With support of NSF Award no. EEC-0754741

ENABLING FEEDBACK FORCE CONTROL FOR COOPERATIVE TOWING ROBOTS

NSF Summer Undergraduate Fellowship in Sensor Technologies
Clarence E. Agbi (Electrical Engineering) – Yale University
Advisors: Dr. Vijay Kumar and Jim Keller

ABSTRACT

Present research looks at the manipulation of a payload using multiple cooperative robots to drag it with cables to an assigned position within a given degree of accuracy. This project will increase the autonomy of simple tasks such as towing and will improve the efficiency of such systems. Moreover, these results will be applied towards improving cooperative technology and incorporated into the research of Jonathan Fink, et al [1] at the GRASP Lab. The project endeavors to accurately measure cable tensions between the payload and multiple cooperative robots in real time, transmitting this information to the user. We use cost-effective load cell sensors to measure tensile forces, outputting a change in frequency in proportion to the change in tension on the cell. Next, these signals are routed into the I/O pins of the BASIC Stamp 2 which wirelessly communicate this data to another BASIC Stamp 2 using an RF transmitter and receiver at 433.92 MHz. Our preliminary results show that we can wirelessly transmit information to a user, thus enabling crucial feedback to this system. This feedback would allow the robots to correct their position and velocity to maintain tension, allowing for less positional error in navigating the payload.

1. INTRODUCTION

As an emerging field, robotics takes an interdisciplinary approach towards efficient automation. Systems such as the robotic arm integrate mechatronics and artificial intelligence to maximize efficiency of simple tasks at minimum costs. Present cutting-edge research attempts this feat using cooperative robotic manipulation, allowing for multiple robotic systems to manipulate or move the same object. This combines the social theory of cooperation practiced in nature by animals (e.g. birds, bees, humans) with a mechanized task or process.

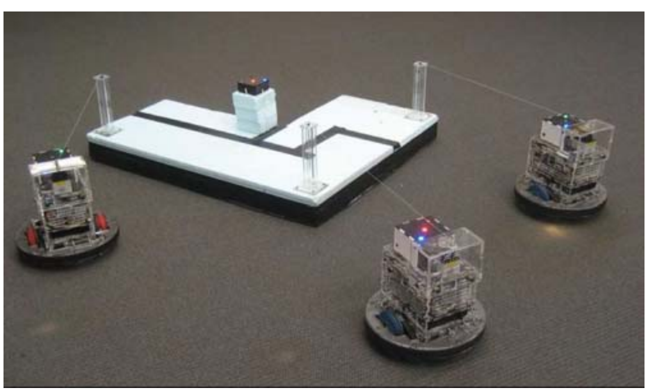

Figure 1: Robots cooperatively tow a payload using cables. This system employs no feedback of cable tension, causing some cables to go slack in the process.

While cooperative robotics is not a new idea, it has only recently been applied to tasks such as towing, a process that currently requires human input. The lack of studies in this important manipulation process has led to our research in cooperative towing [1]. We are investigating the use of multiple robots to maximize the accuracy and efficiency of towing large objects while minimizing positional error and human input. Specifically, we experiment with feedback control to improve our current system of cooperative towing.

This system, as shown in Figure 1, currently uses three automated ‘tugboats’ or robots to pull a heavier payload using a thin cable. As the robots tow the payload, they exert tensile forces on the attached cables causing the payload to move against opposing frictional forces. Sometimes, however, some of the cables go slack, causing the payload to move independently of the robots and increasing positional error of the payload. Thus, feedback control becomes very important in driving down this error and increasing efficiency. Our current stage of research seeks to implement feedback of the cable tensions into the system to monitor slackness, and to force robots to correct the cable tension. Consequently, the cables experience slack less frequently which means increased towing efficiency.

This paper is organized as follows: Section 2 discusses the background and basis for this formative work in cooperative towing, using prior research in the robotics and automation field. Sections 3 and 4 address the design aspects of the feedback system, focusing on the load cell and the BASIC Stamp respectively. Section 5 illustrates preliminary experimental results of the feedback system, and uses data analysis to discuss the consequences of these results. Section 6 details recommendations for future work, and Section 7 acknowledges those entities that made this research possible. Finally, Section 8 lists references on which this research is based.

