Measurement of the Three-dimensional Yarn Path and Tension in Yarn Unwinding

Thesis

Clemson University, Department of Mechanical Engineering

Advisor: Dr. Chris Rahn

by Markus S. Loffler

September 1996

TABLE OF CONTENTS

TABLE OF FIGURES

1. Introduction

1.1 Motivation

In many textile manufacturing systems, yarn is transported at high speed. Spinning and over-end unwinding are two important example processes. The objective of this research is to develop an analysis system that enables the design of high performance yarn transport systems.

Ring spinning combines the twisting of loose fibers into yarn and the winding of the yarn into a single operation. The yarn passes through an eyelet and reaches the traveler. The traveler spins freely around the ring and drags slightly behind the bobbin rotation, causing the yarn to be wound onto the bobbin. (See Figure 1-1a)
The unwinding process transfers yarn from packages to manufacturing systems. The yarn is pulled axially from a helically wound, stationary package, through a fixed eyelet. (See Figure 1-1b)

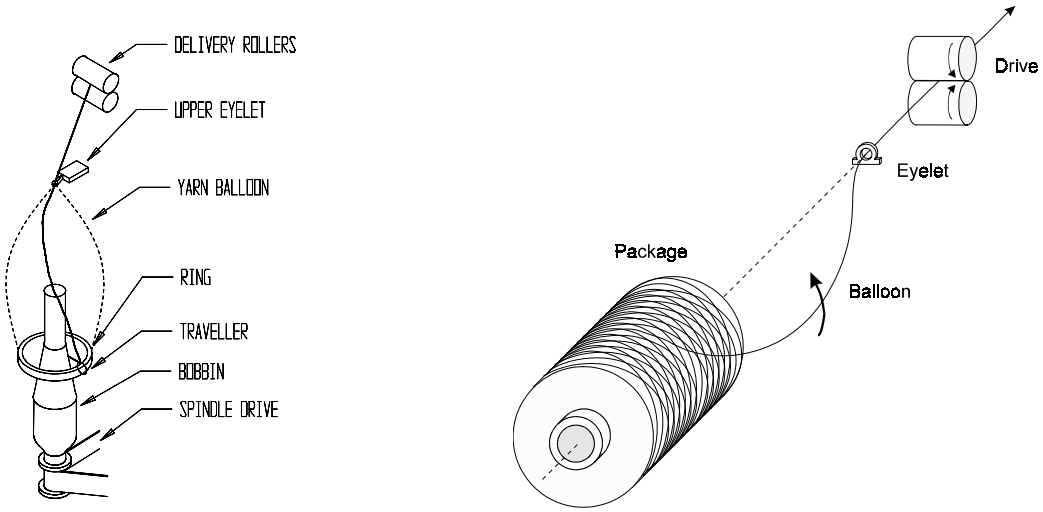


Figure 1-1. (a) Ring Spinning *(b) Unwinding*

Common to both processes is yarn rotating at high speed. The dynamics in the thread cause different shapes of the yarn path. The surface generated by the is called a "balloon."
A number of parameters influence the tension, including balloon shape and speed. If the tension gets too large, the thread breaks, resulting in a costly and time consuming interruption of production. The tension limits the maximum speed and therefore the productivity of the whole system. The objective is to optimize the processes to minimize the tension and allow maximum manufacturing speed without failure.

Theoretical studies of the yarn motion have been performed with the aim to calculate yarn path and the tension in the yarn. To verify these studies, measurement systems are needed to acquire yarn path and tension for comparison to theory.

A measurement system for the spinning process, called the "Yarn Balloon Test Stand" (YTBS), has already been developed at the department [1]. Since the balloon formed in spinning has a steady shape that is rotating at constant speed, it was possible to freeze it by adjusting a stroboscope manually to the rotation frequency. The result was a stationary yarn path that could easily be recorded with a camcorder. Simultaneously, a sensor measured the yarn tension. Then, the yarn path was determined from the video images by using a capture board and special purpose software packages. When the experimental balloon shapes combined with measured tension, the theory of yarn spinning could be verified.

This thesis describes the development of measurement system for the unwinding process, called the "Unwinding Analyzer." Because balloon shape and tension change very quickly during unwinding, it is no longer possible anymore to freeze the balloon with the stroboscope. Furthermore, a part of the yarn path touches the package surface and is therefore difficult to locate. Thus, this task is much more sophisticated.

A video technique has been developed to capture the high speed rotating yarn and determine the three-dimensional yarn curve. Simultaneously, the tension curve is acquired. Additionally, the system contains various functions to synchronize the capturing of the yarn path and tension acquisition to certain events.

The YBTS system used several independent standard hardware and software packages to acquire and process data. Rather than use this time consuming approach, the Unwinding Analyzer acquires and processes data on one PC. A Windows software package integrates all functions to control the system and to display and process the acquired data into one user interface. Hence, it's possible to do many experiments very quickly. Export functions are added to allow further data analysis and display.

1.2 Yarn Unwinding Theory

Figure 1-2 shows the configuration of yarn unwinding. The yarn is helically wound on the package. The angle ϕ of winding is called wind-on angle.
The thread is pulled from the package through the eyelet *O* with the constant speed *v*. It first starts moving away from its stationary position on the package surface at the point *U*, called "unwinding point." The point *L* where the yarn first leaves the package is called "lift-off point." Between lift-off point and eyelet, the yarn forms the "balloon."

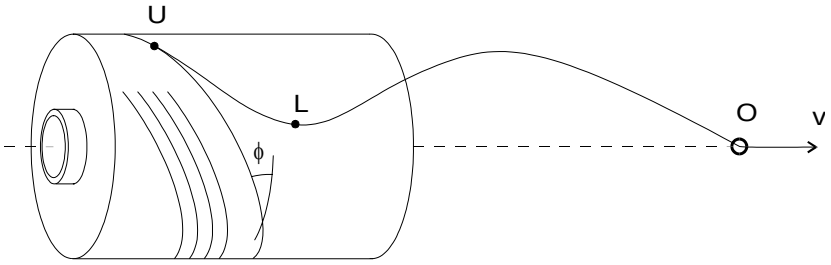


Figure 1-2. Configuration of Yarn Unwinding

A basic equation of motion had been developed that describes the yarn dynamics. This equation is applied to yarn spinning as well as to yarn unwinding.
At spinning, the fixed position of the eyelet and the at constant speed rotating traveler define the boundary conditions. These boundary conditions lead to a constant balloon shape in the steady state, which could be observed in experiment, too. One major result of the spinning investigations was that balloon shape and tension vary significantly with the length of the yarn path [1].
For unwinding, the eyelet position is also fixed, but the unwinding point and the lift-off point move up and down the package. This causes large changes in the length of the yarn path and therefore in balloon shape and tension.
The major problem to calculate the yarn path of unwinding is the part between unwinding point *U* and lift-off point *L*, where the yarn slides on the package. The development of theory for yarn unwinding has not been completed. The Unwinding Analyzer will be heavily used, however, to verify future yarn unwinding models. To compare theoretical to experimental results, the measurement system has to acquire the whole yarn path from the unwinding point to the eyelet. The tension varies along the whole path to the eyelet. To measure just the tension at the eyelet is sufficient for comparison to theory.

Additionally, the following parameters of the unwinding process are desired:
- Type of yarn (thickness, weight, material, wind-on angle)
- Unwinding speed
- Diameter and length of package
- Distance of the package to the eyelet

1.3 Objectives

A measurement system for the yarn unwinding process is developed that meets the following objectives:
- Capture the three-dimensional yarn path from unwind point to eyelet at a certain time point and determine the yarn curve from it.
- Develop a synchronization sensor for yarn rotation. This synchronization is used to fire the strobe at every revolution of the yarn balloon to freeze it. Also, the synchronization can be used to trigger capturing.

- Acquire the tension versus time curve immediately before and after the capture time. Determine the tension minimums and maximums and use them to trigger the capturing.
- Acquire the synchronization, trigger, strobe flashe and the capture times.
- Acquire all parameters of the experiment; including package type, yarn type, position and size of package, position of eyelet, and unwinding speed.
- Export all acquired data, images and parameters.
- Provide an easy to use graphical user interface.

1.4 Outline

Chapter 2 introduces the experimental setup and explains the concept of the measurement system. In the following six chapters, each component is described in detail. Chapter 9 describes how the components are integrated.
Chapter 10 presents the experiments made with the measurement system. Concluding remarks are made in Chapter 11.

2. Concept

2.1 Experimental Setup

The following hardware and software components are used the measurement system:

- Test stand for yarn unwinding. It contains a support for the yarn package and the yarn eyelet. The distance between package and eyelet is adjustable.
- Reiter Scragg yarn drive. The drive allows different withdrawing speeds, up to 2000 meters/minute. Compressed air pulls yarn into a waste container. The drive contains a speed sensor and displays the current unwinding speed. Figure 2-1 shows test stand and yarn drive.

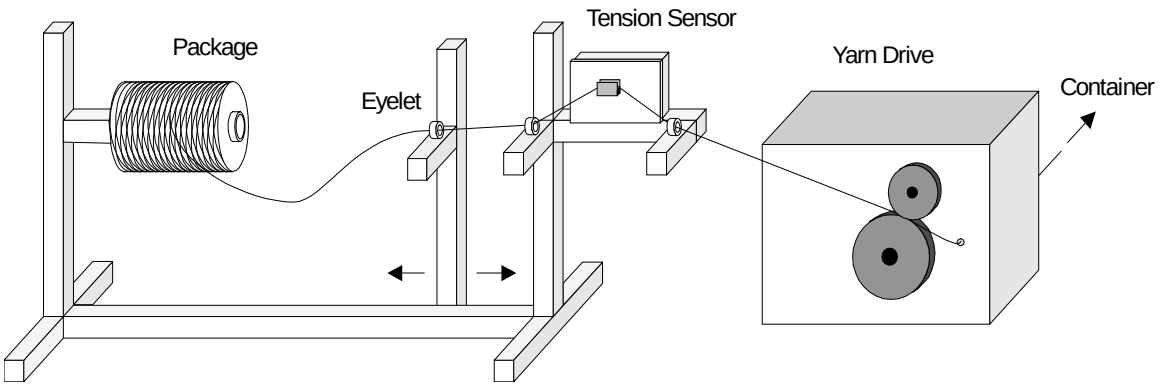


Figure 2-1. Setup of Yarn Unwinding Measurement

- PC, equipped with Pentium processor, 90 MHz, 16 Mbytes RAM. As a software development environment, Borland C++ and Borlands Object Windows Library are used. The PC contains three boards, described as following.
  - DSP board equipped with Digital Signal Processor TMS320C30, 33 MHz, 1 Mbytes RAM, two D/A converters, two A/D converters and two timers. The PC can access the DSP's memory and interrupt the DSP. A C-library allows the DSP to access these functions.
  - DS/2 input/output board. This board is directly connected to the DSP board and provides another two A/D and D/A converters as well as four digital inputs and four digital outputs. A C-library is used to access input and output functions.
  - Video capture board. It captures images from a common video signal to PC memory. A software development kit allows to access the capture board.
- Stroboscopes. Used in internal mode, the strobes generate flashes with an adjustable rate up to 12,000 flashes per minute. In external mode, an external TTL signal triggers the flashes.
- Tension sensor. This sensor is especially developed to be used for tension measurement in moving thread. It gives a voltage proportional to the tension.
- Video camera. It has special external synchronization and an internal frame buffer.

2.2 Overview

The whole system is controlled by the PC as shown in Figure 2-2. A Windows application integrates data display, control functions and image processing functions in one graphical user interface.

The yarn path is captured by a video camera. The image is transferred to the PC by the video capture board. The PC displays the image and does all image processing to get the yarn curve from the image.
The tension sensor determines the yarn tension. The tension is filtered and logged and the tension maximums and minimums are determined in real-time. The tension graph can be acquired unsynchronized or exactly at the image capture time. The PC reads the tension and displays it.
A sensor was developed that gives a synchronization to the balloon rotation. It provides a pulse, called “sync pulse” at every revolution of the balloon.

The system offers various trigger functions. The user can choose between sync pulse, tension minimum and tension maximum as a trigger source. These events trigger the capturing of the image or trigger the stroboscope. In the latter case, the stroboscope fires at every trigger to make the balloon visible to the naked eye.

As an important parameter, the unwinding speed is measured automatically and interfaced to the PC.

Generally, all signal processing and acquisition functions are made by software. As few as possible functions are implemented in hardware, because software is more error-safe than hardware, development is faster and changes are more easily made. As the PC with Windows does not have real-time capabilities, these functions are done by DSP board. The DSP transfers the acquired data to the PC. The tasks of the PC are to display and export the data.

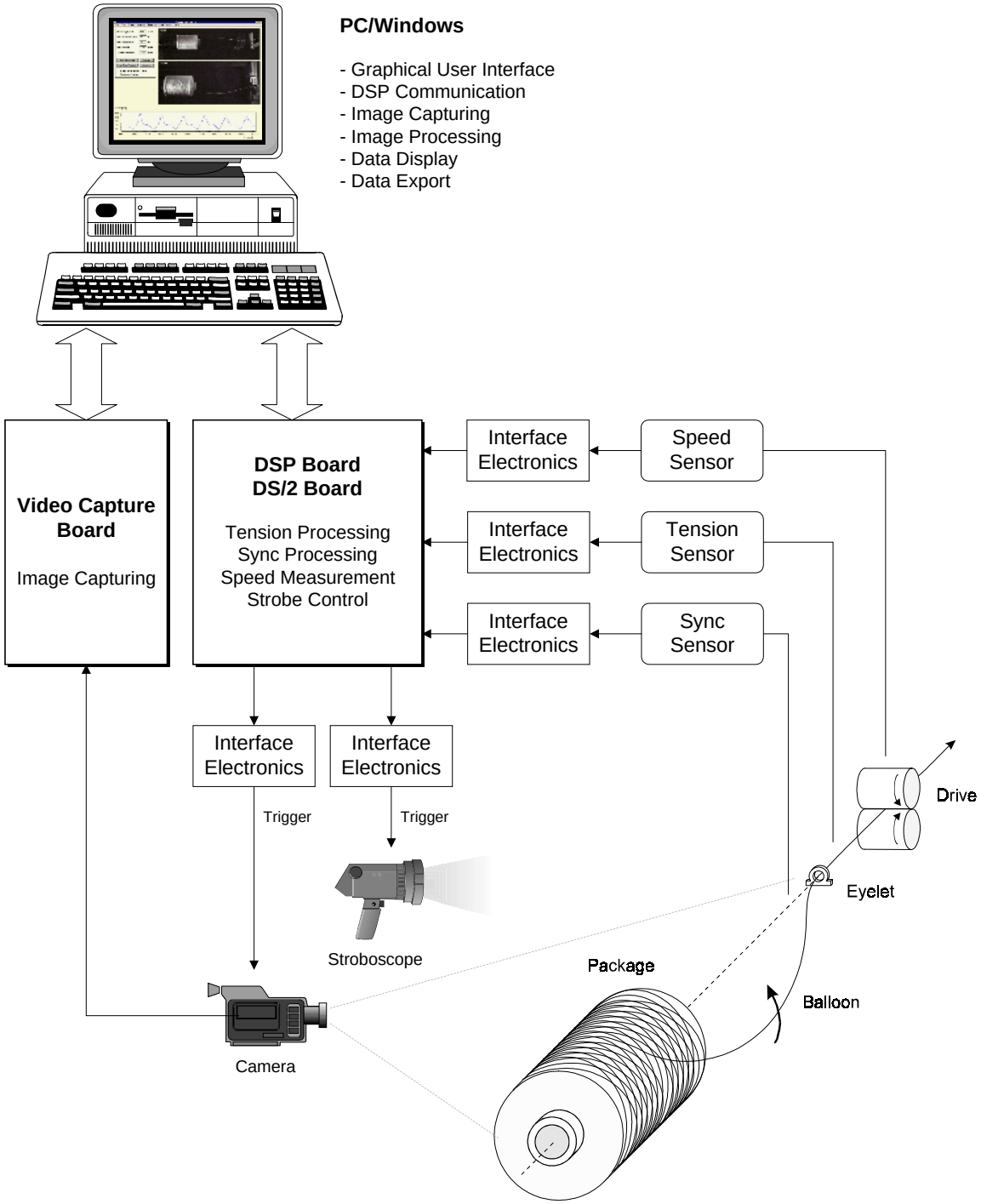


Figure 2-2. Block diagram of system

2.3 Components of the System

2.3.1 Synchronization

It is very important to get a signal that synchronizes to the balloon rotation. Otherwise the capturing would be at random points which make it hard to compare to the theoretical results. So an optical sensor is developed that gives a synchronization pulse when the balloon passes a certain angle of revolution. This pulse is used to trigger the path capturing.
In yarn spinning, the stroboscope could be adjusted to a constant frequency to freeze the balloon. Experiments showed that this is not possible in unwinding, because the revolution frequency is not constant. Hence, the synchronization pulse is also used to trigger the stroboscope to make the balloon visible to the naked eye.
The sync pulse is interfaced to the DSP board.

