Stony Brook University

High-Resolution Gas Metering and Nonintrusive Appliance Load Monitoring System

A Thesis Presented

by

Mahder Tewolde

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Master of Science

in

Mechanical Engineering

Stony Brook University

May 2012

Stony Brook University

The Graduate School

Mahder Tewolde

We, the thesis committee for the above candidate for the

Master of Science degree, hereby recommend

acceptance of this thesis.

Prof. Jon P. Longtin – Thesis Advisor
Mechanical Engineering

John Kincaid – Chairperson of Defense
Professor – Mechanical Engineering

David Hwang – Committee Member
Professor – Mechanical Engineering

This thesis is accepted by the Graduate School

Charles Taber

Interim Dean of the Graduate School

Abstract of the Thesis

High-Resolution Gas Metering and Nonintrusive Appliance Load Monitoring System

By

Mahder Tewolde

Master of Science

in

Mechanical Engineering

Stony Brook University

2012

This thesis deals with design and implementation of a high-resolution metering system for residential natural gas meters. Detailed experimental measurements are performed on the meter to characterize and understand its measurement properties. Results from these experiments are used to develop a simple, fast and accurate technique to non-intrusively monitor the gas consumption of individual appliances in homes by resolving small amounts of gas usage. The technique is applied on an existing meter retrofitted with a module that includes a high-resolution encoder to collect gas flow data and a microprocessor to analyze and identify appliance load profiles. This approach provides a number of appealing features including low cost, easy

installation and integration with automated meter reading (AMR) systems. The application of this method to residential gas meters currently deployed is also given. This is done by performing a load simulation on realistic gas loads with the aim of identifying the necessary parameters that minimize the cost and complexity of the mechanical encoder module. The primary benefits of the system are efficiency analysis, appliance health monitoring and real-time customer feedback of gas usage. Additional benefits of include the ability to detect very small leaks and theft. This system has the potential for wide scale market adoption.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor, Professor Jon Longtin for the opportunity do my graduate studies under his guidance. His extensive knowledge in many areas of engineering has proven invaluable in allowing me to do this work.

I would like to thank Professor David Hwang and Professor John Kincaid for being on my thesis committee. I appreciate their valuable time. I would also like to Joseph Fritch for his assistance in this project.

Finally, I would like to thank my parents and siblings. This would not have been possible without their love and support.

CHAPTER I

INTRODUCTION

Natural gas is a vital component of the world's supply of energy. It is one of the cleanest and most useful of all energy sources with applications in heating buildings, generating electricity, transportation and as a chemical feedstock in the manufacture of many commercial products. It is also the fastest growing energy source in the world [1]. As of 2010, natural gas accounts for 25% of the total energy usage of the U.S, up from only 5% in 1950. Fig. 1-1a shows the history of natural gas supply and consumption over the past 20 years and projections into the future. The Energy Information Administration (EIA) expects a steady increase in natural gas usage going into 2035 [2]. In addition, rapidly improving extraction are unlocking vast storehouses of natural gas from within shale formations [3], which is expected to supply as much as half the natural gas production in North America by 2035 (Fig. 1-2b). Natural gas is the fuel of choice for over 40% of households, totaling nearly 5 trillion ft^3 consumed annually [2]. As a result of continually growing demand, investment in gas transmission and distribution networks is expanding. These factors necessitate fast, accurate and reliable measurement techniques. The current metering infrastructure especially in residential homes, which constitute a major portion of the country's natural gas usage, is inadequate for this task. Developing new and improved ways of monitoring the total energy consumption in homes and how it's broken down by end use represents a significant step towards achieving this goal.

1.1 Motivation and Background for Research

The main uses of natural gas in a residential home include heating, cooking and providing hot water. Residential customers receive gas via a dedicated piping infrastructure from city gate stations. The supply lines are connected to mechanical gas meters, which are owned and operated

by the gas utility provider, to measure the volume of gas delivered. To provide constant and measurable gas pressures, regulators adjust the gas pressure before it enters the gas meter and then into the home. The volume of natural gas measured, typically in cubic feet (ft^3) or cubic meters (m^3), is subsequently converted into number of energy units consumed by the end user. This is done using the average energy per unit volume (calorific value) of the gas, also known as its heating value. Information regarding other thermodynamic property variations of the gas is handled by the supply company, which then charges for gas consumption in units of energy used, typically expressed in therms (1 therm ≡ 100,000 Btu = 105,506,000 J ≈ 29.3072 kWh).

The average heating value for a particular gas depends on its composition. Natural gas is primarily composed of methane but also contains varying proportions of other hydrocarbons that are in the gas phase at standard temperature and pressure (STP). Natural gas with different proportions of hydrocarbon components thus has varying heating values. Most fields of the world produce raw gas with calorific values ranging up to 1,800 Btu/ft^3, which processing reduces to 960 Btu/ft^3 to 1,050 Btu/ft^3 for sale to market. A factor of 1,000 Btu/ft^3 is commonly used, i.e. 1 ft^3 ≅ 1,000 Btu and 1 MMcf = 1,000 MMBtu [1]. Monthly gas consumption volumes are expressed in millions of cubic feet (MMcf). Daily gas consumption rates are expressed in thousands of cubic feet per day (Mcfd/Mcfd). At the appliance level, gas consumption rates are expressed in ft^3/hr (CFH) or in liters per minute (L/min). Gas volume measurement made independent of its temperature and pressure is expressed as actual cubic foot (acf) and gas measurement that has been corrected for STP conditions is denoted as standard cubic foot (scf). In this work, all gas volumes are expressed in actual ft^3 or L and gas consumptions flow rates are expressed in actual ft^3/hr. or L/min. The conversion factors (1 ft^3 ≅ 28.32 L) and (1 ft^3/hr. ≅ 2.118 L/min) are used.

Because of the growing share of natural gas usage in the residential sector, it is important to properly manage this resource and reduce utility usage in homes. One way of doing this is to develop better ways of monitoring the amount of gas that is consumed in households. Residential natural gas usage monitoring can be classified into two major categories. First, the total amount of gas consumed needs to be recorded for billing purposes. This is achieved by using meters connected at the end point of the utilities which are read 1-3 months. Most gas meters are read manually but some are provided with automated meter reading (AMR) capabilities. As the name implies, AMR is the ability to read meter consumption data without the need for field personnel to go out and read the meter. It is presently being adopted by some major utilities and the current state of the art includes wireless capabilities, integration to the smart grid and other features.

The second category involves monitoring consumption by individual appliances to know in what proportion the major end-uses and gas requirements are divided. This can be done in one of two ways. First, each appliance inside a home can be fitted with dedicated meters that monitor its activity and relay the information to the utility directly. This is known as intrusive monitoring because it requires access to all the appliances inside a home. It is generally costly and inconvenient to the consumer and the costs increase swiftly as data requirements become complex.