2. DEVELOPMENT OF COOPERATIVE TOWING

2.1 Cooperative Robotics

Cooperative robotics uses multiple robots, working in coordination through sensory perception or explicit communication [2], to accomplish a set task. Thus, there are several advantages to cooperative robotics over using a single robot to accomplish the same task [2]. Inherent to cooperative robots is the ability to be flexible and adapt to different situations. Whereas some systems rely on one leader and many followers, cooperation suggests a decentralized system where work is equally shared among all systems or robots [2, 3]. Likewise, it also means that the failure of one robot does not result in the disruption of the entire system, since the remaining functional robots can adjust and adapt to the change in environment. The assumption is that cooperative robots are knowledgeable of their position and movement in real time as well as their neighbors’ position and movement in real time [2].

2.2 Applying Cooperative Robotics to Simple Tasks

Prior research in the area looked at the use of robots with arms or casters to carry the object, [3] eliminating the problem of slack cable. An improved version of this research was later proposed where no force sensors were used, but rather looked at error estimation [4]. Other studies looked at fostering a decentralized system by switching the roles of leader among the robots and examining the way the robots communicate with each other [2]. Later research ventured into cooperative towing using cables, though the authors acknowledged the shortcomings in control and suggested feedback as a future improvement [1]. Based on these findings, we propose to implement a feedback system using microcontrollers and force sensors to measure the tension and enhance the performance (dynamic response and accuracy) of the system.

3. LOAD CELL TECHNOLOGY

An integral component of our research is to measure the tensile forces between the payload and the towing robots. Good force measurements require very precise load cells or force sensors that provide consistent readings in spite of slight changes in condition such as temperature. Additionally, we require miniature sensors that will not add weight to the payload and can measure maximum forces of 10lbs as determined by the dry friction between the payload and the floor. Consequently, only a handful of load cells or force sensors could be considered for this project.

3.1 Strain Gauge Load Cells

Most load cells use precise strain gauges to measure strain, which is directly proportional to force. However, strain gauges are very sensitive pieces of equipment and expensive to manufacture. To cut manufacturing and packaging costs, we endeavored to produce a cost-effective strain gauge load cell. This proved to be difficult, though, since factors such as sensitivity to environmental conditions, amplification of small strains, and bonding gauges to appropriate metals could not be easily controlled.

3.2 Capacitive Load Cells

In our quest for better force sensors, we look to the iLoad Mini produced by LoadStar and recently adapted by Parallax, Inc. Shown in Figure 2, this load cell uses breakthrough capacitive technology for load sensing, which harnesses changes in capacitance to measure loads quickly and accurately [5]. The iLoad Mini is the smallest load cell made by LoadStar with a diameter of 1.25 inches. Under applied loads, it outputs a 5V Transitor-Transitor Logic (TTL) square wave whose frequency is proportional to compression or tensile forces associated to the load.

Figure 2: The iLoad Mini

LoadStar also claims 2% full scale accuracy for hysteresis, linearity, and repeatability as well as environmental compensation from 5°C to 40°C [5]. Using this device, we could minimize errors due to hysteresis and thermal conditions while still being cost effective and precise in measurement. Lastly, the iLoad Mini is relatively inexpensive compared to other strain gauge load cells from competing companies. It features a starting price of $99.

3.3 Implementing the Capacitive Load Cells

Next, we integrate the load cells into the overall design of the feedback loop. Each load cell is governed by the set of equations below, relating the changes in frequency output to the applied load. They operate at 5V DC and require a bit-operated input control pin, CTRL. When CTRL is set high (1), the frequency pin outputs Cx which refers to the load-related frequency. Otherwise, a low input (0) to CTRL will cause the frequency pin to output Cref, a reference frequency used for environmental compensation. Because all three load cells do not behave exactly the same, they have different constants (K, qA, qB, qC) to calibrate for minor differences. Table 1 show these different constant values provided by Parallax for each sensor as identified in the governing equations.