2.3.2 Tension Measurement

The tension sensor provides a voltage proportional to the tension in the yarn. This voltage is interfaced to one of the A/D converters of the DSP.
The processing of the tension signal is then realized in DSP software. The tension is filtered and logged. Since the objective of unwinding analysis is to find out how to decrease tension in the yarn, tension minimums and maximums are detected. They are used as a trigger for capturing and the stroboscope, to study the balloon at minimum and maximum tension.

2.3.3 Trigger

Sync pulse, tension maximums and tension minimums can be used as triggers. The user can select between these three trigger sources.
The trigger is used to flash the strobe continuously and to start the image capturing.
Additionally, the trigger can be delayed. These functions are performed by the DSP.

2.3.4 Image Capturing

The video camera takes a snapshot of the rotating yarn, triggered by the DSP board. The stroboscope is used for additionally lighting. The image provided by the camera is transferred to the PC by the video capture board. A software development kit for the board is used to access the image. The PC then displays the image.

The yarn path is determined from the image. Two methods are implemented, manual path detection, where the user sets the path points, and automatic path detection.

To get the three-dimensional yarn path, two orthogonal views of the balloon are necessary, for example the front view and the top view. From the two dimensional projections of the path in these views, the three-dimensional path is reconstructed.
There are three possibilities to get these two views. First, to use two cameras, one installed at the top and one in front of the test stand. This means additional costs of buying another camera and capture board. Second, take a picture of the balloon and a

delayed picture of the 90 degree rotated balloon. This results, however, in an error caused by changes of the balloon within 90 degrees.
Because of these difficulties, the third method is developed. A mirror is mounted on the top of the yarn stand at an angle of 45 degrees. The video camera positioned in front of the test stand simultanously captures the front view and the top view in the mirror. The three-dimensional path is reconstructed from the two-dimensional paths of the two views.

2.3.5 Strobe Control

The strobe is used in two ways. First, to flash continuously at the sync pulse to freeze the balloon. Second, as lighting for the image capturing with the video camera. The DSP can trigger a strobe flash by using one digital output of the DS/2 board.
An additional function is implemented to generate multiple strobe flashes. These flashes freeze a series of consecutive balloon shapes, making it possible to see the development of the yarn balloon.

2.3.6 Speed Measurement

A very important parameter of unwinding experiments is the withdrawing speed. To accelerate experiments, the speed is automatically measured by the system. The yarn drive's internal speed sensor measures the speed. This signal is interfaced to the DSP.

2.3.7 Data Export

The image from the video capture is exported as a Windows bitmap file. All other data, tension curve, yarn path and parameters, are exported as a Matlab file. Matlab can read the bitmap file and display it with the other data.

2.3.8 PC-DSP Communication

To get the data acquired by the DSP board and to trigger DSP functions, the PC software has to communicate with software running on the digital signal processor. The capability of the PC to interrupt the DSP is used to trigger DSP functions. For data exchange, the DSP memory is used. This memory can be accessed by the PC software.
A communication library is developed to provide these functions.

2.3.9 User Interface

The user interface has three tasks:

- Display all acquired data in an easy to survey form. This includes the display of the captured image and the graph of the tension curve as well as parameters like unwinding speed.
- Allow the user to control all functions of the system. This is done by a menu bar. Additionally, some buttons on the desktop provide quick control of these functions.
- Allow the user to set all system parameters. This is done in various modeless dialog boxes. Some parameters are also displayed directly on the desktop.

Windows is used as a platform for a graphical user interface (GUI). The software is written in the object-orientated language C++. Borlands Object Windows Library simplifies the programming of the graphical user interface. The menu and dialog boxes are created graphically with a resource editor and linked to the program.

3. Synchronization

3.1 Concept

To get a synchronization to the balloon rotation, a setup is necessary that gives a synchronization pulse when the yarn passes a certain angle. As in Figure 3-1 (a) illustrated, the lowest point of rotation is chosen as the synchronization point. At this point, the thread conforms an angle of zero degree to the vertical axis at the eyelet. It's important to know that the angle varies along the balloon because of air-drag effects, so the balloon doesn't lie in one plane. Therefore a zero angle at the eyelet doesn't automatically result in a zero angle of the whole balloon.

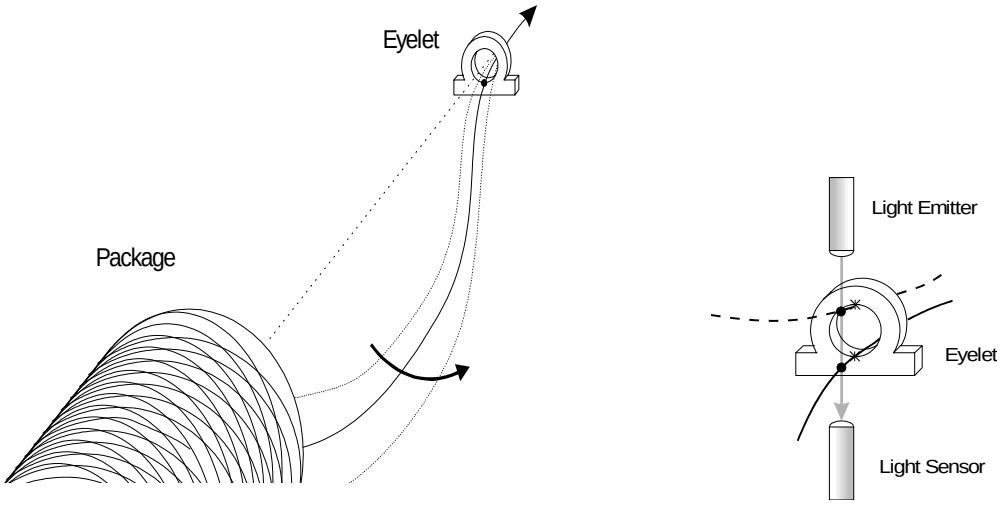


Figure 3-1. (a) Balloon angle at synchronization pulse, (b) Emitter/sensor arrangement

To detect the yarn at the synchronization point, an arrangement of light emitter and light sensor is used. The emitter and sensor are positioned as shown in Figure 3-1 (b). The yarn interrupts the light beam and causes a decrease of the light falling onto the sensor. This arrangement causes two sync points, one at the top and one at the bottom of the eyelet. The signal of the sensor will be the same for both points, so it is not possible to tell them apart. However, it is possible to use only every second sync pulse to detect just the bottom point.

The selection of the emitter and sensor is very important. The yarn moves quite fast and is very thin, so the interruption of the light beam is short. Thus, the sensor has to be fast. A plain infrared photodiode fulfills this requirement. An infrared light emitting diode (IRED) produces the light. Because the light beam of the IRED is not focused and the photodiodes light sensitive area is much bigger than the yarn diameter, the yarn interrupts only a small part of the light beam. Hence, the change in the signal generated by the photodiode is very small. Sensitive detection electronics are needed. On the other hand, there is no precise alignment of the IRED and the photodiode necessary as it would be with focused light.

3.2 Detection Electronics

Figure 3-2 shows a test circuit for the sensor arrangement. The usual voltage drop of the IRED is *1.6V*; the maximum current is *100mA*. To get the maximum light output, the maximum current has to flow through the IRED. So the resistor R_1 is chosen to be *34* Ω *= (5V - 1.6V) / 100mA*. The photodiode is used in reverse-biased mode. Then the current in the photodiode is proportional to the light. The current causes a voltage drop at the resistor R_2 that is proportional to the current and therefore to the light. Because this setup is very sensitive, a unity-gain voltage follower is used after it.

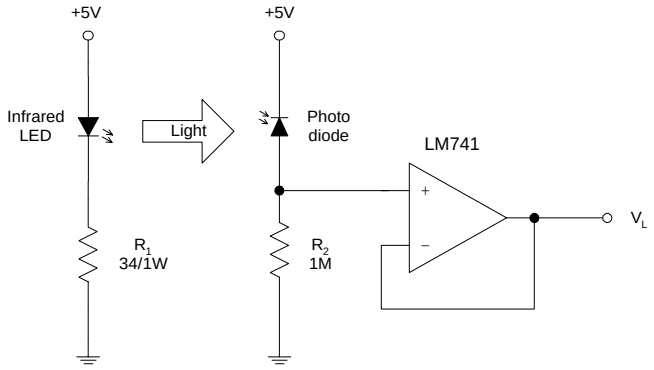


Figure 3-2. IRED/Photodiode test circuit

The top graph in Figure 3-3 shows an example output signal V_L. Each passing of the thread causes a drop of the voltage V_L. The length and depth of the drops vary.

With an operational amplifier used as a comparator, the analogue signal can be transformed to a digital one. Thereby, the signal is compared to a constant threshold voltage, indicated as the dotted line. A signal lower than the threshold produces a *-12V* output, otherwise the output is *+12V*. The middle graph shows this signal V_{DIG}. Voltage drop (a) in Figure 3-3 shows that the threshold adjustment has to be very precise. There is still a problem using this concept, namely there is noise on the signal. The noise leads to the effect, that the comparator starts toggling very rapidly at a signal near the threshold. That means bad edges of the pulses. It can be improved by using a comparator with hysteresis, but this still gives wrong results with voltage drops like drop (b) in Figure 3-3. Because the thread is spun from of tiny fibbers, one passing can result in a bunch of voltage drops causing multiple pulses in the digitized signal. The solution is to use a pulse generator. This module produces a digital pulse of a constant length triggered by an edge of the input signal. During the pulse, the input signal is ignored. So multiple pulses merge into one, as the bottom graph in Figure 3-3 illustrates.

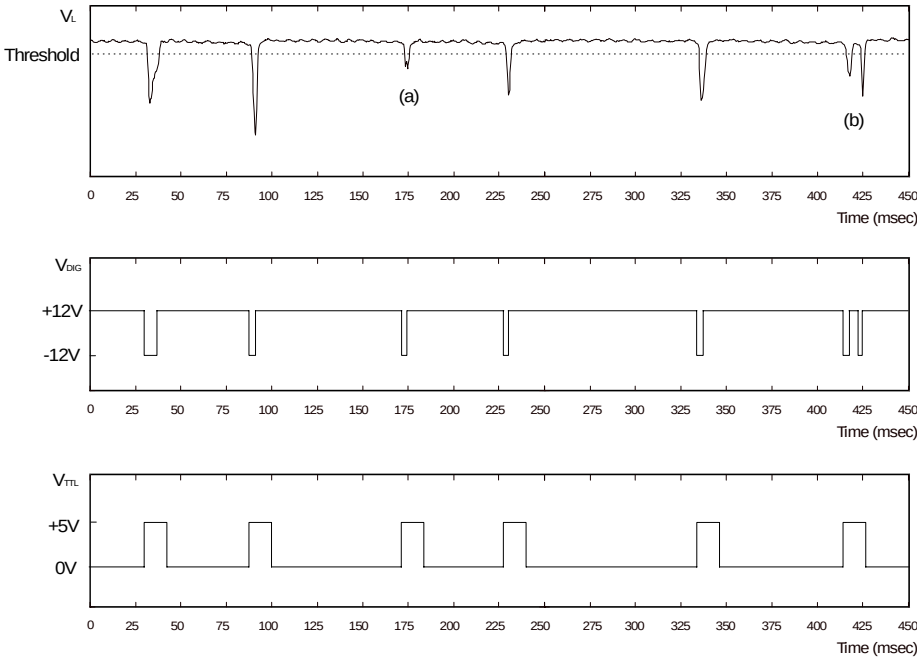


Figure 3-3. Photodiode signal and pulse generation

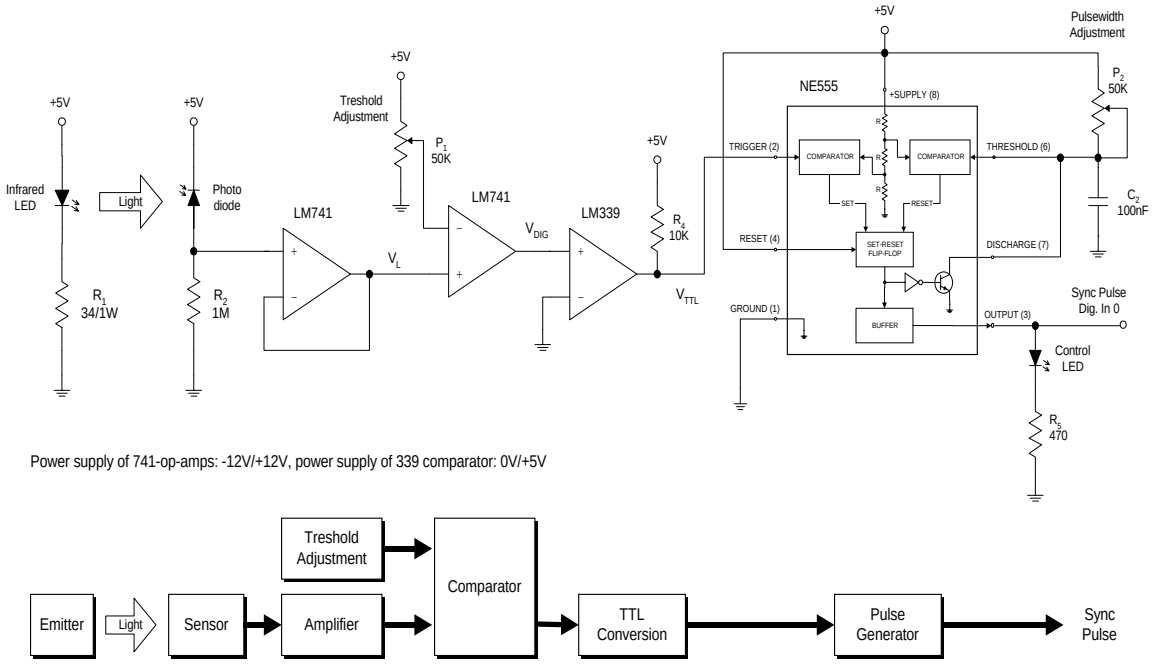


Figure 3-4. Wiring diagram and block diagram of sync pulse circuit

Figure 3-4 shows the complete circuit. The threshold voltage is adjusted by a potentiometer. Since the comparator op-amp has *-12V/+12V* output levels, an LM339 comparator is used to convert these levels to TTL's *0V/5V*. The TTL levels are needed

for the NE555 chip that is used as a pulse generator [6]. Every falling edge of the input signal results in a pulse of the length $t = 1.1\ RC$. With the 50K adjustable resistor, the pulse length can be set from *0ms* to *5.5ms*.
An LED is added as control signal of the sync pulse.
During tests of the circuit, the moving thread caused static charging of the whole yarn test stand. Suddenly the test stand discharged to the circuit which resulted in destroyed electronic parts. So the whole test stand was grounded. It was also necessary to use a shielded cable for the IRED and photodiode, to eliminate noise caused by the yarn drive.

3.3 DSP Data Acquisition

The sync signal is interfaced to one digital input of the DSP board. The DSP software detects the sync pulses. It is not possible to interrupt the DSP by one of its digital inputs. So the only way is to poll the digital input continuously till a raising edge is detected. Each sync pulse is further processed, shown in Figure 3-5.

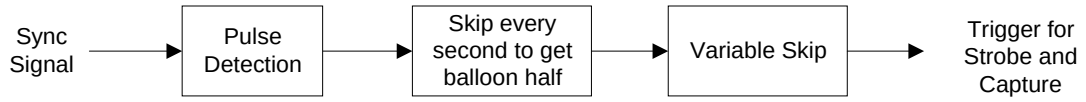


Figure 3-5. Function block diagram of sync signal processing

First, a function block makes it possible to get the thread only in the upper or lower position. It simply takes only every second sync pulse. (Each revolution causes two sync pulses, see Figure 3-1 (b), page 11)

Primarily, the sync pulse is used as a trigger for the capturing, it can also be used to trigger the stroboscope, too. Then, the strobe flashes at every trigger. This is an attempt to freeze the balloon at a certain position as in the same manner as in the spinning experiments. But experiments showed that the balloon formed at unwinding is not steady. As from theory supposed, the moving of the lift-off point up and down the package causes a different balloon at each revolution. As an interesting effect, it could be observed that the strobe freezed three or more different balloon shapes at the same time, and these balloon shapes were moving slowly. The reason is probably that after (in this case) three revolutions the lift-off point is almost at the same position again, forming a similar balloon. So a mechanism is added to skip sync pulses to freeze just one balloon at one time. This was possible, but the one balloon was still not stable and changing its shape continuously.