A second and more preferable approach is to determine the operating history of appliance loads from analysis of measurements made solely at the entry point of a home. These methods are collectively known as nonintrusive appliance load monitoring (NIALM) techniques. Implementing NIALM technique requires the use of high-resolution meters with sophisticated data acquisition and processing capabilities, which positive displacement type mechanical meters do not have. One option is to replace them with more modern meters, like ultrasonic meters,

equipped with these advanced capabilities. However, given the vast majority of gas meters currently deployed are mechanical in nature, outright replacement of all of them is labor intensive and very costly. Therefore, a retrofit option that can increase the resolution of these meters and provide sub-metering capabilities is most practical solution.

The residential gas supply and monitoring system that is currently in place cannot meet this demand. Since the metering infrastructure is already in place, the possibility of obtaining high-resolution usage data from existing meters economically and reliably should be explored. Any method that can achieve this goal at low cost and minimum physical installation can have tremendous impact.

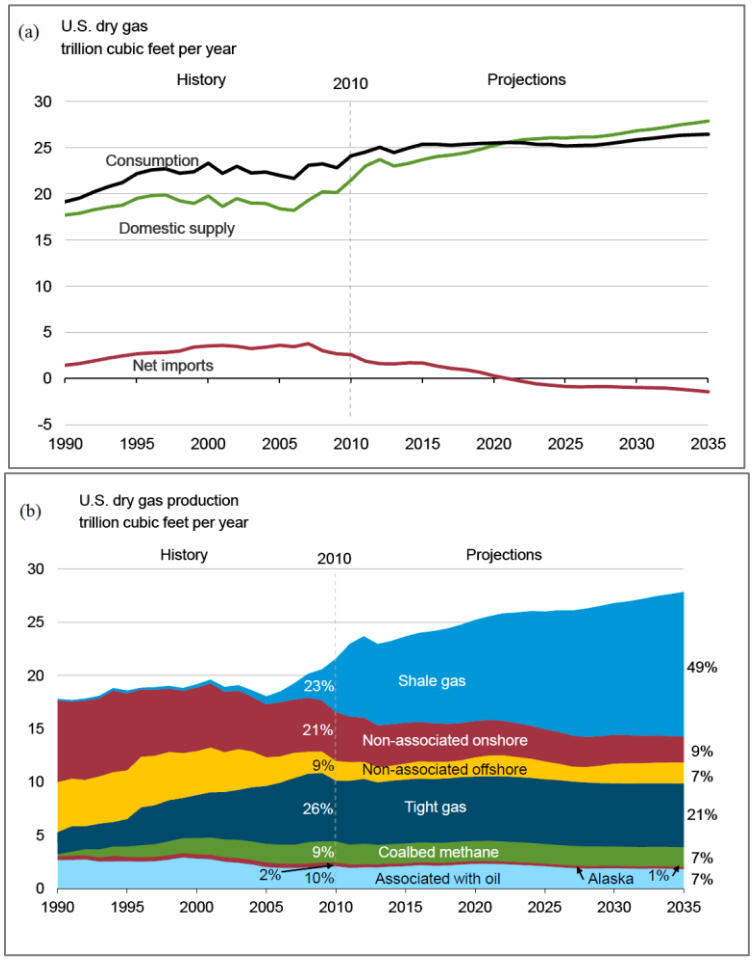


Fig. 1-1 – (a) Annual natural gas consumption and domestic supply and (b) production in the U.S

1.2 Brief History of AMR and NIALM and Literature Review

The idea of automated meter reading (AMR) was first developed in 1962 at AT&T in cooperation with a group of utilities and Westinghouse. The modern era of AMR took off in 1985, when several major full-scale projects were implemented by utility companies [4]. With the rapid growth of AMR technology also developed interest in developing an automatic metering infrastructure (AMI), which refers to the integration of AMR capable meters to provide an overview of the entire gas distribution, and non-intrusive appliance load monitoring. Presently, gas utilities are adopting AMR solutions to decrease costs, eliminates missed meter reads because of bad weather or difficulty accessing meters, reduce estimated bills and provide better service to their customers. These AMR solutions have been implemented as retrofit solutions, and several designs for a retrofitted meter modules have been proposed in the patent literature [5-7] . These modules can record daily or even hourly aggregate usage data, however, they do not have provisions for NIALM, nor do they provide a high-resolution metering capability.

Sub-metering of individual appliances for different utilities have been reported in several papers [8-10]. Sub-metering of appliances even at relatively small, statistically representative samples of houses is expensive and problematic [11]. It requires sensors that are flexible and robust enough to fit a variety of pre-existing gas appliance models. It also inherently involves multiple sensors, which increases both the technical complexity (e.g., network communications) and the complexity of deployment. For natural gas, this method poses the risk of fire or explosion.

The first practical application of NIALM for electric utilities was introduced by Hart in 1985 to monitor the power consumption of individual electrical appliances in residential homes [12-

14]. This was achieved by sophisticated analysis of current and voltage waveforms of the total load to infer the energy consumption of appliances turning on and off. Harts' algorithms were only viable for large electrical appliances with steady power consumption but were later extended to include a higher number of electrical appliances and to resolve specific loads for electrical systems [15-17]. Since then, significant advances have been made in monitoring electric load non-intrusively. Although some problems such as event overlap present challenges, NIALM for electricity is a mature field at present. A review of current advances in NIALM and power disaggregation in the following references[18, 19].

The mechanical nature of water and gas metering equipment has made it difficult for nonintrusive monitoring techniques to be implemented. As a result, there has been much less activity in this area. Water metering using different approaches to disaggregate total usage have been presented by Fogarty [20], Kim [21] and Larson [22]. The first application of NIALM for gas meters was presented by Yamagami and Nakamura in 1996 using Hart's steady-state method for Japanese households [23]. They utilized meters installed with data loggers and developed a method which they applied to homes so that they could develop demand models for the various gas appliances as a function of household configuration and occupant. Conventional sub-metering was used to validate the models. The algorithms could not consistently identify variable rate gas appliances and reported an overall accuracy rate of 95%. They suggested that further development would be required to adapt the procedure for the American market. More recently, Chon *et al.* have also presented different approach utilizing the acoustic signature of gas meter pressure regulators to perform NIALM [24].