Equations 1-3, below, describe the set of equations that relate applied loads to frequencies Cx and Cref. The first equation uses Cref, Cx, and K to produce a corrected capacitance, Cc, accounting for environmental conditions. This allows for a more accurate reading in case of environmental changes, slight or otherwise. The quadratic form of Equation 2 describes the non-linear behavior of Cc as it relates to the load. The negative sign calibrates tension measurements from negative to positive. Finally, Equation 3 calibrates all weights in relation to a zero load or tare weight. This produces a precise net weight that should correspond with the actual weight.

$$C_c = 5E6 - \left[10\left(C_X - K \cdot C_{ref}\right)\right] \quad (1)$$

$$\text{Load} = -\left(qA \cdot C_c{}^2 + qB \cdot C_c + qC\right) \quad (2)$$

$$\text{Weight} = \text{Load}_{x\ \text{lbs}} - \text{Load}_{0\ \text{lbs}} \quad (3)$$

Table 1: These are calibrated constants for each sensor as it applies to the equations above, and the reference frequency (Cref) used for environmental compensation as well as the frequency proportional to the load (Cx).

Label	Sensor 1	Sensor 2	Sensor 3
Serial Number	M081700881	M081700883	M081700887
K	8.254556E-01	5.291063E-01	6.676898E-01
qA	-1.627213E-12	-1.355136E-12	-1.308366E-12
qB	3.275433E-05	2.780857E-05	2.794689E-05
qC	-1.184411E+02	-9.107356E+01	-9.726224E+01
Cref (Hz)	228,540	239,340	235,670
Cx, 0lbs (Hz)	214,120	218,450	217,920

4 PARALLAX BASIC STAMP

Figure 3: The BASIC Stamp 2 connected to the Parallax 433 MHz RF Receiver

Next, we devise a feedback system to read, process, and send the data collected from the cable tensions to a user interface, using the BASIC Stamp 2, manufactured by Parallax. The BASIC Stamp 2 offers compatibility with other Parallax devices such as the wireless transmitter and receiver, reliable support/documentation, and a simple programming language, PBASIC 2.5.

4.1 Wireless Communication

To maintain the autonomy of the payload, we use wireless communication to send the tension data to the user PC, which controls the tugboats/robots. This wireless communication consists of the Parallax 433 MHz RF Transmitter and Receiver shown in Figure 3, which can easily be accessed and programmed by a BASIC Stamp 2. These modules use Linx RF chips, operating on 5V DC and high baud rates of 12k-19.2k. It also has a communication range of a couple hundred feet which is sufficient for this particular setup.

Lastly, in keeping with the cost-effective nature of this project, the RF transmitter and receiver are relatively inexpensive.

Each RF module is easily integrated, connecting to Pin 15 of the BASIC Stamp 2 and using two very simple commands in PBASIC 2.5: SEROUT (to send information) and SERIN (to receive information). The transmitter pulses at 2.4ms to sync with the receiver, first sending characters to denote the start of transmission and then sending data one byte at a time.

4.2 Shortcomings of the Basic Stamp 2, Revision J

4.2.1 Limited Processing and Memory Constraints

One shortcoming is that the BASIC Stamp 2 is a single-threaded microprocessor, meaning it can execute only one instruction at a time. As a result, we cannot interrupt the main program to count the frequencies of three different load cells simultaneously. Therefore, frequencies are counted one at a time for 100 milliseconds (ms), meaning it would take a minimum of 300 ms to just count frequencies on all three load cells. Moreover, the instruction set is executed at 4000 instructions per second, which is approximately 1 line of code every 250 microseconds (us). Assuming an additional 120 lines of code executed at 4000 instructions/sec, the entire program would run a total of 330 ms, looping about 3 times per second.

For the purposes of our experiment, updating the sensors three times per second is still an adequate sample rate for the measurements to be considered current and up-to-date. While counting less than 100ms would increase the amount of time the sensors are updated, the information would likely be inaccurate because the counted frequencies would return less significant digits. Alternatively, counting for more time would allow for very precise but out-of-date measurements. Our current setting of 100ms, therefore, is optimal for this experiment. This also means minimal coding is crucial to faster update rates.