3.4 User Interface

The User Interface gives the user the following settings for the sync sensor:

- Switch the “Skip every second pulse” on and off
- Set the variable skip

The current mean revolution rate is displayed on the desktop.

4. Tension Measurement

4.1 Tension Sensor

The Reiter Scragg Type 1 tension sensor performs the tension measurement. Figure 4-1 is a functional drawing of the tension sensor head. The moving yarn slides against the two stationary outer pins on the tension head. The center pin is connected to a flat torsion spring which deflects under yarn tension. A circular metal disk is rigidly attached to the sensor end of the center pin. Tension in the yarn pulls the center pin down, twisting it. An inductive sensor produces a voltage linearly proportional to the deflection in the spring. An analog circuit conditions the voltage signal and outputs a *0 - 10V* proportional to the yarn tension. The range of the sensor is *0 - 50 gram*.

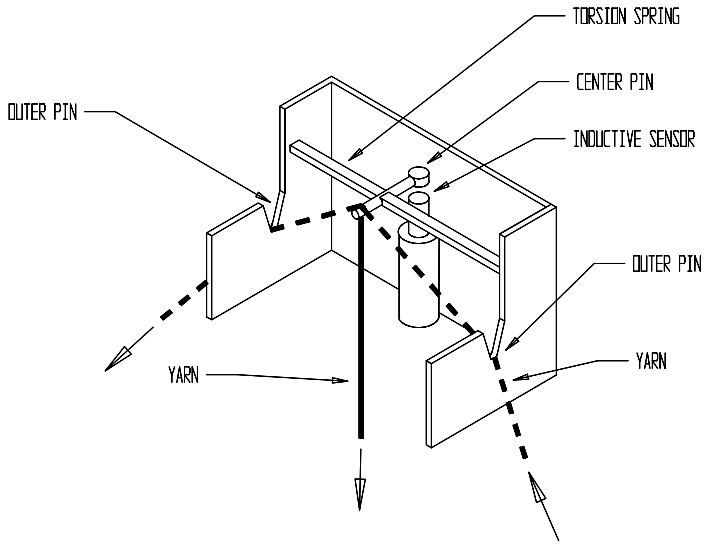


Figure 4-1. Tension sensor head

The tension in the yarn is different at each point along the thread. The only possibility to measure the tension is at the eyelet. So the tension sensor is mounted directly behind the eyelet, see Figure 2-1, page 5.

4.2 Interfacing Electronics

As mentioned, the output voltage of the tension sensor varies from *0V* to *10V*. The signal is interfaced to the A/D converter of the DSP board. This converter has the range *-3V* to *3V*, so a small circuit is necessary to adapt the signal range. Figure 4-2 shows the circuit. The unity-gain voltage follower is used as a buffer between sensor and circuit. The following op-amp operates as an adder. It amplifies the tension signal and shifts it to the

right range. It has the side-effect of inverting the signal, but the DSP software can invert it again to get back the right value.

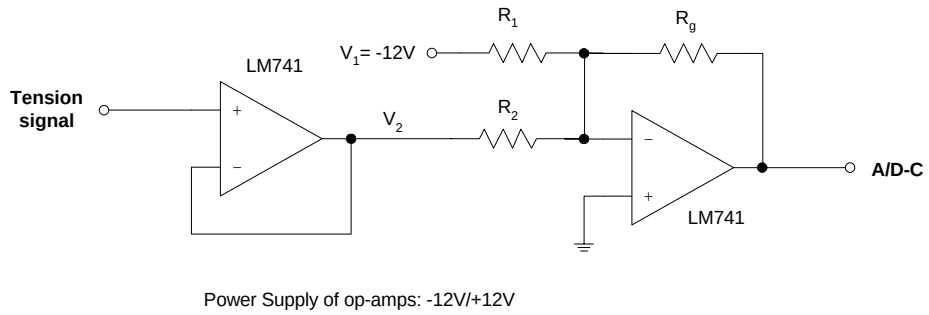


Figure 4-2. Tension signal interfacing circuit

The output voltage is:

$$V_{out} = -\frac{R_g}{R_1}V_1 - \frac{R_g}{R_2}V_2 = -\frac{R_g}{R_1}(-12V) - \frac{R_g}{R_2}V_t \tag{4.1}$$

The resistor values can be calculated as following:

(1) Gain: $\frac{R_g}{R_2} = \frac{10V - 0V}{3V - (-3V)} = \frac{5}{3}$ (4.2)

(2) Shift: $\frac{R_g}{R_1}(-12V) = 3V \;\Rightarrow\; \frac{R_g}{R_1} = \frac{3V}{12V} = \frac{1}{4}$ (4.3)

With $R_g = 12K$: $R_1 = 48K$, $R_2 = 20K$, $V_{out} = 3V - 0.6V_t$

The A/D converter of the DSP board has a built in filter, so it's not necessary to filter the signal externally before sampling to avoid aliasing effects.

4.3 DSP Data Processing and Acquisition

The tension signal is sampled continuously in the DSP program. One of the DSP's timers is used to get a constant sampling frequency. Experiments have shown that the bandwidth of the tension signal is less than *1 KHz*, so *2 KHz* sampling rate is chosen.

The ADC converts each sample to a 16 Bit signed word (*-32768* correspond to *-3V* input, *32767* correspond to *+3V*). In a calibration procedure, the ADC word for zero tension, t_0, and for *20 gram* tension, t_{20}, is acquired. The actual tension in gram is then calculated by fitting a line through these two points:

$$tension = 20gram \cdot \frac{(ADCWord - t_0)}{t_{20} - t_0} \tag{4.4}$$

The sampled tension goes through several processing functions, illustrated in Figure 4-3.

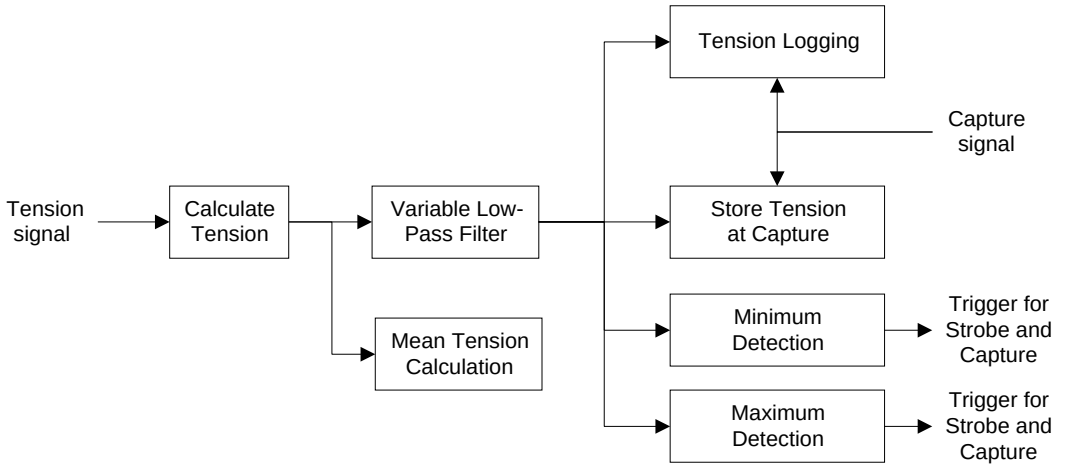


Figure 4-3. Function block diagram of tension processing

First of all, two digital low-pass filters are applied. The first has a constant cut-off of 1 Hz to calculate the mean tension. The second has a variable cut-off frequency. The spinning experiment showed that the tension signal is very noisy, illustrated in Figure 4-4 (a). The noise comes from the tension sensor itself and from the yarn drive. The digital low-pass filter realized in software removes the noise. It has an adjustable cut-off frequency. It's also useful to get only the lower frequencies of the tension signal.

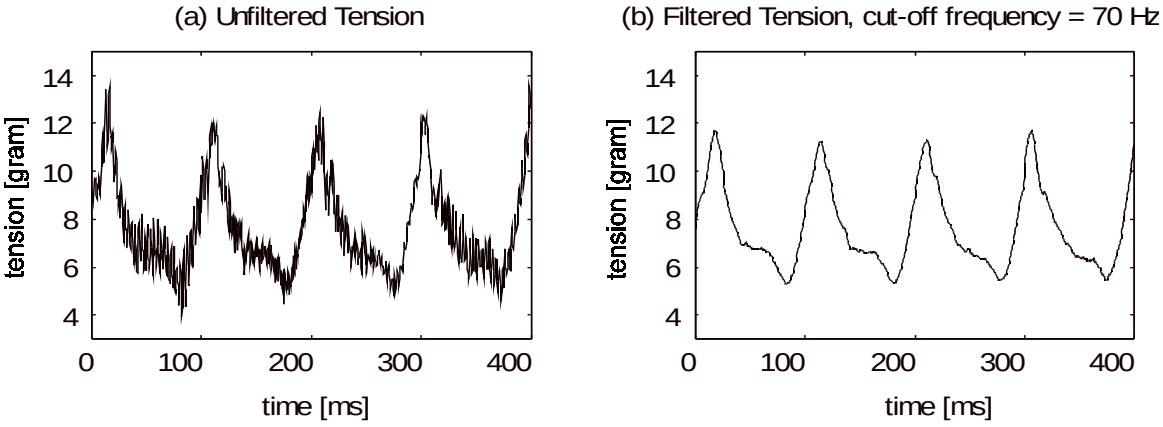


Figure 4-4. Filtered and unfiltered tension

The so filtered tension is then used for logging and maximum/minimum detection, that is described in 4.3.2 and 4.3.3, respectively.
The tension graph at the capture is interesting, so a mechanism is added to save the logged tension data around the capture point. Therefore, the capture signal, that indicates the start of a capturing, is needed. It comes from the capturing function blocks of the DSP software, described later. The capture signal also triggers the current tension value to be stored at the capturing point.

4.3.1 Tension Filtering

For determination of the mean tension and for the adjustable filter, the design of a digital low pass filter is necessary. A unity gain, first order Butterworth filter with the cut-off frequency ω_n can be described in the Laplace domain as:

$$H(s) = \frac{\omega_n^2}{(\omega_n + s)^2} \tag{4.5}$$

The transfer function $\left|H(s = jw)\right|^2 = \frac{\omega_n}{\omega_n^2 + \omega^2}$ is given in Figure 4-5.

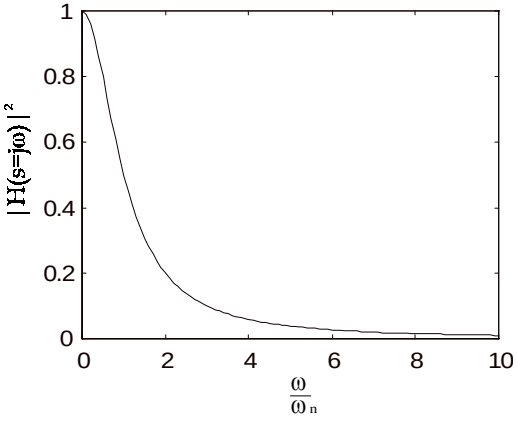


Figure 4-5. Spectrum $|H(s=j\omega)|^2$ of Butterworth lowpass filter

The bilinear transformation $s = \frac{2}{T} \cdot \frac{z-1}{z+1}$ converts the filter to the z-domain.

$$H(z) = \frac{\omega_n^2}{\left(\omega_n + \frac{2}{T}\frac{z-1}{z+1}\right)^2} = \frac{\omega_n^2 T^2 (z+1)^2}{\left(\omega_n T(z+1) + 2(z-1)\right)^2} \tag{4.6}$$

T is the sampling period, here $T = \frac{1}{f} = \frac{1}{2kHz} = 500\mu s$

With the substitution $m = w_n T$, $H(z)$ can be expressed as:

$$H(z) = \frac{m^2(z^2 + 2z + 1)}{m^2(z+1)^2 + 4m(z+1)(z-1) + 4(z-1)^2} \tag{4.7}$$

$$= \frac{m^2 z^2 + 2m^2 z + m^2}{(m^2 + 4m + 4)z^2 + (2m^2 - 8)z + (m^2 - 4m + 4)} \tag{4.8}$$

$$= \frac{m^2 + 2m^2 z^{-1} + m^2 z^{-2}}{(m^2 + 4m + 4) + (2m^2 - 8)z^{-1} + (m^2 - 4m + 4)z^{-2}} \tag{4.9}$$

The general form of a second order digital filter is

$$y(k) = a_0 u(k) + a_1 u(k-1) + a_2 u(k-2) - b_1 y(k-1) - b_2 y(k-2) \tag{4.10}$$

$$y(z) = a_0 u(z) + a_1 z^{-1} u(z) + a_2 z^{-2} u(z) - b_1 z^{-1} y(z) - b_2 z^{-2} y(z) \tag{4.11}$$

$$H(z) = \frac{y(z)}{u(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{4.12}$$

Compare of the coefficients gives

$$a_0 = \frac{m^2}{m^2 + 4m + 4}, \quad a_1 = 2a_0, \quad a_2 = a_0, \quad b_1 = \frac{2m^2 - 8}{m^2 + 4m + 4}, \quad b_2 = \frac{m^2 - 4m + 4}{m^2 + 4m + 4},$$

$$m = w_n T$$

Equation 4.10 can be realized directly in software.

4.3.2 Tension Logging

To get a continuous tension curve, the tension samples are stored in a buffer. The chosen buffer size of 50000 samples allows up to 25 seconds of tension data to be stored (sampling frequency is *2kHz*). The buffer is realized as an array in the DSP memory. To get the tension data, the PC reads the memory block of this array. The DSP library contains a function to read a complete memory block from the PC. To accelerate it, not the whole 50000 samples are transferred to the PC, but only a certain amount set by the user.

The samples are written consecutively into the tension buffer, see Figure 4-6. "`CurrentWritePos`" points to the latest tension sample and is increased at every new sample. The tension buffer is a ring buffer. That means, if the write position is at the end of the buffer, it starts again from the beginning. So the buffer contains always the latest 25 seconds of tension data, see Figure 4-7.

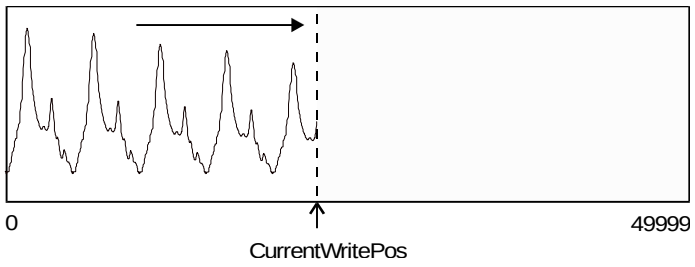


Figure 4-6. Data storage in the buffer

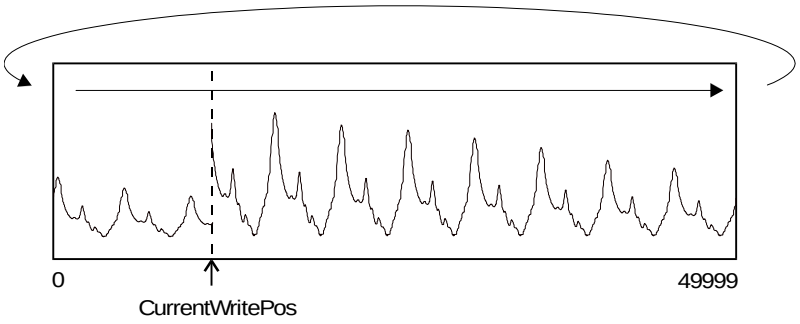


Figure 4-7. Ring buffer structure

Using a ring buffer means that the buffer isn't continuous anymore. Dependent on the current write position and the size of data requested, the PC can read one continuous block or has to read two blocks and put them together in the right order. Figure 4-8 illustrates the reading and reordering of 30000 tension samples from the buffer.