1.3 Overall Description of Work

Natural gas meters installed in most residential homes are similar. They record the total volume of gas flowing through the meter and keep track of total usage. This work presents an approach that leverages an understanding the mechanical behavior of an industry-standard meter to gather very high-resolution usage data. It is implemented in the form of a retroactively installed module that can be attached to existing gas meters. The module is equipped with an encoder that attaches to the mechanical index output of the meter. The digital output from the encoder is processed to obtain instantaneous gas flow rates and determine the total consumption and individual appliance consumption rates in real time. Meter reading algorithms are developed using the information collected from the flow rates of various appliances to identify the appliance characteristics. This solution and can also be easily integrated with current wireless AMR technologies.

Work in the field of NIALM for electricity has progressed because power readings from electrical meters provide independent voltage and current waveforms that can be used as unique appliance signatures. In contrast, mechanical meters resolve gas loads in such large quantities that there are no signatures available for load decomposition. An appliance signature can be defined as a measurable parameter of the total load that gives information about the nature or operating state of an individual appliance in the load [12]. Since natural gas usage is settled on a monthly basis, it is impossible to decompose appliance loads from monthly bills.

Gas utilities have tried to rely on conventional means of directly inferring end-use consumption based on monthly bills, the end-uses present at a particular site, and occupant profiles. However, these methods are not accurate enough to allow the utility to reliably forecast

changes in demand due to changes in end-use or customer profiles [11]. The successful application of load decomposition techniques requires high-resolution usage data.

If high-resolution usage data is available, gas utility usage becomes advantageously distinct from electricity in two ways. First, there are typically only three to four gas consuming appliances in a household, e.g., heat, hot water, dryer, and stove/oven, in contrast to 30–60 electrical appliances and devices (electronics, small appliances, lighting, A/C, refrigeration, chargers, computing equipment, audio, etc.). In general, given n appliances, there are 2^n–1 unique ways in which they can be arranged on and off. For $n=4$ gas appliances, for example, there are 15 unique combinations of on-off states, whereas for $n=30$ electric devices there are over a billion different on-off combinations. The small number of gas consuming appliances results in a manageable number of unique on-off combinations. If the individual gas consumption rates of each appliance are known, an aggregate gas consumption rate can then be decomposed to determine which devices are currently consuming gas. As the number of on-off combinations increases, however, the ability to decompose an aggregate usage profile becomes more difficult.

Second, with the notable exception of a gas-fueled cooktop, most gas-consuming appliances do not provide a mechanism to control the amount of gas flow and consume gas in a binary fashion. The device is either off completely or consumes gas at a fixed rate. Such two-state loads are significantly easier to detect than devices in which the load varies continuously. This is demonstrated in Fig. 1-2 with two constant load appliances. In practice, when two or more gas appliances are on at the same time, their combined load may be slightly less than the sum of the individual loads due to pressure losses in the gas piping network, but these variations can be easily accounted for.

Finally, individual gas appliances have unique attributes in their usage signatures. For example, furnace loads tend to be infrequent with long durations, while dryer loads have shorter durations with more frequent on-off cycles. Cooktop/oven and hot water heater usage tend to occur at specific times of the day. Variable-flow gas-consuming devices, such as a gas cooktop or gas grill, present a challenge, because their usage signatures do not follow a simple binary pattern. Rather, as the gas flow is adjusted, the measured flow rate will vary accordingly. Several approaches can be taken. First, despite having a continuously variable flow rate, cooktops and grills are often operated in only two or three common states, e.g., high/low or high/medium/low, which, while not binary in nature, are significantly more tractable than a continuous variation. Second, behavior-based decisions can be made, e.g., for cooktops and grills, the hours of operation are limited to the typical waking hours. The time of day can also be used as a possible signature. A summary of possible signature that can be used to decompose total load gas load into constituent appliance loads is shown in Fig.1-3.

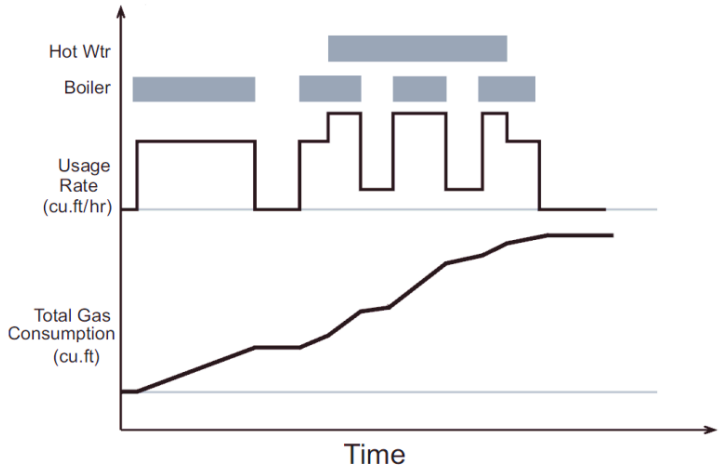


Fig. 1-2 – Load decomposition of residential appliances from total load

This work presents a system to monitor small (<0.3L (0.01ft^3)) volumes of natural gas from an existing installed gas meter. The high-resolution usage data can then be used to identify which appliances in the home are consuming gas, from which an aggregate summary of total gas

usage and usage times can be complied. The concept can be implemented retroactively through a module that can be attached to existing gas meters in residential homes, and can also be integrated with current wireless AMR technologies. The methods are also applicable for water metering as well.

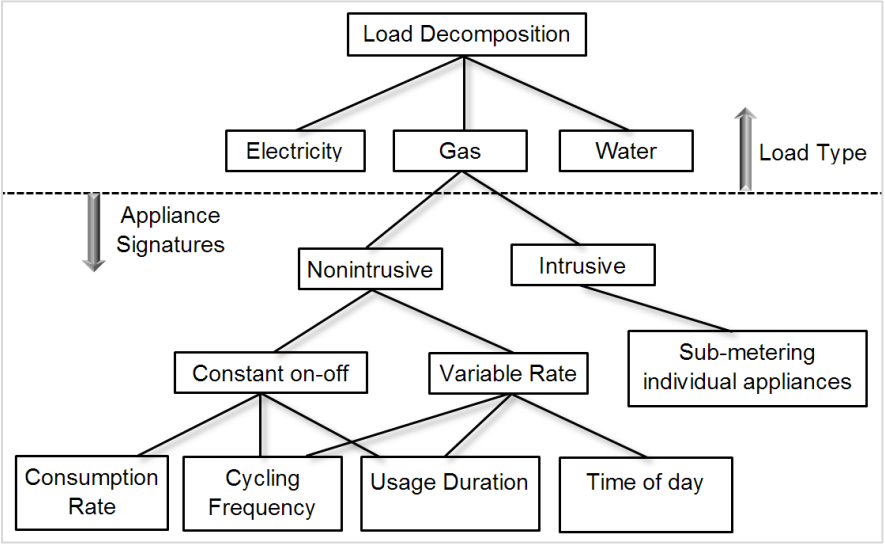


Fig. 1-3 – Gas load decomposition techniques and possible appliance signatures, adapted from work by Hart [12]

1.4 Benefits of Work

The benefits of AMR/AMI technology have been well documented [25, 26]. The difficulties involved in sub-metering gas appliances has resulted in a lack appliance specific data comparable to that available for electric appliances [23]. Because such data have resulted in large scale improvements in the distribution and use of electrical power [27], gas utilities can benefit as well determining factors affecting energy consumption and demand at the appliance level.