Moreover, the BASIC Stamp 2 has limited memory which holds a maximum of 26 one-byte variables. This is a constraint that is difficult to navigate around because we use word-sized variables (equivalent to 2 bytes) to maintain 5 significant digits. Limited to only 13 word variables, our code cuts corners by using many constants and reusing word-sized variables.

4.2.2 Aliasing in the I/O pins

Another major shortcoming, specific to earlier revisions of the BASIC Stamp 2, is the inability to accurately count frequencies higher than 120 kHz even though the Stamp runs at a clock speed of 20 MHz. This presents a problem because the load cells output a maximum frequency (Cx and Cref) almost twice that of 120 kHz. One solution around the problem is to use IC prescalers to divide down the load cell frequencies enough to be counted. However, space and power constraints on both the proto-board and the prescalers make it very difficult to integrate.

Consequently, we test the effects of reading frequencies higher than 120 kHz through the I/O pins using a function generator, outputting a 5V DC square wave at 50% duty cycle with varying frequencies. Table 2 shows the results of this test. From 0 – 123 kHz, the output frequency follows the actual input frequency as Parallax claims. From 124 – 126 kHz, the output frequency fluctuates a lot since the input is too fast to be accurately counted. From 126 – 250 kHz, Equation 4 models the actual frequency while at frequencies higher than 250 kHz, Equation 5 models the actual frequency, where N is the quotient of Output Freq/250 kHz.

$$\text{Output Freq (kHz)} - 250\text{ kHz} = \text{Actual Freq (kHz)} \quad (4)$$
$$\text{Output Freq (kHz)} + N*250\text{ kHz} = \text{Actual Freq (kHz)} \quad (5)$$

This phenomenon is known as aliasing which is an error where the input is faster than the twice the sampling rate. According to Parallax, the BASIC Stamp 2 (Revision J) can see transitions in a pulse width of 4.16 micro-seconds or 2 transitions (one period) in a pulse width of 8.32 micro-seconds. A period width of 8.32 micro-seconds is approximately 120 kHz. However, the load cells operate in the frequency regime of 180-230 kHz, which means that they have smaller periodic widths; consequently, the BASIC stamp will only catch some transitions and not others. From the results in Table 2, though, we can accurately model the operating frequencies of the load cell using Equation 4, accounting for the aliasing. This is reflected in the transmitter source code found in Appendix A.

Actual Freq. (kHz)	Output Freq. (kHz)	Actual Freq. (kHz)	Output Freq. (kHz)
100.2	100.04	129.2	121.04
110.2	110.01	130.2	120.02
120.2	120.02	140.2	110.02
121.2	121.00	150.1	100.11
122.2	122.03	200.1	50.17
123.2	123.02	240.1	10.23
124.3	123.90-124.00	250.3	.14
125.2	123.04-124.27	260.4	10.04
126.2	123.90-124.05	300.5	50.04
127.2	123.01	340.5	90.07
128.2	120.01	373.5	123.02

Table 2: These are the results of inputting a 5VDC square wave at 50% duty cycle into the I/O pins of the BASIC Stamp 2.

5 EXPERIMENTAL RESULTS AND CONCLUSIONS

Preliminary results show wireless communication and readings from the load cell under different weights. Towers were also built to encase the load cells and to ensure proper tension alignment from the load cell to the tugboat. This is illustrated with the side, top, and bottom view of the tower in Figure 4. The tower structure, however, introduced another element of error into the feedback system caused by the dry friction between the cable and the hole at the top of the tower (see Figure 4). As a result, simulated tensions on the load cells using metric weights provide readings with fluctuating errors. For weights greater than or equal to about 500 grams, the load cell provides accurate readings with an error of about 10 percent. Meanwhile, weights less than 500 grams produce inaccurate results with errors greater than 10 percent. Some of the error is due to rounding errors in the program, found in Appendices A and B, since the BASIC Stamp does not process floating numbers. However, a big part of the error is due to the friction between the cable and the hole at the top of the tower; thus, reducing friction becomes an important design challenge.

