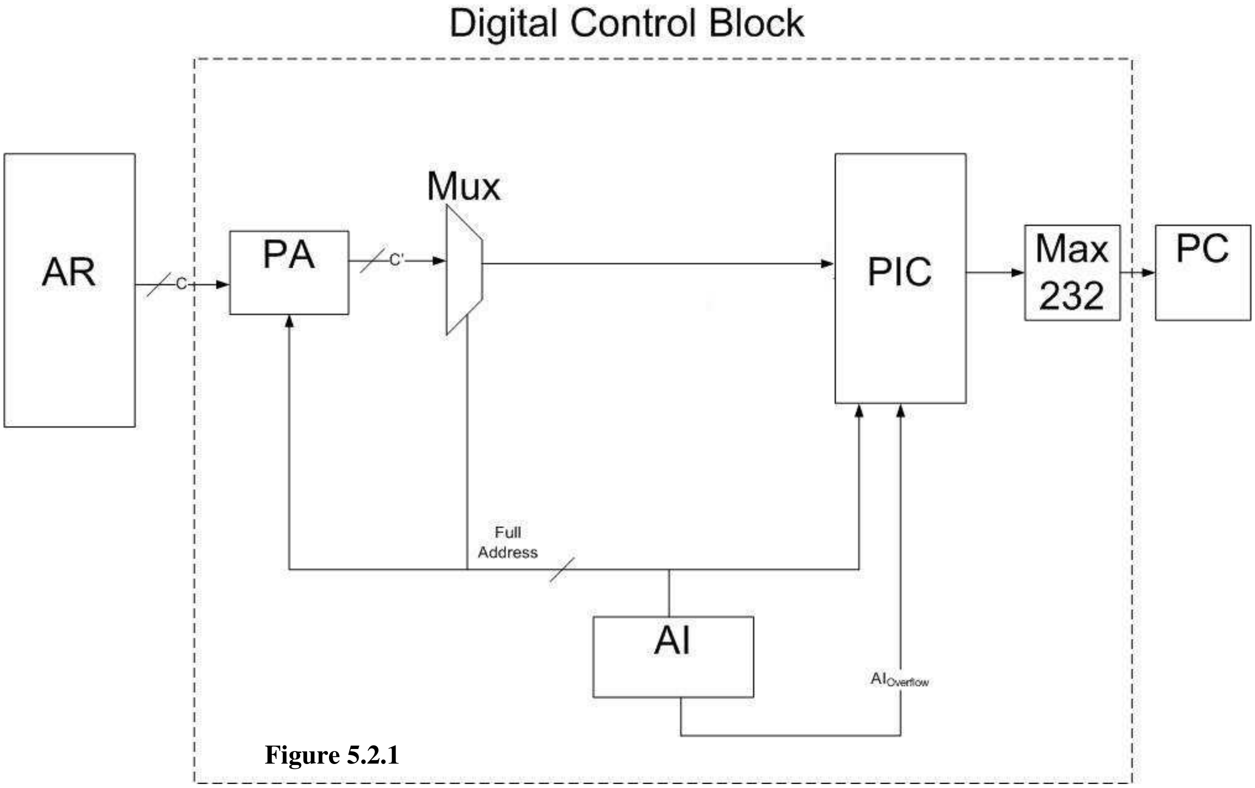


Figure 5.2.1

Before we can discuss the flow of control, it is important to have a better understanding of each of the main components. The first main component is the Address Incrementer (AI). This particular component is a counter that drives a majority of the other components. The AI iterates from 0 to 255, where each of these numbers corresponds to a unique bike ID. The AI also outputs an overflow bit every time it resets to zero. The next main component is the Pulser Array (PA). Its main purpose is to ensure that each bike is only counted once and it is controlled by the AI. Another major component is the multiplexer. The MUX selects bike data indicated by the AI. The last major component is the PIC microcontroller, but in order to better understand the PIC, it is necessary to describe the flow of data up to the PIC.

The flow of data begins when the ‘C’ bit for each bike comes into the DCB from the Analog Receiver. The ‘C’ bit is then held in the Pulsar Array until it is selected by the MUX. The selected ‘C’ bit then enters into the PIC along with its unique address from the AI. The other data entering into the PIC consists of the Address Overflow (generated by the AI). The PIC takes all this data and performs a number of operations which is described below.

The main role of the PIC is to capture the lap time data in real time. It maintains an Address Counter which simply stores the number of AI overflows. This is accomplished through an interrupt service routine triggered by the AI overflow bit. The lap time data consists of the Bike Identification (AI) appended to the end of the Address Counter. The lap time data is used to not only determine the lap time, but also the particular Bike ID. The PIC buffers this lap time data in an interrupt service routine triggered by the C input. When the PIC is not servicing an interrupt, it continually sends out buffered information to the PC over the serial interface.

5.3 Laptop Interface

The design and implementation of the system up to this point is obviously far more challenging than this simple data transfer and manipulation. As such, in order to focus our efforts on the difficult task of collecting the data, we plan to keep the Laptop Interface simple, with enough implemented to show functionality. In future development stages, an Ethernet connection would be a logical upgrade. Most laptop computers on the market today not only come standard with a high speed Ethernet connection, but they also lack serial communications ports. An Ethernet connection would also facilitate easily connecting multiple receivers to the laptop via a network hub in the case when the finish line is wider than the range of the radio transponders. It would also make the data available to a larger number of host computers, such as television media and commentators.

The laptop will employ a simple command line interface that will allow the user to capture and view that data as it is recorded and produce a simple text output file with each rider's number and time. If time allows, the laptop's user interface can be extended to include more features that would be desirable in a final production version of the system.

Appendix A

Contents of bike.m:

```
function[x] = bike(y, f, p)
% solve the distance equation for D for all values of y
len = length(y);
L=299792458/(f*10^6);
for i = 1:len,
   ty=y(i);
   tp=p(i);
   temp = solve('tp = cos(2*pi*(sqrt((L/8+D)^2+ty^2)
                                  -sqrt((L/8-D)^2+ty^2))/L)', 'D');
   x(i) = eval(temp(2));
end
```

Contents of runbike.m

```
function [] = runbike(phase, frequency)
% for a given phase limit and frequency ...
y=linspace(0,15,375);
p=linspace(phase,phase,375);
L=299792458/(frequency*10^6);
% calculate the distance from the finish line
%    for many values of y
x=bike(y,frequency,p);
slope=(y(375)-y(2))/(x(375)-x(2))
theta=atan((x(375)-x(2))/(y(375)-y(2)))
plot(x,y,-x,y,x,-y,-x,-y)
```