Unlike electricity and water usage, which are often the result of direct human actions such as watching TV, doing laundry, or taking a shower, gas usage of most appliances is automated. Therefore, there is a large gap between activity and consumption which leads to a lack of consumer understanding about how gas is used in the home and, in particular, which appliances

are most responsible for this usage. Knowledge of exactly how much an appliance consumes can also benefit the customer in many ways. Appliance-based usage data can be collected on current appliances by the utilities or third-party data processing companies. The consumer can be provided with this detailed information either in report form, on the web, or via an in-house display. This has been shown to encourage customers alter their daily energy consumption habits to save money [28-30]. The data can also be used to compare the customer's furnace or hot water efficiencies to those of modern, energy-efficient appliances, and provide estimates on the payback time of purchasing a new higher-efficiency appliance and other energy efficient measures can be implemented based on a more accurate cost-benefit calculation. In this way, consumers will be more likely to upgrade to energy-efficient appliances, while also reducing energy consumption. In addition, calorific value conversions as well as adjustments for thermodynamic property variations of the gas can be performed with increased accuracy.

Energy usage histories can also be compiled, and anomalous trends can be detected, e.g., if the furnace starts to consume more gas than usual, then a cleaning/inspection can be recommended. This will keep appliances in top working condition, thus again saving energy. If a customer calls the utility for an explanation on why their bill is high, it will be possible to determine exactly which appliance was the cause for the increased load. With current capabilities, the utility can only provide information on the times where usage spiked.

With the ability to detect small flows, it is possible to detect defective devices, leaks, theft, and other inconsistencies. With the ability to monitor historical usage, utilities can use the aggregate data collected to make more accurate estimates of energy consumption by major residential gas appliances and better evaluate the success of their conservation programs.

Equipped with high-resolution usage data, utilities can make analysis of usage data based on geographic location or demographic data. The can perform Load research, monitor demand side management programs and address high bill complaints. In the gas industry, meter errors, involve substantial amount of money. This method provides a way of checking usage profiles against historical records to ensure that the meter is working properly. Another benefit of this system is that it can be used detect malfunctioning appliances e.g. appliance rates change drastically, and the user can be notified

Finally, there is a safety benefit. This system can much more easily detect small leaks in the gas system that would be difficult with current resolutions of meters. Large efficiency reductions can indicate problems in gas-burning equipment, which can also result in carbon monoxide production. The system can detect the flows in real-time and provide immediate warnings. Maintaining appliances in top working condition thus has safety benefits as well.

1.5 Objective and Outline of Thesis

The main objective of this thesis is to develop a method of extracting high-resolution usage data from a typical residential gas meter and decompose the total load into individual appliance usage. This thesis is intended to provide a firm experimental basis for developing a retrofit solution that can be used for monitoring gas loads features such as consumption rates, appliance event identification and load decomposition. An assessment of the immediate application of the work using a low cost mechanical encoder module is also presented.

The thesis outline can be broadly classified into two categories. First, a detailed study of positive displacement type mechanical meter is conducted to understand its principle of operation and measurement performance. This includes analysis of the meter, experimental measurements and analysis of the results to determine key meter parameters that can be used to

develop algorithms increase the resolution of the meter and decompose the total load. Then using this knowledge, a system is developed to apply these methods which can be implemented in a retrofit design. The second category is looking at the practical application of this system to monitor residential gas usage of a typical home. This involves a load study of individual appliances and their combinations inside a home to be used for the development of inexpensive encoder based measurements that can be used to implement the method. This includes modeling the usage profile of gas in homes, developing a usage model to analyze the patterns and come up with the optimum parameters to be used in developing a package that allows for the realization of this work. Such a package was developed by another student working on this project.

Chapter 2 discusses the materials and methods used in this work. This will include a description of the meter, its internal workings and principle of operation. The meter mechanism motion is analyzed and its expected behavior is studied. The repeatability and linearity of the meter are established along with possible reasons for deviations from its expected behavior. Based on this study a detailed account of the experimental set up and procedures used to measure the meter parameters is also provided.

Chapter 3 presents the data that is collected from the experimental measurements performed. Analysis that was performed is also presented and also sets the precedent for possible methods and/or directions that could be taken in order to develop a high resolution metering capabilities of constant loads. These will include analysis of the meter timing data, repeatability, error analysis and meter linearity.

Chapter 4 presents the algorithm developed in this work to handle simple cases when load events are isolated and techniques to improve the meter reading accuracy. The development of the algorithm is presented with features such teaching mode and event detection.

Chapter 5 presents the results from tests performed on the high resolution meter reading algorithm. The tests include constant load, variable load and combination of different appliance loads. It also presents results from evaluation of the teaching mode features in terms of appliance detection. The discussion looks at the application of this method for real residential applications with a low cost encoder module by looking at the characteristics of usage profile of a home. Results from simulations carried to determine the optimum design of a low cost mechanical encoder module that is cost effective. The overall accuracy of the system is also discussed. Chapter 6 gives a summary of the present work and discusses some future work that could be done based on this effort.

CHAPTER II

MATERIALS AND METHODS

In this chapter, materials and methods used in this work are described in detail. This includes a description of a residential natural gas meter and its principle of operation An explanation of the different terms used in assessing meter measurement performance is also provided. A detailed description of the experimental setup and data acquisition devices is also provided.

2.1 Residential Natural Gas Meters

Several different types of metering technologies are available to measure gas volume and flow rates. These include positive displacement meters, rotary meters, turbine meters, orfice meters, coriolis and ultrasonic meters [31]. For domestic natural gas metering applications, the first practical meter developed was a positive displacement type meter with wet drums. Its high maintenance cost, large size and issues with freezing led to the invention of the first “dry” positive displacement meter in 1844 [32]. It utilized movable diaphragms to measure volumetric consumption flow rates and came be known as a diaphragm type meter. Its operating principle has remained the same since its invention, even though major changes in materials, construction and assembly have improved its life expectancy and durability [33].