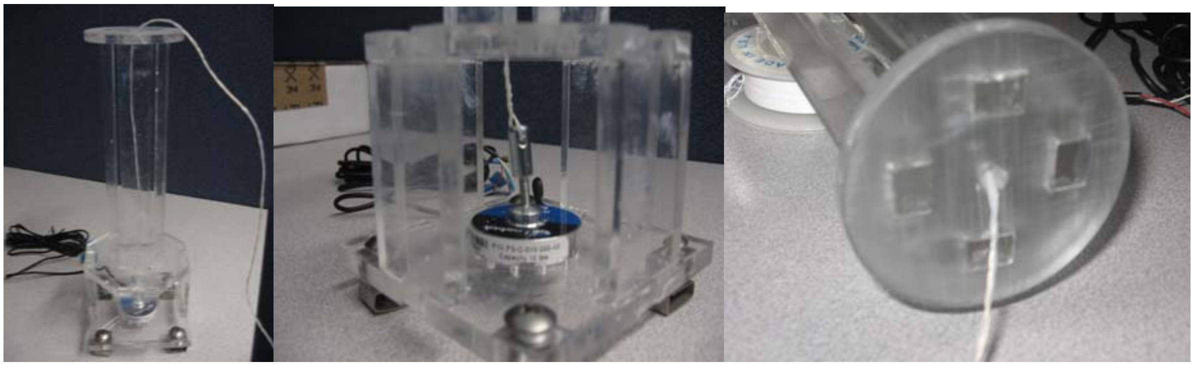

Figure 4: (Left) This tower was constructed to encase the load cell, and replaced existing towers on the payload. (Middle) The bottom view of the tower displays the iLoad Mini connected to the cable-threaded screw, allowing us to measure tension. (Right) The top view of the tower shows the cable coming through the small hole which is a major source of friction the heavier the load becomes.

Using an angler drill-bit, we scraped the sides of the hole and create a cone-shaped hole. We also twisted multiple cables together to increase the load capacity of the cable and to ensure that the Simulations show that this change in design helped, but did not eliminate a lot of the frictional errors. Likewise, adjusting the program code to account for these forces would require coding different error conditions to compensate for the friction. This would increase the program size, and cause the system to update less frequently.

6 FUTURE WORK

To avoid issues surrounding the BASIC Stamp 2, future works should use the BASIC Stamp 2p24 to replace earlier revisions of BASIC Stamp 2, such as revision j. This particular Stamp can easily be replaced in our setup, and has an increased program execution rate of 12,000 instructions per second and can count up to 300 kHz. The downside is that it draws significantly more current and power, and is more expensive than our current stamp.

Further improvements on our design might include testing different load cells such as the expensive strain gauges. These load cells might be less error prone and will eliminate the issue of aliasing and sampling because they output voltages proportional to the load. The only concern with this design would be to read the different voltage inputs. In order to accomplish this task, the voltages would have to be put through external Analog to Digital Converters (ADCs). Depending on space and power constraints, there might need to be a 3 to 1 multiplexer (MUX) with the output attached to an ADC. As a result, there is less space and power taken up.

Friction should also be taken into consideration in future designs. It comprised of a majority of the error in the load cell readings, even though we smoothed much of the surfaces in contact with the cable. Future designs should consider further both the hardware and software aspects of the problem, perhaps calibrating sensor constants to take into account frictional forces on the cable.

Lastly, future work should look into the actual integration of this feedback system into the overall control system for the towing of the robots. There needs to be an interface between the output of the BASIC Stamp 2 and the input for the towing robots, so that the robots can automatically correct their speed and velocity based on the tension data. This crucial implementation would enhance the overall research in cooperative towing.

7 ACKNOWLEDGMENTS

I would like to thank Professor Vijay Kumar and Jim Keller who have afforded me the opportunity to participate in this research. I would also like to thank Soonkyum Kim, with whom I worked closely for the construction and completion of this project, as well as the GRASP Laboratory for allowing me to use their space and equipment. Lastly, I would like to acknowledge and thank Dr. Jan Van der Spiegel and the NSF-funded SUNFEST Program for their research support.