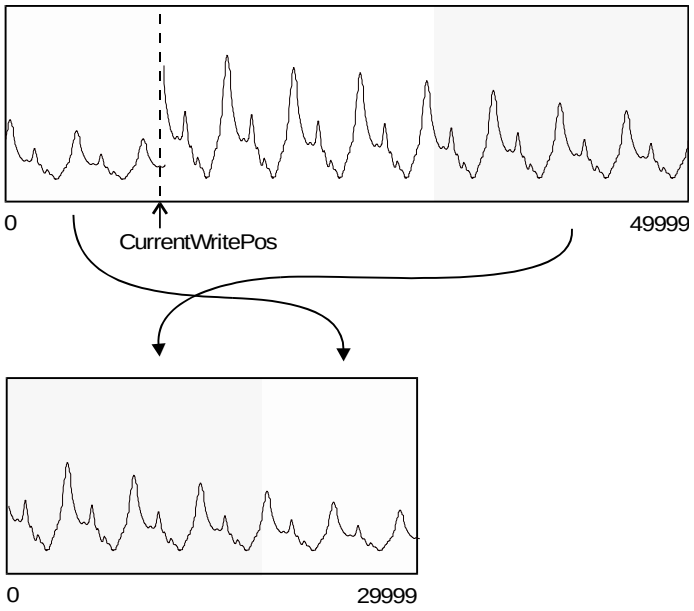


Figure 4-8. Reordering of the two buffer blocks

One very important thing is that new tension samples are still written into the tension buffer while the buffer is read by the PC. Therefore, the whole block of 50000 samples can't be read, otherwise tension data would be overwritten while reading, causing a corrupted buffer. To limit the maximum block that's read to 40000 samples bypasses this problem, since the data is written in an area while reading, that's not part of the block.

The tension at certain events like the sync pulse, strobe flash, trigger or capturing is very interesting. So the occurrence of these events is stored in the tension buffer as well. Since one word of the tension buffer is 32 Bit long (because of the memory organization of the

DSP) and the tension samples need only 16 Bit, the rest is used as flags for these events. Figure 4-9 shows the structure of one word of the tension buffer.

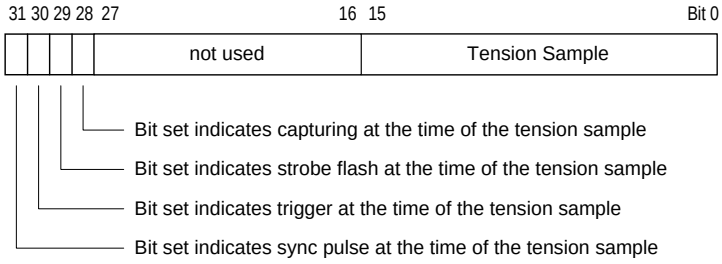


Figure 4-9. Structure of a word of the tension buffer

Most interesting is the tension graph at the time of a image capturing. Therefore, the current write position at a capturing is stored to a variable `CaptureWrPos`. As soon as the image capturing is completed, the PC uses this stored write position to request the tension buffer at the capture time. Because data before and after the capture time is needed, the tension buffer from `CaptureWrPos - num/2` to `CaptureWrPos + num/2` is taken if `num` samples are requested by the user.

4.3.3 Minimum/Maximum Detection

Figure 4-10 shows a typical tension graph. There are a lot of local maximums and minimums in the tension curve during one period. Only the global maximums and minimums during one period are interesting. So a threshold is set and just the first maximum that is bigger than the threshold is taken, analogically the first minimum that is smaller than the threshold. Figure 4-10 shows the dotted threshold line and which maximums the algorithm uses as a trigger.

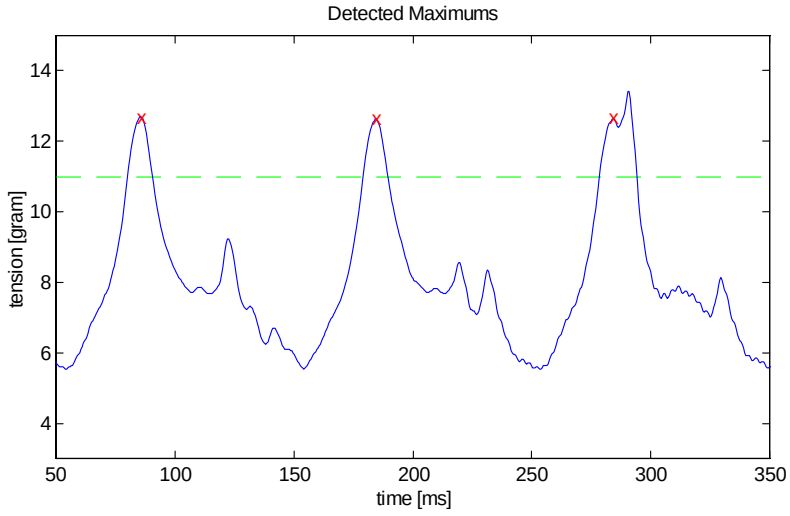


Figure 4-10. Detected Maximums

The following algorithm is used to detect tension maximums:

1. Determine the difference $\Delta T(k) = T(k) - T(k-1)$ of the current and last tension sample. Is $\Delta T(k) > 0$, the tension is rising, otherwise falling
1. Is $\Delta T(k-1) > 0$ and $\Delta T(k) < 0$, then there is a local maximum
1. If $T > T_{Threshold}$, and if it's the first local maximum after T has become bigger than $T_{Threshold,}$ the maximum causes a trigger

This procedure is done in real-time by the DSP, while the tension samples are read. The algorithm for detecting minimums is similar.

4.4 User Interface

The User Interface has four tasks:

- Display the tension graph
- Allow the user to change the tension acquire settings, that is: Filter cut-off frequency, size of tension data requested from DSP
- Allow the user to change the display settings of the tension graph
- Allow the user to set the threshold for tension maximums and minimums graphically

Figure 4-11 shows an example graph of the logged tension. The horizontal line is the threshold for maximum tension, which is used as trigger source in this example.

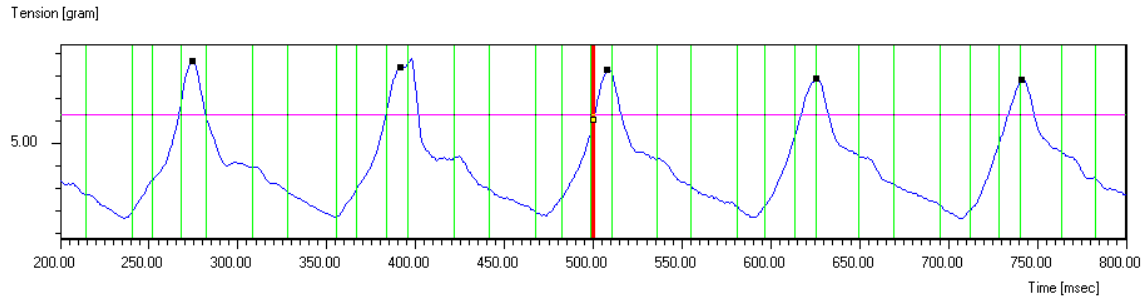


Figure 4-11. Tension graph

The special events that are stored together with the tension samples are displayed in the graph, too:

- Sync pulses are displayed as green vertical lines
- Triggers are displayed as black dots (here at maximum tension)
- Strobe flashes are displayed as yellow dots
- The capture time is marked with a thick vertical red line (here at 500 msec)

5. Trigger

5.1 DSP Processing

As shown in Chapter 3 and 4, there are different trigger sources that the user can select:

- Sync pulse
- Tension maximum
- Tension minimum

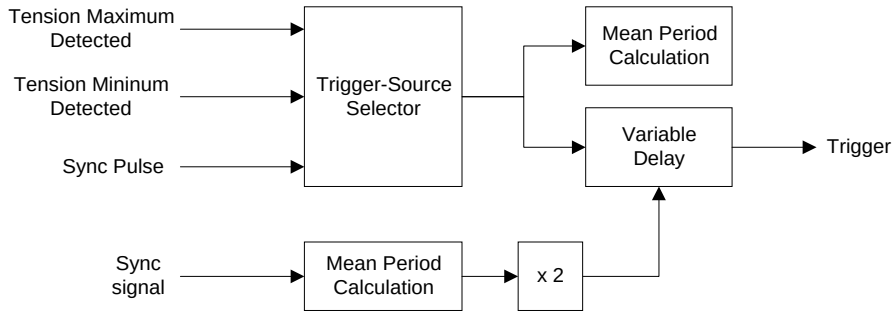


Figure 5-1. Function diagram of trigger processing

The selector in Figure 5-1 chooses one of these sources. A function is implemented to delay the trigger. So not only the balloon at the trigger can be investigated, but also the balloon that is rotated for a certain angle. The delay time is:

$$t = \frac{(delay - angle)}{360\ deg} \cdot (revolution\ period)$$

The revolution period calculation is done by another function block. One of the DSP timers is used to count a system clock. This clock can be used to calculate the time between two sync pulses. Multiplied by two, this time gives the revolution time. Because of the changing rotation frequency, it's necessary to determine the mean value of a bunch of revolution times.

The selected trigger is used for:

- Continuous strobe flashing at every trigger
- Start of the Capturing

See chapter 6, Image Capturing and Processing and chapter 7, Strobe Control for more information about the use of the trigger

5.2 User Interface

The User interface offers the following settings:

- Selection between sync pulse, maximum tension and minimum tension as trigger source

- Adjustment of the trigger delay

Additionally, the mean trigger rate is displayed on the desktop.

6. Image Capturing and Processing

6.1 Introduction

The image capturing of unwinding yarn leads to special requirements:

- Capturing of high speed movement
- Interfacing of the captured image to a Windows application
- Synchronization of the capturing to an external trigger
- Detection of the yarn path from the image
- Three-dimensional capturing and path detection

The following chapters will have a closer look at these points.

The video system has three components: Video camera, video capture board and stroboscope.
The video camera is a PULNIX TM-720VM black and white camera. It's resolution is 768 x 494 Pixels. It has two features that are very useful for the task of capturing the yarn path. First, it can be synchronized to an external signal. Second, it has an internal image buffer, that can store one snapshot.
The MONARCH Nova Strobe is used additionally to freeze the high speed yarn movement, to brighten the picture and to create shadows on the package surface.
The ComputerEyes CE/1024 video capture board digitizes the image and transfers it to the PC. A software library is provided to control the capture board and transfer the image data to an own Windows application.
Video camera and stroboscope are mounted on tripods in front of the unwinding test stand. Behind the test stand, the walls are covered with black fabric to get a black background for the video shots and therefore a good contrast to the white thread.

6.2 Video Capturing

6.2.1 Introduction

Figure 6-1 shows the configuration of image capturing by the PC. The video camera outputs an NTSC video signal. The capture board digitizes one image, triggered by the PC, and stores it into its internal frame buffer in maximum quality (1024 x 512 pixel, 16 million colors). This happens immediately with a maximum rate of 30 images/second.
The PC can read the frame buffer in different resolution and color modes. This takes usually some time (up to 1 second), dependent on the PC's performance.
This two-way concept is very advantageous, because the image can be captured quickly and afterwards, there's time to read it from the frame buffer. It can even be read from the frame buffer several times in different resolutions for different purposes.
For the unwinding capturing, it is read three times:

- To display it on the screen. The user can select the image resolution, make it lower to get more speed, make it higher to get more quality. For displaying the image on the screen, it has to be converted to a Windows bitmap first
- For the image processing in a constant high resolution
- To export the image. The user can select the image resolution, make it lower to save disk space (One 640 x 480 image in 16 Mio colors takes about 1 Mbyte memory)

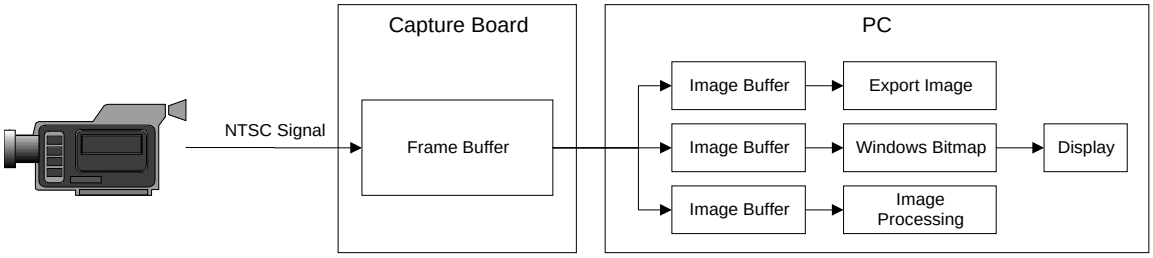


Figure 6-1. Image capture block diagram

To control the capture board, a C-library for the capture board is available.

6.2.2 Capturing of High Speed Movement

When talking of capturing high speed movement, this shouldn't be confused with high speed capturing. High speed capturing refers to the numbers of frames (images) per second that can be captured. There're several high speed cameras with over 200 frames/second. For the yarn unwinding measurement system, only one snapshot is required, so the capture rate is of minor interest. The normal capture rate of 30 frames/second is sufficient.

The decisive point of capturing high speed movement is the duration of exposure for the image. If the thread moves too much during this time, it causes blur in the image.
To expose the image, the video camera opens its shutter for short time. This time is usually called "shutter speed", although it is actually a time specification. The PULNIX camera has a maximum shutter speed of $\Delta t = 1/30{,}000\ sec$, that is fast enough for clear unblurred pictures. Unfortunately, the short exposure time results in very dark images. So the stroboscope flashes at every capturing to brighten up the image.

Even with a fast shutter, the capturing gives an unexpected result. The image shows two yarn paths of the yarn instead of one. The cause is the field concept of common video cameras, see Figure 6-2. The camera takes one image, also called "frame", in two separate "fields", one after the other. The first field contains all even lines of the image, the second all odd lines. A monitor or the capture board take both fields and put them together to the complete image. With a frame rate of the common 30 frames per second, the field rate is 60 fields per second. During the time between the two fields, 1/60 second, the thread moves, so field 2 captures the thread at a completely different position than

field 1. Using the stroboscope to freeze the movement at field 1, causes a dark field 2, so instead of a double image, the result is a single image with every second line black.
The only way to get rid of it is to take only one field. Fortunately, the capture board offers an option to get only field 1 or field 2. This eliminates double images or black lines, respectively. The inevitable consequence is a halved vertical resolution.

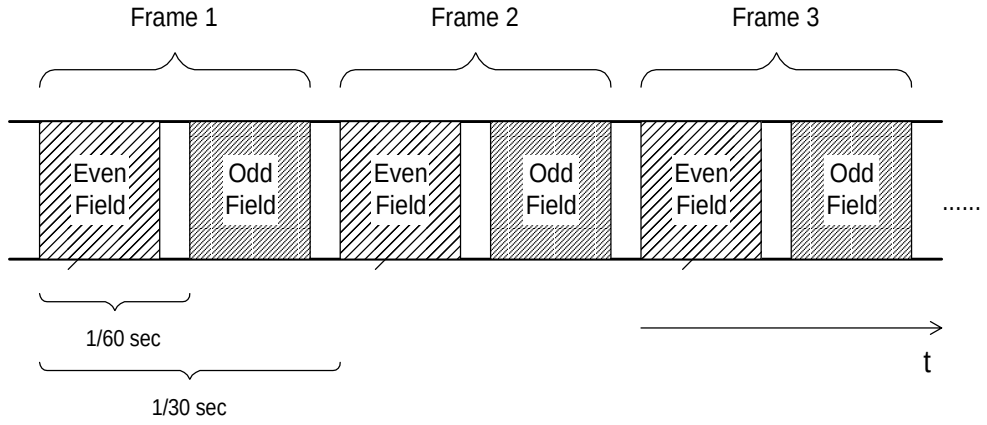


Figure 6-2. Field concept of video

6.2.3 Synchronized Capturing

Sync pulse, mininum tension and maximum tension can be used to trigger the capturing. Therefore, the camera needs to be synchronized to the trigger.
The TM-720VM provides the important feature of external synchronization, also called “asynchronous reset”. That means, if the camera gets an external TTL pulse, it interrupts its current field scanning, discharges the CCD elements and opens the shutter immediately to get a new image. Actually, the video capture board needs to be synchronized together with the camera to capture the image at the right time. The ComputerEyes CE/1024 capture board doesn’t have this function. But the TM-720VM has another feature, an internal image buffer. That means that the camera transfers the image at an external synchronization pulse to this image buffer. It can be described as an internal capturing in the camera. After that, the camera outputs from this image memory, giving a still picture of the taken snapshot. That means, the actual capturing to the PC can be done later, no synchronization is needed.
The sync pulse for the camera, a short switch from high to low and back, is generated by a digital output of the DS/2 board. Two options are implemented, unsynchronized capturing and synchronized capturing. In the first case, the pulse is generated immediately when the user starts a new capture. In the second case, the DSP program waits for the next trigger. Figure 6-3 shows this DSP function block, called “Capture Control”. The capture signal is needed for synchronizing the tension data acquiring with the capture, see chapter 4.3.2, and to tell the strobe control to flash the strobe at the capturing point, see chapter 7.