In this work, a new diaphragm type, positive displacement gas meter (AC-250) manufactured by the American Meter Company is used for all experiments and is shown in Fig. 2-1. It is representative of the basic design for the majority of residential gas meters installed in the U.S over the past 50 years. The American National Standards Institute (ANSI) issues standards for different classes of meters which all meter manufacturers follow. Therefore, meters of the same class are uniform in design, construction and performance across all manufacturers. The design, construction and installation of diaphragm type gas meters having a flow rating of less than 500

ft^3/hr are governed by (ANSI B109.1-'92). For natural gas (0.6 specific gravity), the AC-250 meter has a nominal capacity of 250 standard ft^3/hr at maximum pressure rating of 0.5 psig and a maximum of 0.5 in.H_20 (1.25 mbar) differential pressure loss. This means that when operating at maximum capacity, there cannot be more than 0.5 in.H_20 pressure loss across the meter.

Fig. 2-1 – Residential natural gas meter (AC-250) used in this work

The significant features of a diaphragm type gas meter are measurement accuracy, reliability, robustness, low cost and an extremely long service life. It can last in operation for 25+ years with very little maintenance. Furthermore, its construction and design is such that it lends itself to outdoor installations, making it easier for meter readers. For these reasons, it is by far the most common type of meter installed in residential homes for natural gas metering applications. A newly emerging competition is an ultrasonic gas meter, which measures gas flow rates by measuring the speed of sound in the medium within the pipe. Although more accurate than a diaphragm type meter, it is much more expensive and requires regular maintenance.

Diaphragm type gas meters require a constant line pressure for proper operation. All residential gas meters come equipped with a regulator that controls the amount of gas flowing into the meter and provide safe pressure levels to the meter as shown in Fig. 2-2. A basic standard for a safe and reliable operation and construction of self-operated diaphragm type

natural gas pressure regulator is included in (ANSI B109.4-1998). It consists of a diaphragm with a spring-loaded case that adjusts the gas flow rate to maintain a constant pressure supply. The regulator is equipped with relief valves to vent gas harmlessly if a line becomes over-pressurized or the regulator malfunctions. This relief valve is connected to the diaphragm chamber and expels the gas through an external steel tube.

Fig. 2-2 – Natural gas pressure regulator

2.1.1 Operating Principle

The meter operates using mechanical divisions to displace discrete volumes of gas successively. Inside the meter are two disks on each side of a center partition that are attached to a flexible diaphragm (bellows) one side of which is shown in Fig. 2-3(a). It is also equipped with two slide valves that alternately divert gas through the two diaphragms. The slides are controlled by a mechanism that correctly positions the slide valves as shown in Fig. 2-3(b). One full rotation of the crank completely fills and empties the diaphragms. The diaphragm movement is timed to the number of crank turns in the mechanism and is transferred through a worm gear and shaft to the meter index as shown in Fig. 2-3(c). The worm gear contains 18 teeth; a full rotation of the index dial corresponds to 18 revolution of the crank. Fig. 2-2(d) shows the index of the meter. It rotates in the counter clockwise direction to measure the total volume of gas consumed in ft^3. Each rotation of the crank represents of 56.63 L (2 ft^3) of gas consumed.

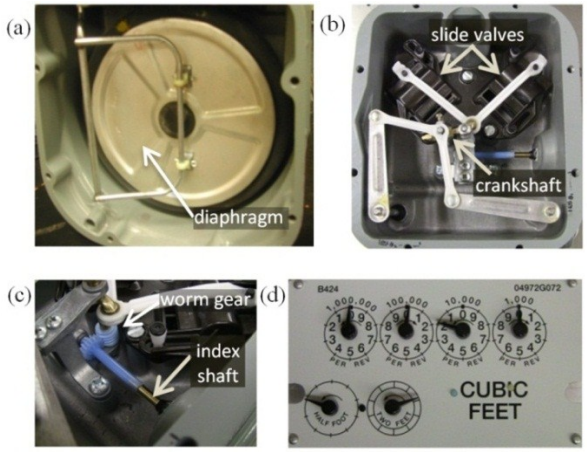


Fig. 2-3 – (a) Meter bellows (b) Internal meter mechanism (c) worm gear shaft assembly (d) meter index (dials)

The force needed to move the whole mechanical assembly comes from the differential gas pressure. The amount of pressure differential limits the operating capacity of the meter[33].

2.1.2 Meter Mechanism

The meter mechanism is composed of two four-bar linkages sharing a common crank, whose rotation controls the timing of the slide valves. These are shown in the Fig. 2-4. The input cranks to the (2) four bar mechanism are r_2 and r_6. The frame of the mechanisms are shown by $\overline{O1O3}$ and $\overline{O1O3}$. Both the input cranks have a range of 20° and are approximately 90° out of phase. Two couplers, denoted by r_3 and r_5, connect the input cranks to the output crank r_4. It can also be observed that the output is driven by the alternating action of each crank.

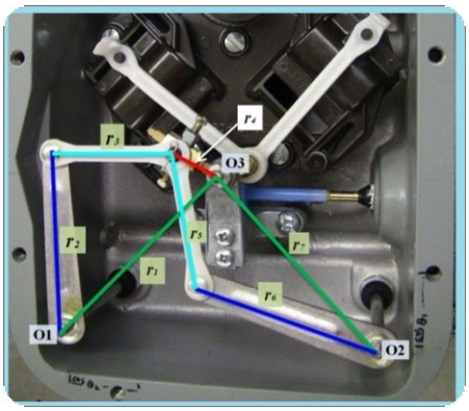


Fig. 2-4 – Meter mechanism

The mechanism inside the meter diverts gas to the two diaphragms at 90º phase so that when one is completely full, the other one is empty. There are four cycles for one full rotation as shown in Fig. 2-5. The slide valves release the increments of gas trapped in each chamber to the meter outlet once it is filled. The motion of the output crank is driven by four subsequent actions of the inputs which represent the filling and emptying of the diaphragms. In addition, its motion, directly transmitted to the index through a worm gear is used record gas volumes flowing through the meter. Therefore, it is an important to understand its motion. The movement of the meter is cyclical and repetitive. At the same pressure, each revolution passes the same volume of gas and the movement should be similar. Therefore, it is essential to understand its motion characterizes in order to utilize the information it provides to for our metering.