8 REFERENCES

[1] Fink, Jonathan; Cheng, P.; Kumar, V., "Cooperative Towing with Multiple Robots." In ASME International Design Engineering Technical Conferences \& Computers and Information in Engineering Conference.

[2] Chaimowicz, L.; Sugar, T.; Kumar, V.; Campos, M.F.M., "An architecture for tightly coupled multi-robot cooperation," *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference*, vol.3, no., pp. 2992-2997 vol.3, 2001

[3] Kume, Y.; Hirata, Y.; Zhi-Dong Wang; Kosuge, K., "Decentralized control of multiple mobile manipulators handling a single object in coordination," *Intelligent Robots and System, 2002. IEEE/RSJ International Conference*, vol.3, no., pp. 2758-2763 vol.3, 2002

[4] Kume, Y.; Hirata, Y.; Kosuge, K., "Coordinated motion control of multiple mobile manipulators handling a single object without using force/torque sensors," *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference*, vol., no., pp.4077-4082, Oct. 29 2007-Nov. 2 2007

[5] LoadStar Sensors Inc., iLoad Mini Load Cell, http://loadstarsensors.com/iloadmini.html

APPENDIX A: Source Code for Transmitter

```
' Final Transmitter.bs2
' {$STAMP BS2}
' {$PBASIC 2.5}
'
'Final Transmitter with iLoad Mini Test Code (BS2)
'This will track the tensions taken in from the iLoad Mini
'
'Sensor 1: M081700881
'Sensor 2: M081700883
'Sensor 3: M081700887

CTRL1   PIN     6 'Sensor 1
FREQ1   PIN     7
CTRL2   PIN     8 'Sensor 2
FREQ2   PIN     9
CTRL3   PIN     10 'Sensor 3
FREQ3   PIN     11

SEND    PIN     15

qAC     VAR     Word
qBC     VAR     Word
LoadC   VAR     Word
Cc      VAR     Word
Cref    VAR     Word
Cx      VAR     Word
CrefF   VAR     Word
CxF     VAR     Word
Tare1   VAR     Word
Tare2   VAR     Word
Tare3   VAR     Word
Weight  VAR     Word



K1      CON     54097 '.8254556*65536 = 54097
K2      CON     34676 '.5291063*65536 = 34676
K3      CON     43758 '.6676898*65536 = 43758

qA1     CON     10664 '.1627213*65536 = 10664
qA2     CON     8881  '.1355136*65536 = 8881
qA3     CON     8575  '.1308366*65536 = 8575

qB1     CON     21466 '.3275433*65536 = 21466
```

```
qB2     CON     18225 '.2780857*65536 = 18225
qB3     CON     18315 '.2794689*65536 = 18315

qC1     CON     11844 'Decimal 118.44
qC2     CON     9107  'Decimal 91.07
qC3     CON     9726  'Decimal 97.26


Main:

    Tare1 = 0
    Tare2 = 0
    Tare3 = 0

    GOSUB ReadingSensor1
    Tare1 = LoadC-3
    GOSUB ReadingSensor2
    Tare2 = LoadC-3
    GOSUB ReadingSensor3
    Tare3 = LoadC-3

  DO
    GOSUB ReadingSensor1
    GOSUB ReadingSensor2
    GOSUB ReadingSensor3
  LOOP

  END

' (A*65336)**B = A*B

ReadingSensor1:
  HIGH CTRL1
  PAUSE 5
  COUNT FREQ1, 100, CxF
  PAUSE 5

  LOW CTRL1
  PAUSE 5
  COUNT FREQ1, 100, CrefF
  PAUSE 5

  Cx = 25000 - CxF       'Correct the frequency since we can't read above 120 kHz
  Cref = 25000 - CrefF

  Cc = (50000 - Cx + (Cref**K1))       'Cc = X,XXX,X00
```

```
  qAC = (Cc**qA1)**((Cc/100)*66)
  qBC = Cc**qB1

  LoadC = qAC - qBC + qC1          'LoadC = 000X.XX
  Weight = LoadC - Tare1

  DEBUG SDEC LoadC, CR
  DEBUG SDEC Weight, CR

  PULSOUT SEND, 1200                    'Sync pulse for the receiver
  SEROUT SEND, 16468, ["ABC!", Weight.HIGHBYTE, Weight.LOWBYTE]

  RETURN

ReadingSensor2:
  HIGH CTRL2
  PAUSE 5
  COUNT FREQ2, 100, CxF
  PAUSE 5

  LOW CTRL2
  PAUSE 5
  COUNT FREQ2, 100, CrefF
  PAUSE 5

  Cx = 25000 - CxF       'Correct the frequency since we can't read above 120 kHz
  Cref = 25000 - CrefF


   Cc = (50000 - Cx + (Cref**K2))       'Cc = X,XXX,X00
   qAC = (Cc**qA2)**((Cc/100)*66)
   qBC = Cc**qB2

   LoadC = qAC - qBC + qC2          'LoadC is negative for tension so all signs are switched!
   Weight = LoadC-Tare2

  PULSOUT SEND, 1200                    'Sync pulse for the receiver
  SEROUT SEND, 16468, [ "DEF!", Weight.HIGHBYTE, Weight.LOWBYTE]

  RETURN

ReadingSensor3:
  HIGH CTRL3
  PAUSE 5
  COUNT FREQ3, 100, CxF
  PAUSE 5
```

```
  LOW CTRL3
  PAUSE 5
  COUNT FREQ3, 100, CrefF
  PAUSE 5

  Cx = 25000 - CxF       'Correct the frequency since we can't read above 120 kHz
  Cref = 25000 - CrefF

'  DEBUG ? Cx, CR
'  DEBUG ? Cref, CR

   Cc = (50000 - Cx + (Cref**K3))       'Cc = X,XXX,X00
   qAC = (Cc**qA3)**((Cc/100)*66)
   qBC = Cc**qB3

   LoadC = qAC - qBC + qC3          'LoadC is negative for tension so all signs are switched!
   Weight = LoadC-Tare3

  PULSOUT SEND, 1200                    'Sync pulse for the receiver
  SEROUT SEND, 16468, [ "GHI!",Weight.HIGHBYTE, Weight.LOWBYTE]

  RETURN
```