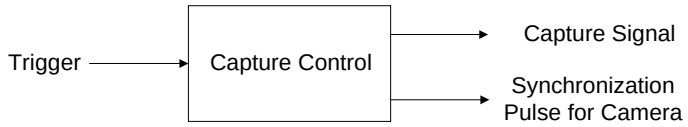


Figure 6-3. Function block diagram of capture control

6.3 Measurement of the Two-dimensional Path

6.3.1 Introduction

Figure 6-4 shows a example capture of the unwinding process. The free balloon part of the yarn is easy to recognize, because it's white thread in front of a black background. The part, where the thread slips from package is more difficult, because it is thread in front of background of wound thread. To get such a clear path on the package as in the image in Figure 6-4, a special stroboscope position is necessary. The strobe has to be positioned to fire in a direction parallel to the package axis. This creates a shadow of the thread moving from the package. Figure 6-5 shows the position of the stroboscope. Finally, four strobes are used simultaneously to get a good lighting. Two of them fire onto the package like described above, the other two fire directly on the balloon

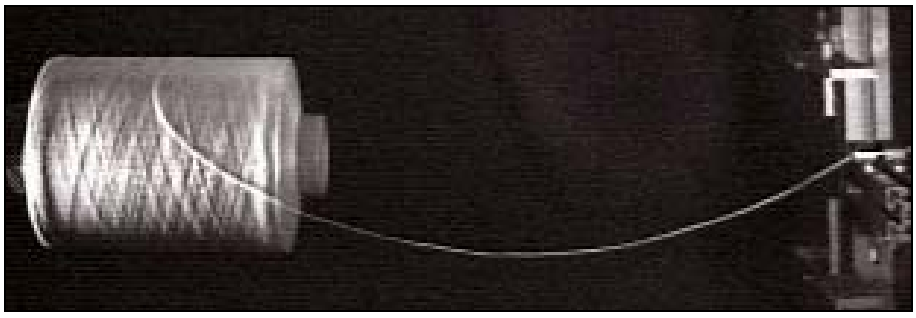


Figure 6-4. Example capture of yarn path

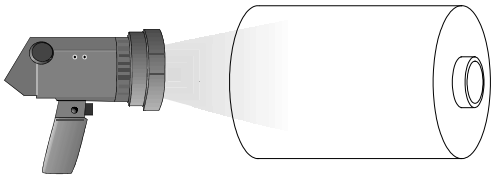


Figure 6-5. Position of stroboscope to create shadows

There are two ways to get the yarn path, a manual and an automatic one. The automatic path tracing algorithm, described in Chapter 6.3.3, finds only the free balloon part, not

the part on the package surface. However, the manual method can be used together with the automatic one to add that part.
The Windows software just exports the plain pixel coordinates of the path to Matlab. Further calculations are made in a Matlab program.

6.3.2 Manual Path Detection

As a first step, a manual detection of the yarn path is implemented. That means that the user clicks on the points of the yarn path. These points are marked and stored in an array. The user can also delete wrongly set points with the right mouse button.
The user may select the points in the wrong order, that means, it is not necessary to set the points continuously from the first point of the last one. It is possible to insert a new point between two existing ones, for example. Because it's still necessary to know the right order of the path points, a sorting algorithm is added. Figure 6-6 shows the algorithm.

$P = \{P_1, P_2, \ldots\}$ is the set of unsorted points

$S = \{S_1, S_2, \ldots\}$ is the set of sorted points

1. Choose the initial S to a random point of P, $S = \{P_x\}$.
   Remove P_x from S, $P = P \setminus \{P_x\}$
2. Find the closest point P_A to the first point of S and get its distance d_A
   Find the closest point P_B to the last point of S and get its distance d_B
3. If $d_A > d_B$, add d_A as the first point to S, Remove d_A from P, $P = P \setminus \{d_A\}$.
   If $d_A < d_B$, add d_B as the last point to S, Remove d_B from P, $P = P \setminus \{d_B\}$
4. Continue with 2. till $P = \{\ \}$

Figure 6-6. Algorithm to sort path points

The sorting algorithm tries to connect all points by the shortest path. This leads automatically to the right order, because a wrong order results in a longer path. The algorithm starts with one random point as the sorted path and adds the next neighbor to the beginning or end of the path, dependent upon which one is closer. It repeats this procedure till the sorted path contains all points.

The distance between two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ is the Euclidean distance:
$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

6.3.3 Automatic Path Detection

To accelerate the data acquiring, an automatic path tracing algorithm is implemented. This algorithm finds the path of the balloon part automatically. The very complex package part is not included in the detection.

The algorithm traces along the x-axis and finds the y-coordinate belonging to each x-coordinate. This method is preferred to tracing along the path, because insufficient lighting can result in interruption of the path line in the image.

To find a path point of a given x-coordinate seems to be quite simple, it is just to find a white point in that column. A closer look at the image data causes the question, what is white and what is black. For one lighting, the grayscale value of the background ranges from 0-10 and the brightness of the yarn is 50. For a different lighting, the background value changes to a range of 30-100 and the yarn value to 200. Though, the first task is to find the right threshold to separate between white and black pixel.

A look at the histogram of one column helps to find a solution. Figure 6-7 shows, that there is a widely spreaded area with a large amount of dark pixels and a small peak of bright pixel, caused by the thread in the image. Because this peak is so small, it can be neglected and the mean value of all pixels will give approximately the mean value of the dark background pixels. The brightness of the peak is approximately the maximum brightness in the column. A certain percentage between mean value and maximum value gives the threshold:

$$Treshold = MeanBright + percentage \cdot (MaxBright - MeanBright) \tag{6.1}$$

A percentage of 80%, as chosen in Figure 6-7, gives a good threshold for both images.

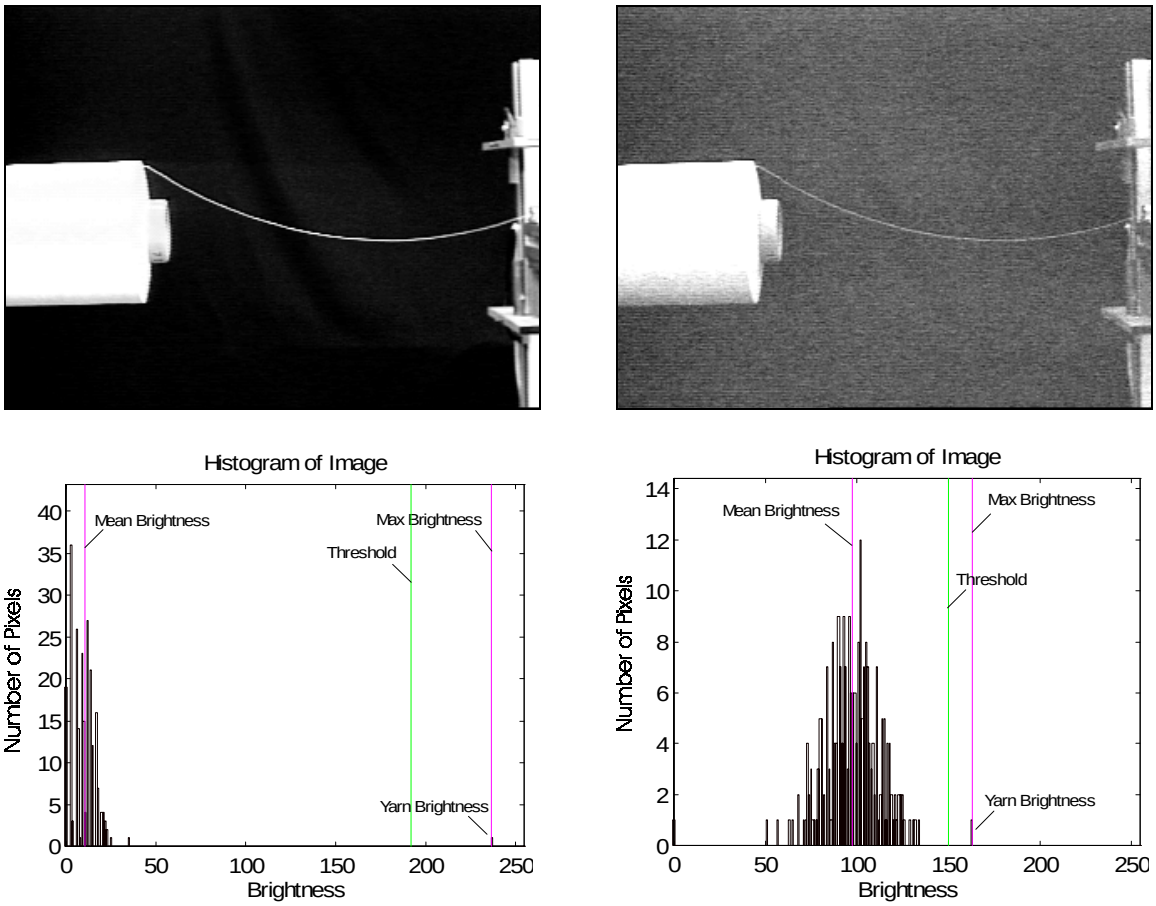


Figure 6-7. Images and histograms at two different lightings

The yarn is pretty thin, but in the image it usually consists of more than one pixel. So the first and the last bright pixel in one column is determined and the mean value between both is calculated as the yarn y-coordinate.

After finding the threshold, there are some other problems, see Figure 6-8:

- The package and metal parts of the test stand are detected as yarn
- Reflections of the metal parts are detected as yarn

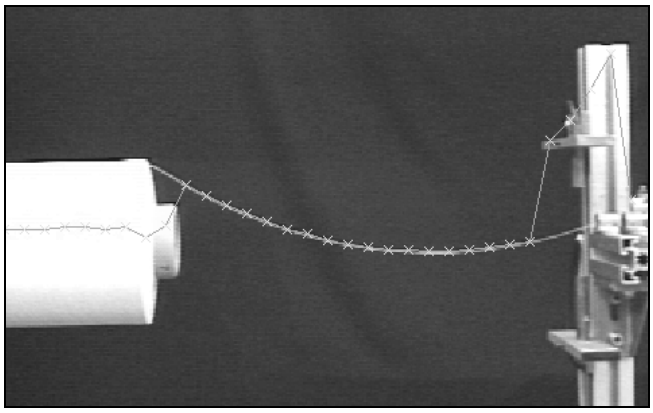


Figure 6-8: Badly detected yarn path

To avoid those wrong detections, conditions are added:

1. The length of the white area in the column is determined. Is it bigger than a threshold, it is not the yarn but the package or metal parts.
1. Reflections of the metal parts are not covered by 1., because their size is often only a few pixels. So the area below and above the detected yarn is examined. Is it yarn that was detected, there must be a certain amount of black pixels below and above the yarn. Otherwise, the white point was caused by a reflection.
1. Because 1. and 2. still at some cases, the gradient of the yarn path is observed. If a point is wrongly detected, it is usually out of order and the gradient gets quite big (see Figure 6-8). So all points beyond a certain threshold of gradient are not taken.

After adding these conditions, the extended algorithm shows very good results for different lightings. Wrong detections are seldom. The algorithm even detects the yarn path correctly in very dark pictures, when the yarn path isn't visible anymore to the naked eye.

6.4 Detection of the Three-dimensional Path

6.4.1 Concept

To capture the three-dimensional yarn path, two orthogonal views are necessary. The two views could be captured with two video cameras and two capture boards. For example, one camera is used for the front view and one for the top view.
Another concept that is cheaper and doesn't require big changes in hardware and software, is the mirror concept, see Figure 6-9. A mirror is mounted in 45° angle. The mirror reverts the top view to the front. The disadvantage is a loss of vertical resolution, because now two images are captured in one.

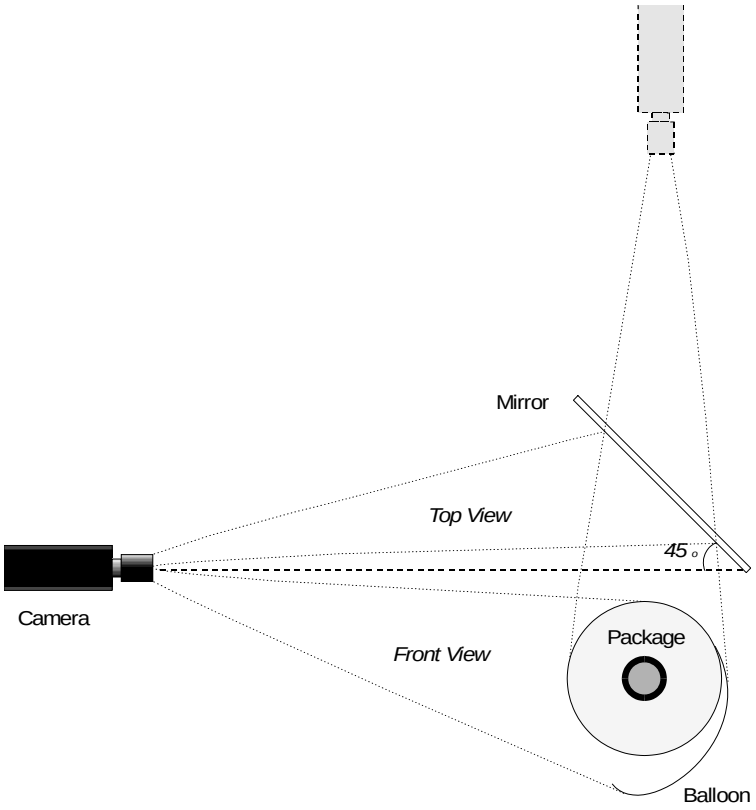


Figure 6-9. Mirror concept

For the path reconstruction, the part of the image, that belongs to the top view, is mirrored by the program. Then, it can be handled as it would have been taken with a separate top view camera.
To get two orthogonal views, the mirror alignment has to be very accurate. The mirror angle must be exactly 45°.

6.4.2 Camera Optics

The camera optics performs a perspective projection of the three-dimensional image to a plane, two-dimensional one. The object is transformed by the lens and imaged at a plane where the sensor CCD elements are located. The ray diagram in Figure 6-10 shows how a point *P* is projected to an image point *P'*.

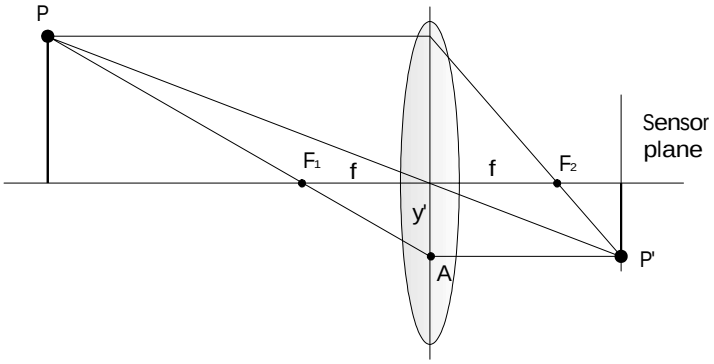


Figure 6-10. Ray diagram of camera optics

F_1 and F_2 are the focal points of the lens. Each parallel ray goes through F_2, and each ray that goes through F_1 is transformed to a parallel ray by the lens. Rays through the middle of the lens pass without change. The rays meet at the sensor plane in the image point *P'*. Instead of calculating the position of *P'* on the sensor plane, the position of point *A* can be calculated, which is easier. A coordinate system is placed with the origin in F_1, as shown in Figure 6-11.

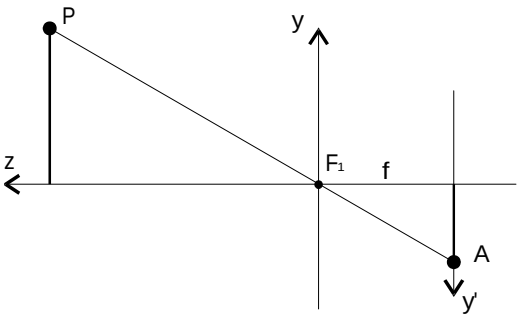


Figure 6-11. Transformation of lens

P, with the coordinates *z* and *y*, is transformed to *A* with the coordinate *y'*:

$$\frac{z}{y} = \frac{f}{y'} \quad \Rightarrow y' = \frac{f}{z} y \tag{6.2}$$

Extending the setup to a three-dimensional one leads to a *(x, y, z)* coordinate system.