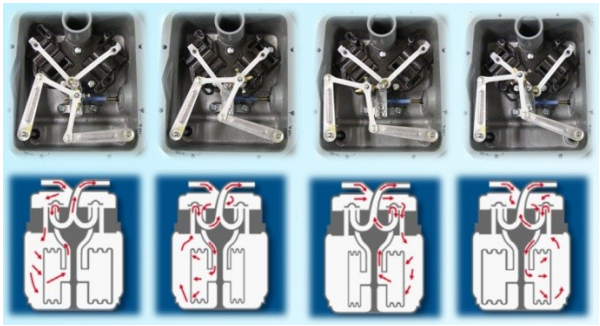


Fig. 2-5 – Four filing cycles of a diaphragm gas meter

2.2 Meter Performance Parameters

There are several terms that quantitatively express the quality of flow measurements. The standard usage of these terms often varies. In this section, an exact description of the terms that are used to express the meter measurement performance will be given.

One of the most important aspects of flow measurement is the uncertainty associated with the measurement. Uncertainty is the range of values within which the true value lies with a specified probability. Uncertainty of ±1% at 95% confidence means the instrument will give a range of

±1% for 95 readings out of 100 (e.g., ±1% of actual calibrated value). It is not acceptable to state an uncertainty without a confidence level. In gas measurement, 95% is legal minimum confidence limit. This value will also be used in this work. Because uncertainty is referred to the true value, by implication it must be obtained using a national standard or facility[34]. To calculate the measurement uncertainty associated with diaphragm type gas meter measurements, all the variables that affect the reading must be independently varied as per several standards available (e.g. AGA Report No.3 Part 1 – 1990). The variables include the atmospheric pressure, static pressure, temperature, relative density (specific gravity), gas composition, and calibration standards. A complete uncertainty analysis of this meter is beyond the scope of this work. Such analysis has been conducted by Nilsson[35] and Peksa [36]. Nilsson concluded that the major sources of error to be: 1) liquid in measuring compartment, 2) leakage in the meter and 3) increased friction in bearings.

Gas meter manufactures often provide a figure for the accuracy of their meter. This accuracy is the proximity between the measured value and the calibrated value. A meter with high accuracy more nearly gives a reading close to this value than one with low accuracy. As such, the reported accuracy may not be the same as meter uncertainty. Gas meters have to be accurate, not only to protect the customer and the utility company, but because of state laws [37]. To measure the accuracy, there must be another standard that is more accurate than the meter itself to be compared against. In a laboratory setting or manufactures test bench, another flow meter (prover) is used in series with the meter. The result from the prover is used as the actual value and the calibrated gas meter value is used as the true value. This process is known as proofing a meter. The accuracy values are given in terms percent accuracy/percent proof and percent error. There two ways a diaphragm meter can be inaccurate. The index of the meter could register more

volume than the correct amount or register less. % Accuracy measures how slowly the index measures from the correct value and % proof indicate how fast. % error is the difference between the proving scale reading and the index reading and must carry the correct algebraic sign, either + or – to designate if the meter is fast or slow [33].

When buying and selling a liquid or gas, repeatability also becomes as important as the accuracy. A precise definition of *repeatability* is the value below which the difference between any two test results, taken under constant conditions with the same observer and with a short elapsed time, are expected to lie with 95% confidence [34]. It should be noted that good repeatability does not guarantee accuracy.

Linearity may be used for instruments that give a reading approximately proportional to the true flow rate over their specified range [34]. It is usually defined by stating the maximum deviation (or nonconformity, e.g., ±1% of flow rate) within which the response lies over a stated range. Positive displacement meters are linear flow meters.

A meter also has a specified range over which values for accuracy, linearity and repeatability can be trusted. The ratio of upper range value and lower range is known as the turndown ratio. All flow meter specifications must clearly state both the accuracy and the repeatability of the meter at minimum, normal, and maximum flows. The manufacturer's uncertainty and repeatability of diaphragm type meters for uncorrected flow measurements are ±1% and ±0.2% of the full scale.

2.3 Experimental Setup

An experimental setup is used to measure key properties that of the meter that can be used to perform high-resolution meter readings. This includes characterizing the motion of the internal meter mechanism and the motion of the output shaft.

2.3.1 Gas Supply System

A typical residential gas consumption system was simulated in the laboratory for testing. A schematic of the complete setup is shown in Fig. 2-6. Air is used in place of natural gas. Because the meter records total gas volume, the actual composition of the gas does not affect the meter operation. Diaphragm meters require a clean process fluid and a constant pressure for proper operation. The compressed air is filtered through a series of air filters that remove moisture and contaminants. The air, which is at 2.3 bar (33 psig), then flows through a pressure regulator and is supplied to the meter at its operating pressure point of 27-29 mbar (11–12 in. wg).

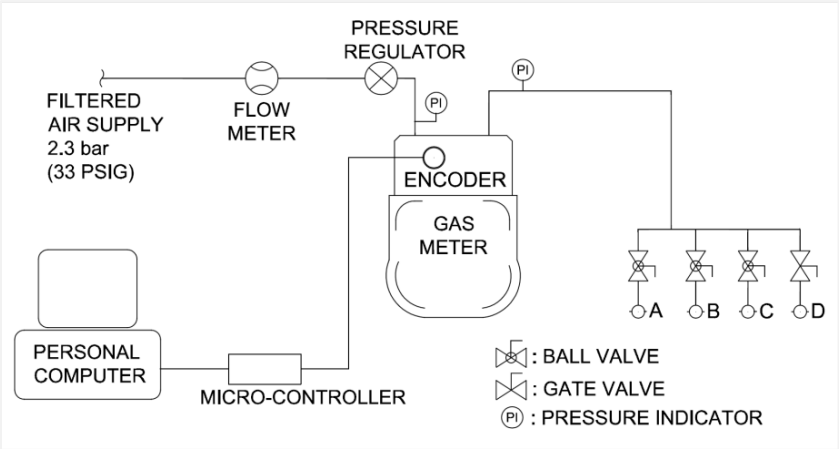


Fig. 2-6 – Experimental Setup Schematic

Before the pressure regulator, an Omega FMA 5540 high-precision mass flow controller is used to verify that the amount of flow coming into the meter is the same as the flow recorded by the meter (Fig. 2-7). The flow meter has a range of 0-200 L/min. The manufacturers' uncertainty

(accuracy) is rated at ±1.5% of full scale with a turndown ratio of 50:1. Its repeatability is rated at ±0.5% of full scale. Flow rates lower than 10 L/min cannot be accurately measured with the Omega flow meter. To verify for very low flow rates, an alternate method of is used. An empty plastic bag is filled with air for a specified period of time and the flow rate is calculated from its measured volume. The flow meter is also equipped with a totalizer that integrates the analog output flow signals from the flow meter and accumulates up to 7 digits of direct engineering units. Both the flow rate and the total volume are displayed on an LCD display.