APPENDIX B: Source Code for Receiver

```
'Final Receiver.bs2
'{$STAMP BS2}
'{$PBASIC 2.5}
'
'Final Receiver with iLoad Mini Test Code (BS2)
'This will track the tensions taken in from the iLoad Mini
'
'Sensor 1: M081700881
'Sensor 2: M081700883
'Sensor 3: M081700887

RECEIVE   PIN      15

Load1     VAR      Word
Load2     VAR      Word
Load3     VAR      Word

Main:
  DO
    SERIN RECEIVE, 16468, [WAIT("ABC!"), Load1.HIGHBYTE, Load1.LOWBYTE]
    SERIN RECEIVE, 16468, [WAIT("DEF!"), Load2.HIGHBYTE, Load2.LOWBYTE]
    SERIN RECEIVE, 16468, [WAIT("GHI!"), Load3.HIGHBYTE, Load3.LOWBYTE]

    DEBUG "Sensor 1: ", DEC Load1 DIG 4, DEC Load1 DIG 3, DEC Load1 DIG 2,".",DEC
Load1 DIG 1, DEC Load1 DIG 0," lbs", CR
    DEBUG "Sensor 2: ", DEC Load2 DIG 4, DEC Load2 DIG 3, DEC Load2 DIG 2,".",DEC
Load2 DIG 1, DEC Load2 DIG 0," lbs", CR
    DEBUG "Sensor 3: ", DEC Load3 DIG 4, DEC Load3 DIG 3, DEC Load3 DIG 2,".",DEC
Load3 DIG 1, DEC Load3 DIG 0," lbs", CR
  LOOP
  END
```