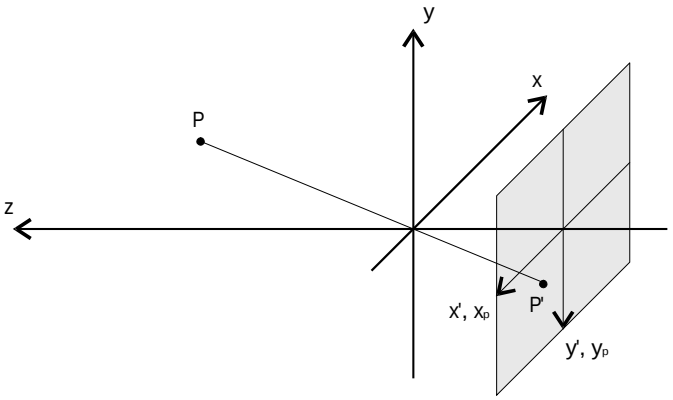


Figure 6-12. Three-dimensional camera optics

Equation 6.2 is applied separately to the x-coordinate and the y-coordinate. *x'* and *y'* is the position of the image point on the sensor plane.

$$x' = \frac{f}{z} x \tag{6.3}$$

$$y' = \frac{f}{z} y \tag{6.4}$$

To be independent from the size of the sensor plane, *x'* and *y'* are normalized to the width *w* of the sensor plane:

$$x_p = \frac{x'}{w} \tag{6.5}$$

$$y_p = \frac{y'}{w} \tag{6.6}$$

The points acquired by path tracing are also normalized to the image width. So (x_p, y_p) matches the coordinates of the acquired path in the image. With $a = f / w$, the equations of projection are simplified to:

$$x_p = \frac{a}{z} x \tag{6.7}$$

$$y_p = \frac{a}{z} y \tag{6.8}$$

$$a = \frac{f}{w} \tag{6.9}$$

An horizontal located object of the length $l = x_1 - x_2$ is projected to the length

$$l_p = x_1' - x_2' = \frac{a}{z} x_1 - \frac{a}{z} x_2 = \frac{a}{z} l \tag{6.10}$$

6.4.3 Coordinate Systems for Reconstruction

To apply equation 6.9 and 6.10 to the two camera views, two coordinate systems are placed properly. Figure 6-13 shows the position of the coordinate system of the front view, (x_1, y_1, z_1), and of the top view, (x_2, y_2, z_2). Additionally, a coordinate system (x, y, z) is used to connect both systems. The final reconstructed path will be related to this system.

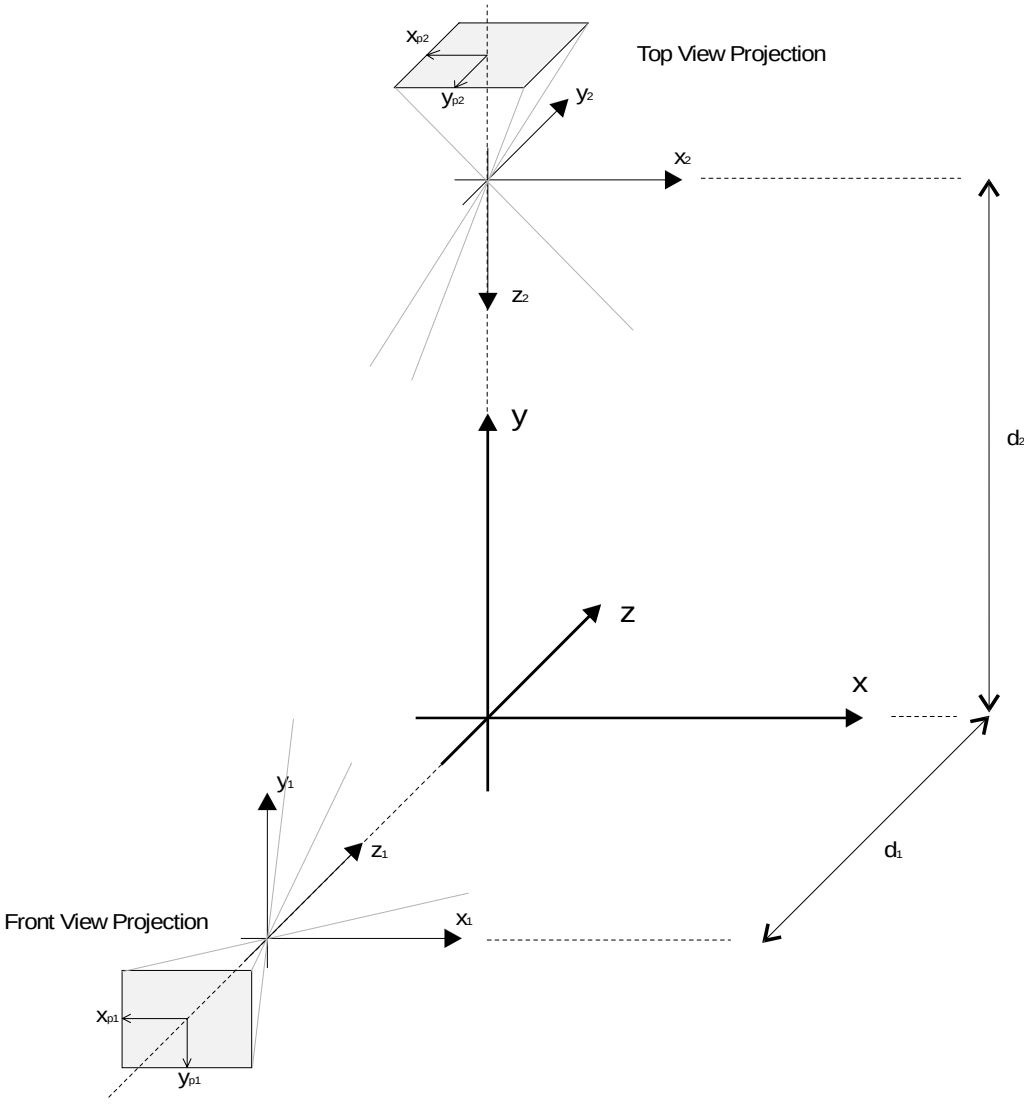


Figure 6-13. Coordinate systems for reconstruction

The conversion equations from one system to the other are

$$x_1 = x,\ y_1 = y,\ z_1 = d_1 + z \tag{6.11}$$

$$x_2 = x,\ y_2 = z,\ z_2 = d_2 - y \tag{6.12}$$

$$x_1 = x_2,\ y_1 = d_2 - z_2,\ z_1 = d_1 + y_2 \tag{6.13}$$

The equations of projection for both views are then:

Front view:

$$x_{p1} = \frac{a}{z_1} x_1 = \frac{a}{d_1 + z} x \tag{6.14}$$

$$y_{p1} = \frac{a}{z_1} y_1 = \frac{a}{d_1 + z} y \tag{6.16}$$

Top view:

$$x_{p2} = \frac{a}{z_2} x_2 = \frac{a}{d_2 - y} x \tag{6.15}$$

$$y_{p2} = \frac{a}{z_2} y_2 = \frac{a}{d_2 - y} z \tag{6.17}$$

The unknown distances d_1 and d_2 have to be determined. Because they are hard to measure, especially with the mirror, another procedure has to be used.
The distance l between eyelet and package is measured manually. The projected length l_{p1} of this distance in the front view and the length l_{p2} in the top view are measured as well.
From the l, l_{p1} and l_{p2}, the z-coordinates of the package axis in the corresponding coordinate system can be calculated by using equation 6.10. They're equal to the z-coordinates of the eyelet, $z_{1,e}$ and $z_{2,e}$, because the package axis goes through the eyelet

$$l_{p1} = \frac{a}{z_{1,e}} l, \quad z_{1,e} = a \frac{l}{l_{p1}} \tag{6.18}$$

$$l_{p2} = \frac{a}{z_{2,e}} l, \quad z_{2,e} = a \frac{l}{l_{p2}} \tag{6.19}$$

The positions of the eyelet in the front view, $(x_{p1,e}, y_{p1,e})$, and in the top view, $(x_{p2,e}, y_{p2,e})$ are measured in the captured image. From the equations of projection, the y-coordinates of the eyelet are calculated

$$y_{1,e} = \frac{z_{1,e}}{a} y_{p1,e} \tag{6.20}$$

$$y_{2,e} = \frac{z_{2,e}}{a} y_{p2,e} \tag{6.21}$$

Knowing the eyelet coordinates both in the (x_1, y_1, z_1) and (x_2, y_2, z_2) coordinate system allows to calculate d_1 and d_2 from the equation 6.13:

$$d_1 = z_{1,e} - y_{2,e} = a \frac{l}{l_{p1}} - y_{2,e} \tag{6.22}$$

$$d_2 = z_{2,e} - y_{1,e} = a\frac{l}{l_{p2}} - y_{1,e} \tag{6.23}$$

6.4.4 Reconstruction Algorithm

The path tracing procedures explained in chapter 6.3.2 and 6.3.3 are applied to both top and front view. The result are two (x_p, y_p) path arrays. The (x, y, z) yarn path has to be reconstructed from these two arrays.

It is possible to calculate a 3D path point from its top and front view projection. But it is not trivial to find the top view point P_{p2} that belongs to a point of the front view P_{p1}. So the following procedure is done: There are infinite 3D points that can belong to a front view point P_{p1}. All these points lay on a straight projection ray, called *projector*. So the unknown 3D point P can be anywhere on this projector. Then, the projector line is projected to the top view. There, it represents again all possible points that belong to the given front view point. Calculating the intersection between this line and the top view yarn path, gives the unknown point P_{p2}. From P_{p1} and P_{p2}, P is calculated.

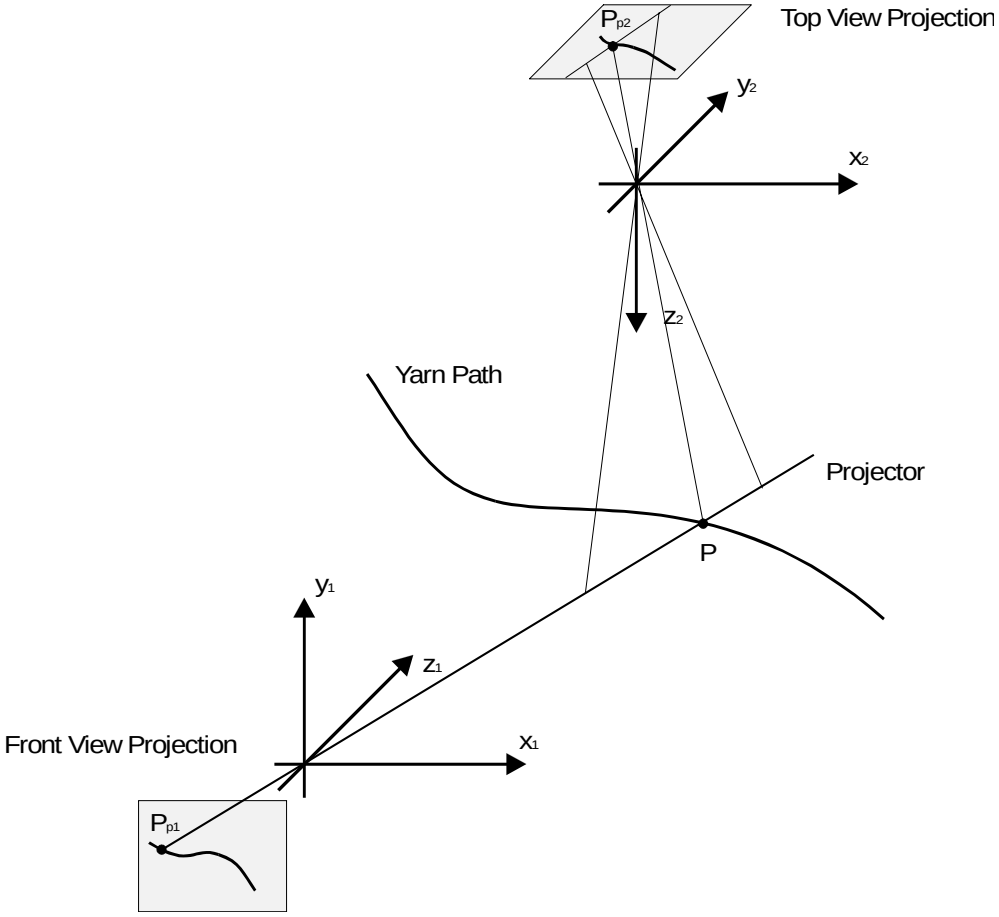


Figure 6-14. Concept of Yarn Path Reconstruction

The point P_{p1} of the yarn path in the front view is given by (x_{p1}, y_{p1}). Solving equations 6.14 and 6.16 for x and y gives

$$x = \frac{d_1 + z}{a} x_{p1} \tag{6.24}$$

$$y = \frac{d_1 + z}{a} y_{p1} \tag{6.25}$$

This is the equation of the projector of P_{p1}. To get the projection of the top view of this line, equations 6.24 and 6.25 are inserted into the projection equations of the top view, 6.15 and 6.17.

$$x_{p2} = \frac{a}{d_2 - \frac{d_1 + z}{a} y_{p1}} \; \frac{d_1 + z}{a} x_{p1} \tag{6.26}$$

$$y_{p2} = \frac{a}{d_2 - \frac{d_1 + z}{a} y_{p1}} \; z \tag{6.27}$$

This is again the equation of a line, because the perspective projection transforms a line to a line. Setting the line parameter z to two different values gives two points of the line

$$z = -d_1: \qquad x_{p2} = 0, \;\; y_{p2} = -\frac{d_1}{d_2} a \tag{6.28}$$

$$z = 0: \qquad x_{p2} = \frac{a d_1 x_{p1}}{a d_2 - d_1 y_{p1}}, \;\; y_{p2} = 0 \tag{6.29}$$

Using these two points gives the vector equation of the line

$$\begin{pmatrix} x_{p2} \\ y_{p2} \end{pmatrix} = \begin{pmatrix} \frac{a d_1 x_{p1}}{a d_2 - d_1 y_{p1}} \\ \frac{d_1}{d_2} a \end{pmatrix} c - \begin{pmatrix} 0 \\ \frac{d_1}{d_2} \end{pmatrix} \tag{6.30}$$

This equation describes a line that contains all possible points in the top view projection that belongs to (x_{p1}, y_{p1}). The intersection of this line with the top view projection of the path gives (x_{p2}, y_{p2}) of the point P_{p2}.
Solving equation 6.26 for z and substituting z in equation 6.24 and 6.25 gives the three-dimensional coordinates of the point P.

$$x = \frac{x_{p1} x_{p2} d_2}{a x_{p1} + y_{p1} x_{p2}} \tag{6.31}$$

$$y = \frac{y_{p1} x_{p2} d_2}{a x_{p1} + y_{p1} x_{p2}} \tag{6.32}$$

$$z = \frac{a x_{p2} d_2}{a x_{p1} + y_{p1} x_{p2}} - d_1 \tag{6.33}$$

This procedure is done with all points of the front view path.
Finally, the *(x, y, z)* coordinates are converted to a coordinate system *(x', y', z').* This coordinate system, that is used for theoretical calculations, has its origin at the eyelet point and the z'-axis is the package axis.

$$x' = -(z - z_e) \tag{6.34}$$

$$y' = y - y_e \tag{6.35}$$

$$z' = -(x - x_e) \tag{6.36}$$

z_e is calculated from equations 6.11 and 6.18. Then it's possible to get x_e and y_e from the image coordinates of the eyelet, using equations 6.24 and 6.25.

6.4.5 Geometry points

To specify the geometry of the projected image, the user can set *geometry points* by marking then with the mouse pointer. Three points are sufficient to get all required values, illustrated in Figure 6-15

- Distance package - eyelet for equation 6.18 and 6.19 and as unwinding parameter
  $l_p = x_1 - x_2$
- Eyelet position for equation 1.34 - 1.36
  $x_{pe} = x_{p1}$, $y_{pe} = y_{p1}$
- Package length and radius as unwinding parameters
  $plen_p = x_{p2} - x_{p3}$
  $pdiamp = y_{p2} - y_{p3}$

The actual value of the distance package- eyelet, package length and radius is calculated from equation 6.10

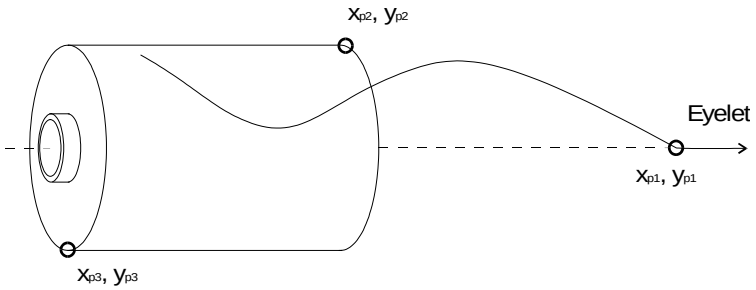


Figure 6-15. Location of geometry points

7. Strobe Control

The stroboscopes are used to freeze the balloon movement by flashing at every trigger. It is also used to brighten up the yarn at the capturing point. The strobes have an external trigger input. It flashes at the raising edge of the external trigger signal. Because both DS/2 and strobe use TTL levels, the external trigger input can be connected directly to the DS/2 digital output 0. To trigger a flash, the DSP just has to set the digital output shortly to “high” and then back to “low”.

7.1 DSP Control

Figure 7-1 shows the block diagram of the generation of this trigger pulse.