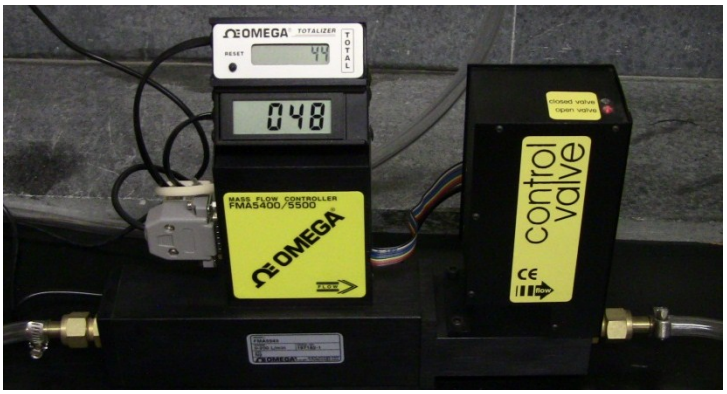


Fig. 2-7 – Omega FMA5500 mass flow controller/meter

Four appliances are simulated: a boiler/furnace, clothes dryer, and a water heater, each of which has a fixed rate of gas usage when they call for heat, and a gas-fueled stove/cooktop, which has a variable consumption rate based on the flame size of the cooktop. To simulate the appliances, ball valves fitted with brass plugs drilled with precision-diameter holes are used to simulate the fixed-consumption appliances, while a variable-flow gate valve is used to simulate the usage of a gas cooktop shown in Fig. 2-8. Standard gas ranges typically consume around 9,000 Btu/hr. per burner and around 17,000 Btu/hr. for the oven. This corresponds to approximately 4-8 L/min. The appliance heating capacity and gas consumption rates are listed on Table 3-1.

Fig. 2-8 – Valves for simulating constant load and variable load appliances

Table 2-1 – Usage rates of simulated appliances

Valve	Appliance Simulated	Consumption Rate, Btu/hr. (kW)	Flow Rate[a] L/min	Orifice Diameter, mm
A	Furnace	101,100 (30)	48	4.19
B	Water Heater	42,100 (12.3)	20	2.69
C	Dryer	23,200 (7)	11	2.06
D	Gas Stove	8,400 -17,000 (3-5)	4-8	Variable

[a] 28.5 L (1 ft^3) gas ~ 1.05 MJ (1,000 Btu)

The pressure drop across the gas meter is also monitored using pressure indicators connected to the inlet and the outlet as shown in Fig 2-9. The pressure drop across the meter can indicate if there are any leaks or other unexpected behavior.

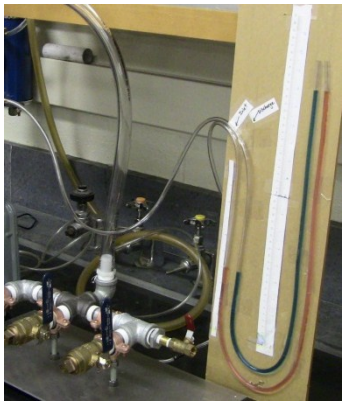


Fig. 2-9 – Pressure indicators

2.3.2 Encoder setup

A rotary optical encoder is used to capture the motion output shaft. A rotary encoder is an electro-mechanical device that converts the angular position or motion of a shaft to analog or digital code. It creates a series of pulses using a disc with alternating transparent and opaque lines placed between a light source and a photo detector. The number lines correspond to number of pulses per revolution (PPR) produced. For example, an encoder disc with a 100 lines can supply 100 PPR to measure position with a resolution of 3.6 degrees. Information on construction and materials for optical encoder can be found in many measurement handbooks [31].

A high-resolution optical encoder (TRDA-SH360) from World Encoders, Inc., is fitted to the index shaft of the meter to record the rotation of the meter index shaft as shown in Fig. 2-10. Because the gearing ratio of the internal mechanism is 18:1, an encoder PPR that is divisible by 18 is chosen to insure that one complete rotation of the internal mechanism produces an integral number of encoder pulses at the index shaft. This encoder has a base resolution of 360 pulses per revolution (PPR) and able to produce up to 1400 pulses per revolution using quadrature. Utilizing 2 channels from the encoder, 720 PPR, or 1 pulse per 0.5^{o} (0.0087 rad) of index shaft rotation, is used to capture an accurate motion profile of the index shaft. Each rotation of the internal mechanisms produces 720/18 = 40 pulses at the encoder output.

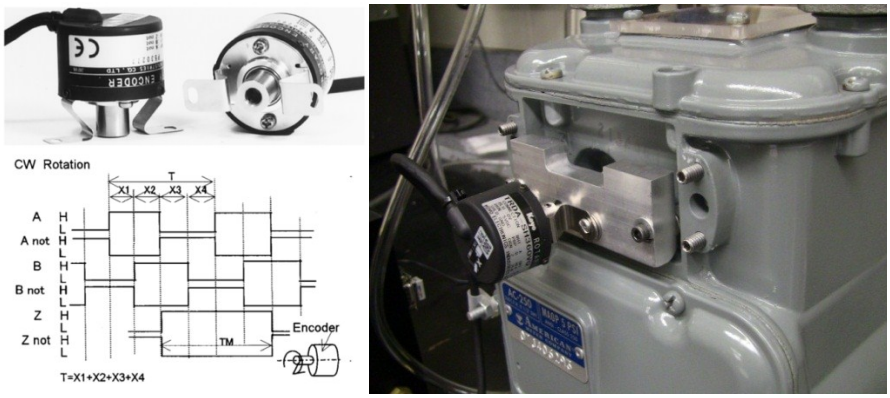


Fig. 2-10 – Optical encoder and installation to meter shaft

When attaching the encoder to the shaft, the friction should be kept at a minimum. This is done by aligning the encoder center to center with shaft. The meter has a small hole at the center out the output shaft shown in Fig. 2-11(a). By carefully center of the encoder shaft with the center of this hole, the encoder is mounted properly. A shaft bearing is used in order to keep shaft losses at a minimum as shown Fig. 2-11(b). Small misalignments in the encoder-meter shaft connection can introduce periodic angular velocity variations over the course of one revolution. Fig. 2-11(c) demonstrates this. Even though the pin on the encoder shaft (O2) will always have same radius with respect to its shaft, the contact point between the pin and the pin window side of meter shaft (O1) varies as function of the angles (φ, θ) and the eccentricity (e). The tangential velocity for the encoder-shaft would then be $v_2 = v_1 \cos(\varphi - \theta)$. Therefore, the angular velocity $\omega_2 = v_2/r_2$ will change as function of φ. One way to know if this occurs is to look at the output data from the encoder. If the motion profile increases and decreases sinusoidally over one revolution, then the shaft is not aligned properly. Misalignment can also introduce friction between the pin and the pin window on the meter shaft with can negatively influence the encoder readings. During the installation of the encoder, this problem caused considerable friction.