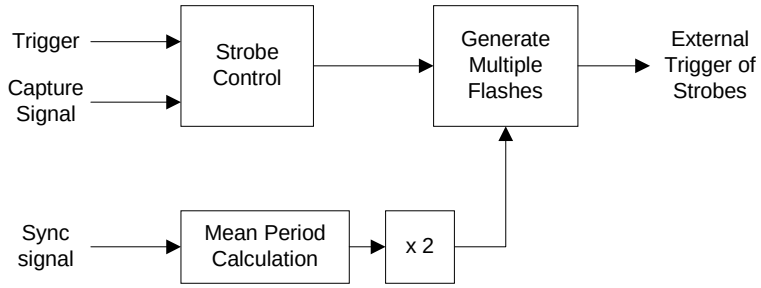


Figure 7-1. Block diagram of strobe triggering

The strobe control provides two function modes:

- Flash continuously at every trigger. This is used to freeze the balloon to see it with the naked eye.
- Flash once at the capturing to brighten up the image

Generating a bunch of following strobe flashes instead of one allows to see the development of the yarn balloon. This function block follows the strobe control and is optional. The number of flashes and the distance between the flashes is adjustable. The distance is given in degree, so the delay time has to be calculated from the delay angle and the current revolution period:

$$t = \frac{(RevolutionPeriod) \cdot (DelayAngle)}{360\ deg}$$

The revolution period detection is equivalent to the one for the trigger delay, see Chapter 5.1.

7.2 User Interface

The user has the following settings:

- Switch “flash continuously at trigger” on and off
- Set the number of multiple flashes
- Set the distance between multiple flashes in degree

8. Speed Measurement

8.1 Speed Sensor

A speed sensor is already built in the yarn drive. It is used to calculate the current speed of the drive and display it on its front panel.
Now, the value of the speed is interfaced to the PC to store it together with the other acquired data. The sensor of the yarn drive is used to get the speed data. It consists of a toothed wheel and a light barrier. The toothed wheel is connected to the wheel that drives the thread. The light beam is interrupted at each passing of a tooth. The output signal of the speed sensor is a square wave with a frequency proportional to the speed. The amplitude of the signal is *23.5V*. Figure 8-1 shows the sensor.

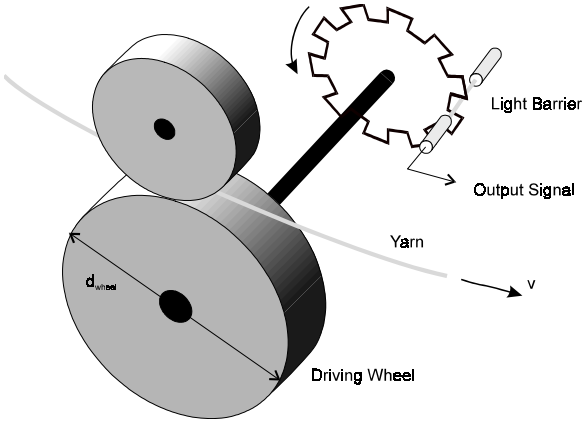


Figure 8-1. Speed sensor

8.2 Interfacing Electronics

A digital input of the DS/2 board is used to interface the speed sensor. Because all digital inputs of the DS/2 use TTL levels, the *23.5V* has to be converted to *5V*. This job is done by a simple amplifier circuit shown in Figure 8-2. Because a gain < 1 forces the use of an inverting amplifier, a second inverting amplifier with gain = 1 is used to get finally the uninverted signal.

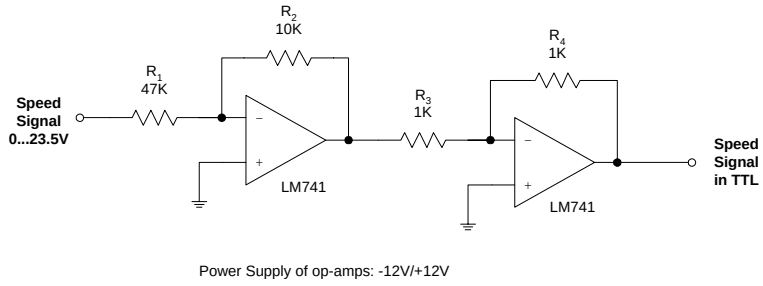


Figure 8-2. Speed signal interfacing circuit

The gain of the amplifier is $\frac{R_1}{R_2} = \frac{5V}{23.5V}$. The resistor values $R_1 = 47K$ and $R_2 = 10K$ exactly match this gain.

8.3 DSP Processing and Speed Calculation

As at the trigger signal, there is no possibility to interrupt the DSP at each pulse of the speed signal. Thus, the DSP software has to poll the speed signal continuously to detect the pulses. At each raising edge, a function is called that determines the period of the speed signal and calculates the speed from it, see Figure 8-3.

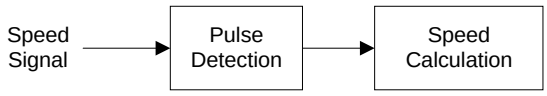


Figure 8-3. Function block diagram of speed signal processing

The period of the speed signal is calculated from the difference of the current system clock and the system clock of the previous pulse. In fact, only every hundredth pulse is taken to get a higher accuracy.

The rotation frequency of the driving wheel is then

$f = \frac{1}{T_{SP} \cdot n_{Cogs}}$; T_{SP}: Speed signal period; n_{Cogs}: Number of cogs

The speed of the yarn is

$v = l_{wheel} \cdot f = \frac{\pi \cdot d_{wheel}}{T_{SP} \cdot n_{Cogs}}$; l_{wheel}, d_{wheel}: Circumference and diameter of driving wheel

8.4 User Interface

The User Interface consists of a display that shows the current speed.

9. System Integration

9.1 Integration of DSP Part

All DSP signal processing and acquisition functions are integrated into one program. The resources provided by the DSP are divided up to the following functions:

- Timer 0 interrupt: Used as a basic system-timer. Counts the system clock that is used for various timing calculations. Used to produce the trigger delay and multiple strobe flashes. It runs with the frequency of 20000 Hz.
- Timer 1 interrupt: Used for sampling, filtering, logging of the tension and detecting the tension maximums/minims. The frequency is 2000 Hz.
- Main loop: The main loop does nothing else than polling sync pulse and speed signal continuously and detect raising edges.
- Interrupt by PC: Used for triggering DSP functions and communication (see Chapter 0)

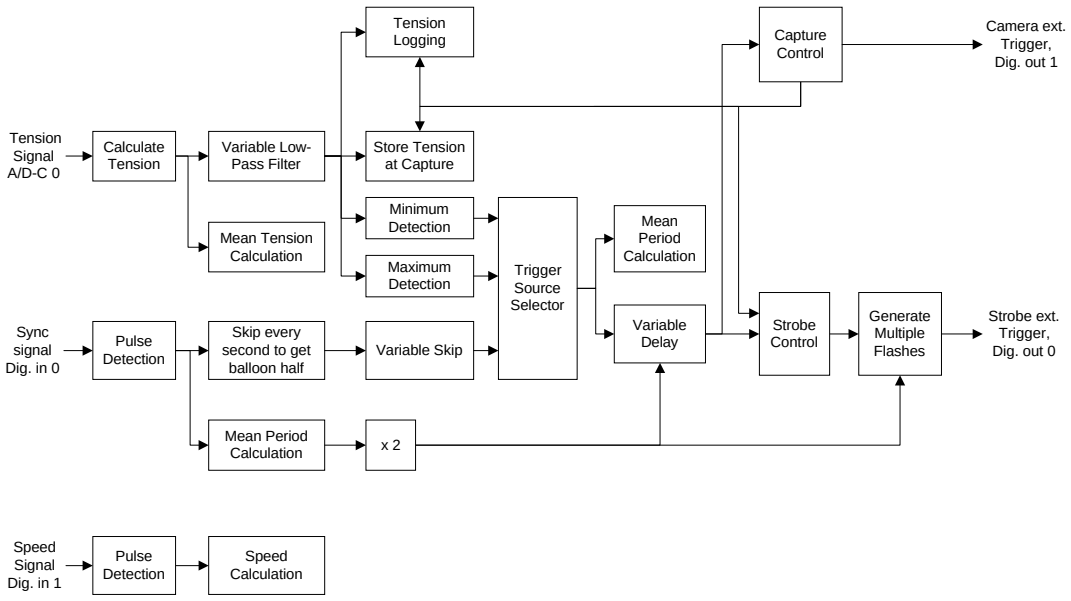


Figure 9-1. Function block diagram of DSP program

The input/output channels are divided up in the following manner:

- D2/2 board:
  Digital input 0: Sync signal
  Digital input 1: Speed signal
  Digital output 0: Strobe triggering
  Digital output 1: Triggering of the video camera
- DSP:

A/D-C 0: Tension signal

Because the A/D converter of the DSP board is more accurate and contains an input filter, this one is used and not the DS/2 converter. The DS/2 I/O board is only used for digital input/output what's not possible by the DSP board.

Figure 9-1 shows the function block diagram of the complete DSP program.

9.2 Communication PC-DSP

To set the parameters of the DSP functions and to get the acquired data, communication between PC program and DSP program is necessary. Two features of the DSP board are used for communication. Firstly the DSP memory can be accessed by the PC, and secondly the PC has the ability to interrupt the DSP program. Figure 9-2 shows the structure of the communication.
There are two layers. The basic communication layer is application independent. It is used by the Unwinding Analyzer specific layer, that contains all specific DSP functions.

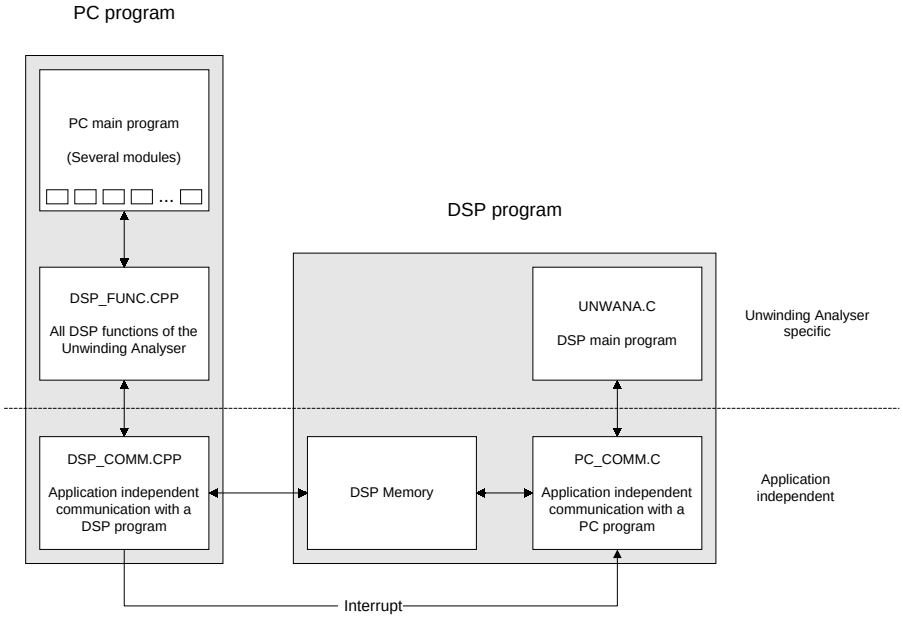


Figure 9-2: PC-DSP communication structure

The basic procedure for a execution of a DSP function is:

- One module of the PC main program calls a function in `DSP_FUNC.CPP`, for example `GetTension()`
- `DSP_FUNC.CPP` now tells `DSP_COMM.CPP` that a DSP function shall be executed and gives function number and parameters
- `DSP_COMM.CPP` puts function number and parameters into the DSP Memory and interrupts the DSP to trigger the function execution

- PC_COMM.C takes function number and parameters from the memory and tells the main program UNWANA.C to execute the function
- UNWANA.C executes the function and returns a reply, PC_COMM.C puts the reply into the DSP Memory and acknowledges the command execution
- C30_COMM.CPP has waited for the acknowledgment and takes the reply and returns it to DSP_FUNC.CPP
- Finally, DSP_FUNC.CPP returns from GetTension() with the reply

Figure 9-3 shows how the DSP memory is used to exchange data. One section is used for communication, realized in the global array CommMem[] in the DSP program. To know where this array is located in the memory, the linker is told to have one fixed memory location, COMM_MEM_POINTER, that points to CommMem[].

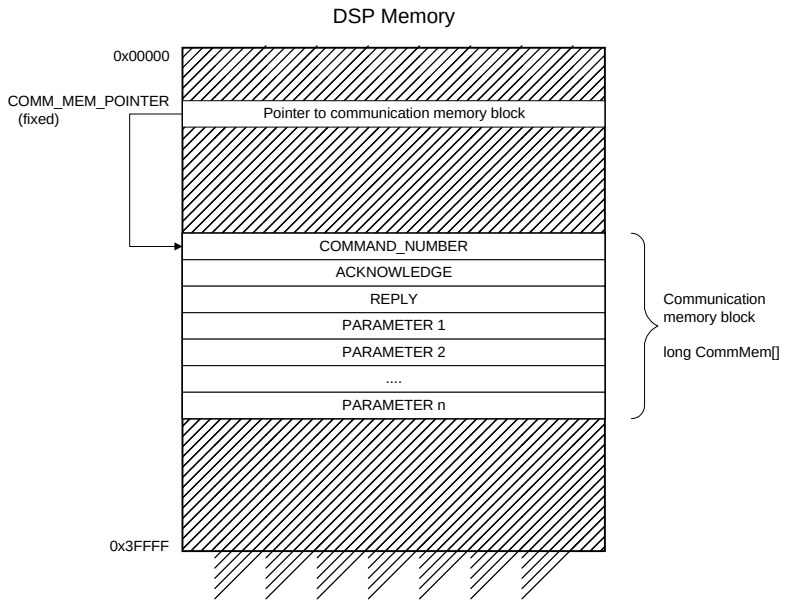


Figure 9-3. DSP memory used for communication

Figure 9-4 illustrates the procedure of a command execution at the example of the function DSP_GetTension(MEAN_TENSION), which returns the current mean tension.

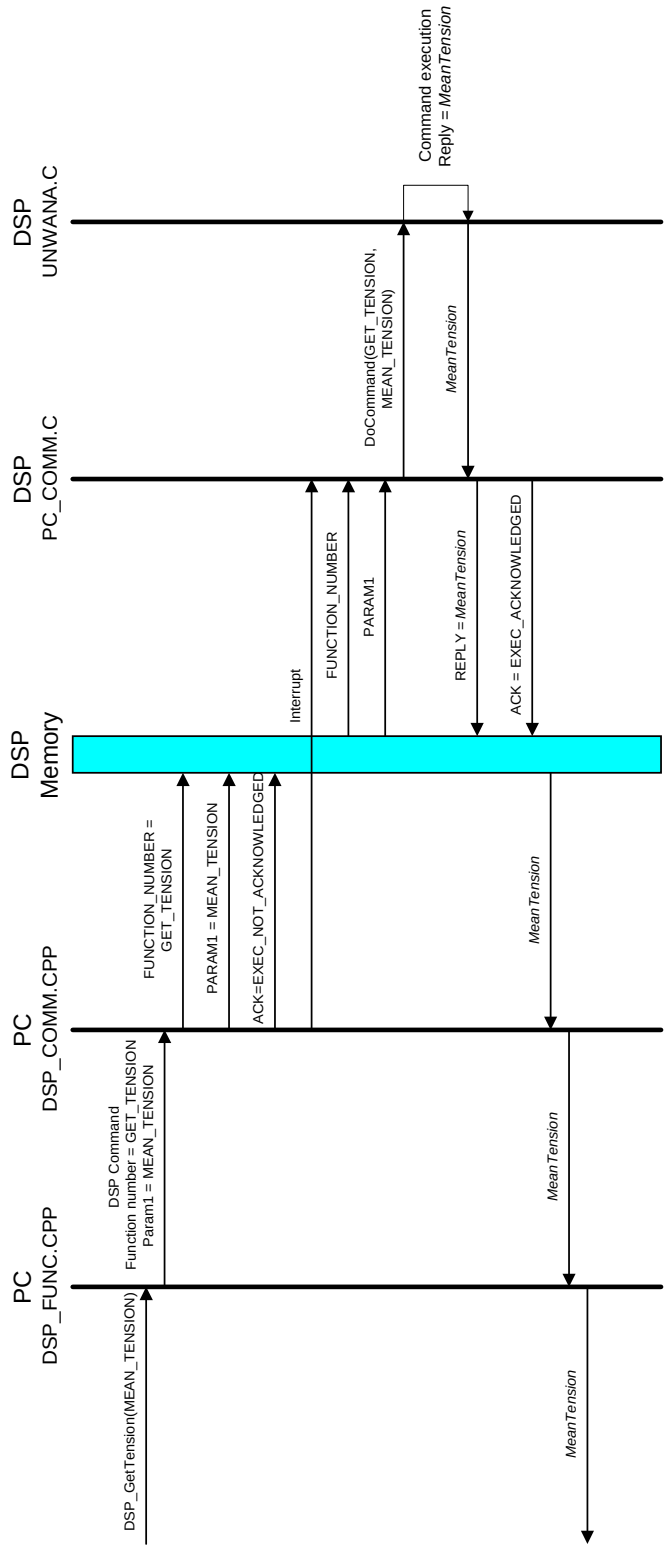


Figure 9-4. Procedure of a command execution

9.3 User Interface

All parts of the user interface are integrated now in one user surface. It consists of a menu bar, dialog boxes and the desktop. On the desktop are placed:

- Captured image
- Tension graph
- A display and tool-box

The display and tool-box contains continuously updated displays of unwinding speed, revolution rate, trigger rate, mean tension and tension at capture. Additionally, there some functions of the main menu as buttons for quick execution. Figure 9-5 shows the desktop.

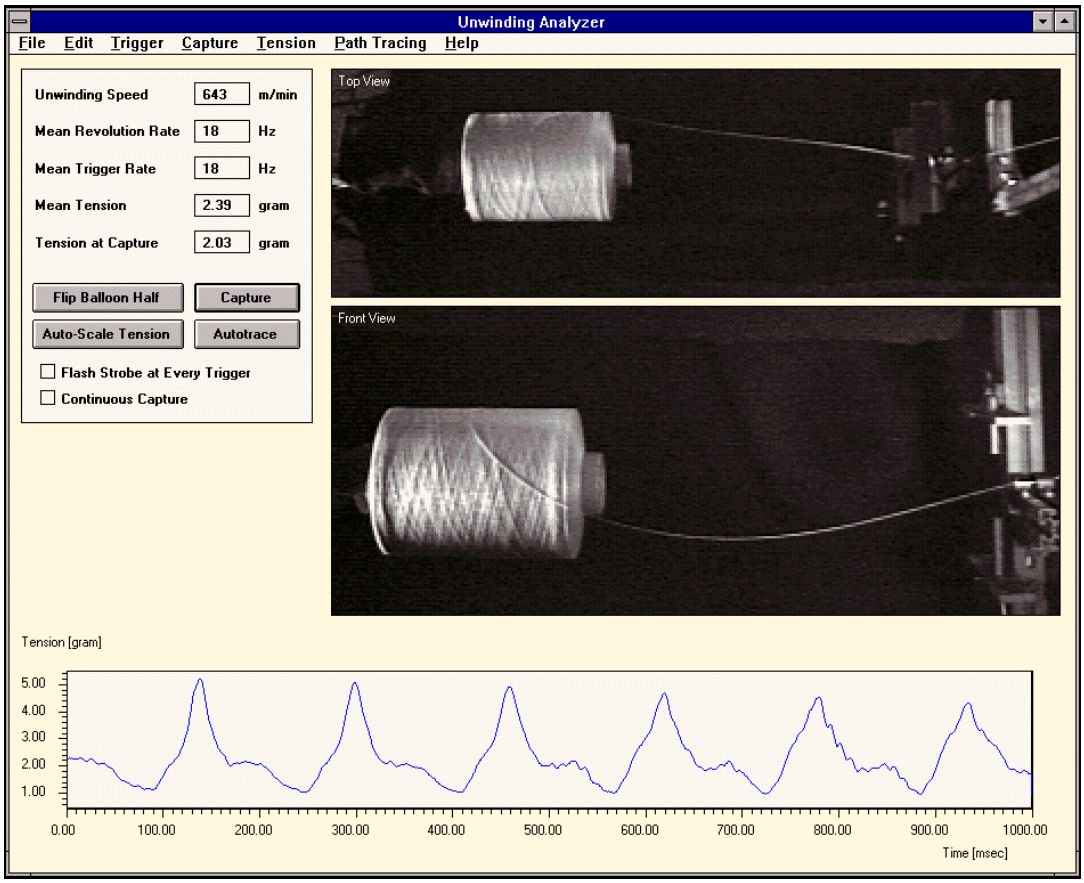


Figure 9-5. Unwinding Analyzer Desktop

10. Experiments

10.1 Tension Measurements

Figure 10-1 shows an example tension graph and its fourier transformation. The tension curve is almost periodic. The fourier transformation shows three main frequencies. One frequency, about 18 Hz, is close to the rotation frequency, indicated by the dotted line.

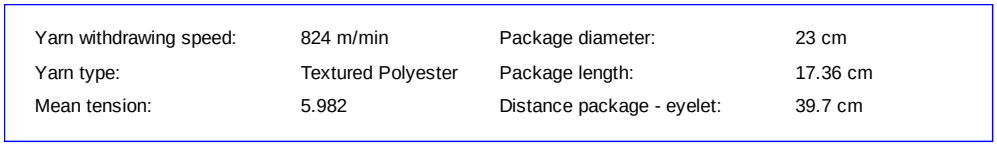


Figure 10-1. Tension curve and fourier transformation

For the used package, it takes about 2.5 revolutions until the unwinding point comes back to the same position. The frequency of the unwinding point moving up and down the package is therefore 18 Hz / 2.5 = 7.2 Hz, which matches the biggest peak of the FFT. There is also another peak at the double frequency, 14.4 Hz.
Hence, the tension changes are synchronous with the unwinding point moving up and down the package.
The biggest peak at 0 Hz represents the mean tension.

10.2 3D Yarn Paths

Experiments showed that a very accurate alignment of the whole system is necessary to get good results. There are many sources of errors, especially the alignment of the mirror and the camera is very important. The axis of the camera has to be exactly 90 degrees to the package axis. The mirror must be mounted precisely 45 degrees to the camera axis. Some errors are also made due to the limited image resolution.
To fulfill these requirements is pretty difficult, therefore the systems doesn't give an accuracy higher than 0.5 cm. But the qualitative balloon shape is more important than the quantitative.

Figure 10-2 shows an example capture, unsynchronized at a random time. The dotted line in the tension curve indicates the capture time. The top view part in the image is vertically flipped by the Matlab program, so it looks like as it would have been taken by a camera at the top. Of course the plot of the reconstructed 3D path can't provide the 3D information. In the Matlab program, however, it is possible to rotate the plot to get a better idea of the yarn curve.

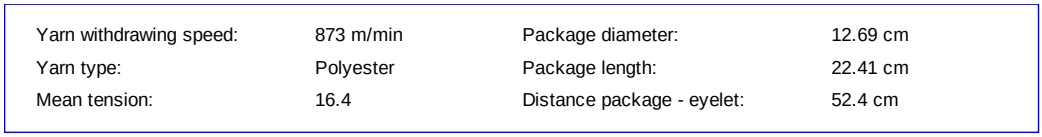


Figure 10-2. Random capture

Figure 10-3 contains the acquired data of a capture at maximum tension. The circles in the tension graph indicate the maximums. It is a problem in this case, that a part of the yarn path is hidden by the package in the top view. The only possibility right now is to

use the best guess and set the path points manually. For the future, the path reconstruction algorithm could be extended to use only the front view and additionally the package geometry for reconstruction.
The 3D path shows clearly what is expected from theory for maximum tension. The threads comes from the very back of the package, forming a long balloon.

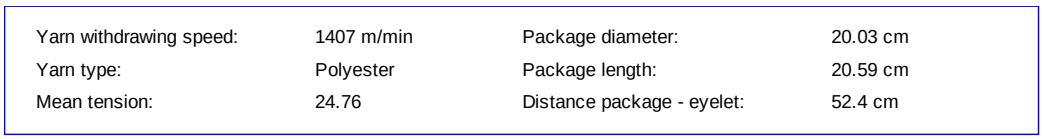


Figure 10-3. Capture at maximum tension

A capture at minimum tension shows the opposite situation, in Figure 10-3. The yarn leaves the package near the edge close to the eyelet.

Yarn withdrawing speed:	1966 m/min	Package diameter:	19.94 cm
Yarn type:	Polyester	Package length:	21.15 cm
Mean tension:	43.69	Distance package - eyelet:	52.4 cm

Figure 10-4. Capture at minimum tension

11. Conclusions

Previously, it was impossible to observe the unwinding process of yarn from a package. The Unwinding Analyzer provides a comprehensive functionality to examine the yarn dynamics. The system determines the whole three-dimensional yarn path, even on the package surface, and the tension curve. Synchronization functions allow to capture the balloon at the most interesting positions.

An easy to handle graphical user interface, the automatic speed measurement and the automatic path tracing help to get quick results. Hence, many experiments can be done in a short time. The system is integrated but open, export functions provide the acquired data to Matlab's powerful processing functions.

Some theoretical considerations were already confirmed by the experiments. The system will help to get an further insight of the behavior of the thread at unwinding, to support the development of theory. Later, the theoretical and experimental results will be compared.

Continued research focuses on improving the system's accuracy, evaluating the experimental results and extending the systems capabilities for general use in textile industry.

References

[1] K. Hall, An Experimental Study of Ballooning Yarn, Clemson University Thesis, May 1996

[2] C. Ramsey, Construction and Implementation of a Sensor to Determine the Position of Yarn on a Package, 1995

[3] W. Fraser, T. Ghosh, S. Batra, On unwinding yarn from a cylindrical package, 1991

[4] E. Hecht, Optics, Addison-Wesley, 1990

[5] L. Phillips, H. Nagle, Digital Control System Analysis and Design, Prentice-Hall, 1990

[6] D. Lancaster, TTL Cookbook, Howard W. Sams & Co. Inc., 1981

[7] TMS0320C30 board manual, Spectrum Signal Processing Inc., 1992

[8] DS/2 board manual, Spectrum Signal Processing Inc., 1992

[9] J. Conger, The Waite Group's Windows API bible, The Waite Groups Inc., 1992

[10] C. Petzold, Programming Windows, Microsoft Press, 1990

[11] P. Wiklen, D. Honekamp, Data Becker, Windows System Programming, 1991

[12] J. Foley, A. van Dam, S. Feiner and J. Hughes, Computer Graphics, Principles and Practise, Addison-Wesley, 1990

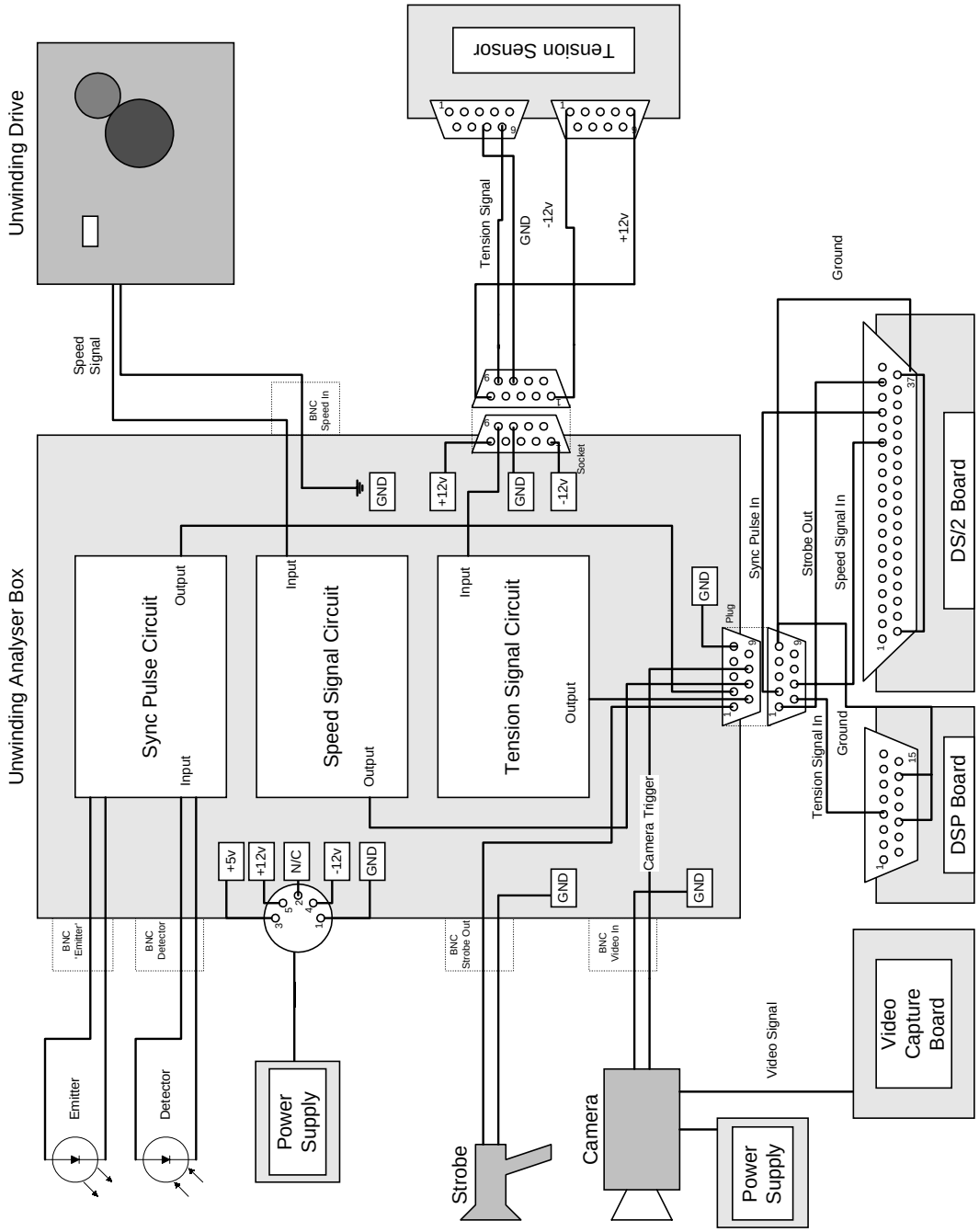
[13] TM-720VM CCD Black&White Camera User Manual, Pulnix America Inc., 1994

[14] Optical Characteristics of CCTV, Pulnix America Inc., 1994

[15] Kernighan and Ritchie, The C Programming Language, Prentice-Hall, 1988

Appendix

A Connection Diagram of the System



B Matlab Programs

The following Matlab programs are used to process the data acquired by the Windows program:

`process.m`	Does a complete processing of all data acquired. Uses the following Matlab programs.
`reconstr.m`	Reconstructs the three-dimensional yarn path from the two projections.
`intrsect.m`	Determines the intersection of two lines.
`splpath.m`	Uses splines to create additional path points.
`readbmp.m`	Reads bitmap file into matrix.

The following Matlab programs are used to display the data:

`prn2.m`	Displays the complete data of a two-dimensional capture
`prn3.m`	Displays the complete data of a three-dimensional capture
`prnt.m`	Displays tension and FFT of tension

They use the functions of the following programs:

`prntens.m`	Displays tension curve
`prnfft.m`	Displays FFT of tension
`prnhead2.m`	Displays unwinding parameters
`prnimg.m`	Displays the image of the yarn path
`prnpath.m`	Displays the two-dimensional projection of the yarn path
`prnpath3.m`	Displays the three-dimensional reconstructed path
`dispimg.m`	Displays an image
`cyclind.m`	Draws a cylinder (used to draw the package)
`setview.m`	Adds buttons to graph to change the viewing angle

C DSP Programs

The program running on the digital signal processor consists of the following modules:

`30unwana.c`	Main program
`c30.c`	Library to access the hardware of the DSP board
`pc_comm.c`	Functions to communicate with the PC
`pc_c30.h`	Defines common symbols of the DSP program and the Windows program. This file is included in both DSP and Windows program

D C++ Windows-Programs

The Unwinding Analyzer Windows program contains the following modules:

`main.cpp`	Main module, does initialization, manages main window
`mouse.cpp`	Processes mouse clicks and movements
`menu.cpp`	Contains all functions of the main menu
`dialog.cpp`	Contains all dialog boxes
`capture.cpp`	Functions to capture image and tension simultaneously
`tension.cpp`	Functions to handle the tension curve
`path.cpp`	Yarn path editing and autotracing
`export.cpp`	Export of all data to Matlab
`videocap.cpp`	Functions to access the video capture board
`image.cpp`	Functions to handle bitmap images
`bitmap.cpp`	A class to handle Windows bitmap
`c30_func.cpp`	All functions of the DSP program

The following modules are used as well, but they are application independent and can be used for other applications:

`c30_comm.cpp`	Does the communication with the DSP
`points.cpp`	Class to handle lists of points
`wingraph.cpp`	Class to draw graphs in Windows
`graph.cpp`	System independent class to draw graphs
`settings.cpp`	Loads and stores settings in .INI - files
`util.cpp`	Some utility functions for Windows