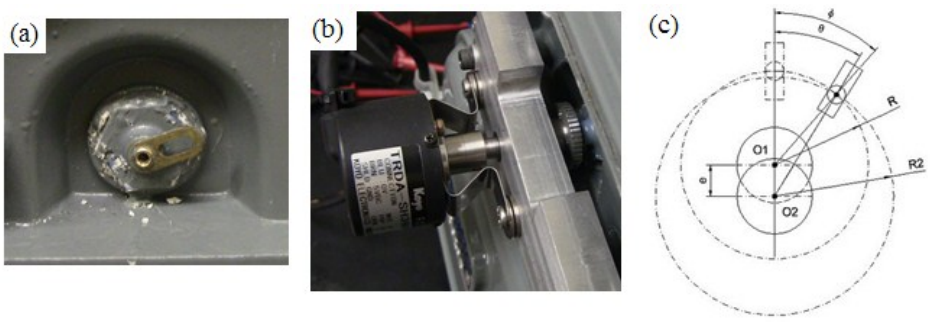


Fig. 2-11 – Encoder shaft mounting detail (a) meter output shaft (b) mounting of encoder (c) off center alignment

2.3.3 Data acquisition

The encoder output is captured by an Atmel ATmega2560 microcontroller (μC), which is used to perform data collection, timing, and data processing. The μC is connected to a personal computer using an Atmel STK600 development interface for programming, long-term data storage and post-processing and is shown in Fig. 2-12. All μC programming is done in the C language using CodeVisionAVR (HP InfoTech, Romania).

The Atmega 2560 μC has a single clock cycle execution for most commands. It contains 256 KB of flash memory, 4KB of electrically erasable programmable read-only memory (EEPROM) and 8 KB of internal static random access memory (SRAM). It also has two (4) external interrupts, (2) 8-bit Timer/Counters and (2) 16-bit Timers with separate pre-scaler. The STK600 development platform offers access to all device pins of the μC, and several useful hardware functions such as pushbuttons and LEDs. The board also has adjustable voltage supply and adjustable clock to the μC so that the clock can be easily adjusted. For debugging, it comes equipped with a JTAG interface compatible with AVR studio debugging environment to access all the register, stacks and variables at runtime.

The encoder output TTL-level pulses (0/5VDC) are connected to an external interrupt on the μC. Each pulse corresponds to roughly 56.6 L/720 = 0.0079 L or about 79 mL (2.68 fl. oz.). One

of the 16-bit high-resolution timers on the μC is used to record the time interval between successive encoder pulses. The timer was driven at the μC clock frequency of 18.46 MHz, resulting in a timing resolution of 1/18.46 MHz = 0.054 μs. If the timer overflowed, an interrupt was generated to capture the overflow event and the timer reset to zero. In this fashion, time differences of up to many seconds could be recorded with sub-microsecond resolution. Because of this high spatial and temporal resolution, round off and quantization errors commonly associated with encoder readings are minimized significantly. The encoder also provides a separate zero-reference output, which triggers when the encoder reaches the reference position. The zero pulse is used as a reference to both when the system is first powered on, and to confirm that no pulses are dropped, as each pulse from the zero reference should correspond to 720 pulses from the encoder outputs. If the accumulated pulses < 720, then some pulses have been missed, and an error is generated.

A real time clock (DS1305) by Maxim Inc. is also connected to the STK600 board. It provides a real-time clock and calendar that can be accessed by a simple serial interface. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1305 will maintain the time and date, provided the oscillator is enabled.

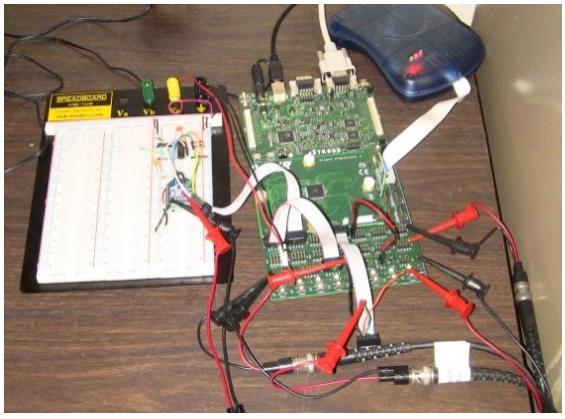


Fig. 2-12 – STK600 development board with Atmega 2560 µC and real time clock

2.3.4 Internal Meter Motion Measurements

The motion profile of the internal meter mechanism is expected to be repeatable since it measures the same amount of gas with each rotation. To verify this experimentally requires taking measurements of the meter's internal mechanism. The internal four-bar mechanism was also made observable by replacing the meter top cover with clear acrylic glass as shown in Fig. 2-13. A Canon HF200 HD digital camera is used to record the motion of the mechanism crankshaft at 29 frames per second. These images are recorded at the maximum possible resolution on the camera (1920x1080p). The recorded video is then processed with Tracker v. 4.05, a video analysis and modeling tool provided at no cost by Open Source Physics [38], in order to track the crank motion and analyze it motion.

Fig. 2-13 – Acrylic Cover to expose internal meter mechanism

CHAPTER III

EXPERIMENTAL RESULTS

The experimental setup described in Section 2.3 is used to gather all the required measurements from the meter and fully characterize its property. The properties of interest are the meter timing data and motion profile to assess the meter accuracy, repeatability and linearity and also determine if there are any deviations from the expected behavior. Measurements are collected by running the meter for several different flow rates using different valve combinations. Both measurements collected from the encoder and the video recordings are analyzed in Microsoft® Excel 2010. In this chapter, the important features of the results will be presented.

3.1 Internal Meter Motion Measurement Results

The video data collected from the internal meter mechanism is processed with Tracker v. 4.05, a video analysis and modeling tool. This tool can track motion specified points in a video. This is done by selecting two points in the first frame and setting a reference distance between them. Based on this information, the program then tracks the relative position of the two points in subsequent frames as the movie progresses. A screen shot of the program is shown in Fig. 3-1. It shows the view from the video camera oriented at the top of the meter, looking through the clear acrylic glass.

In order to track the point on the crank, a distinguishing feature must be placed on the point to be tracked. A small dot with red color is applied to a small area on the crank is used as one of the points and the center of the shaft is used as a stationary point. Seven representative flow rates spanning the entire operating range of the meter are taken, with three runs taken for each flow rate. The primary data that was gathered was the location of specified point on the output crank

as a function of time. The program calculates then calculates the velocity and acceleration of this point automatically. The results are then exported to a spreadsheet for plotting and analysis.

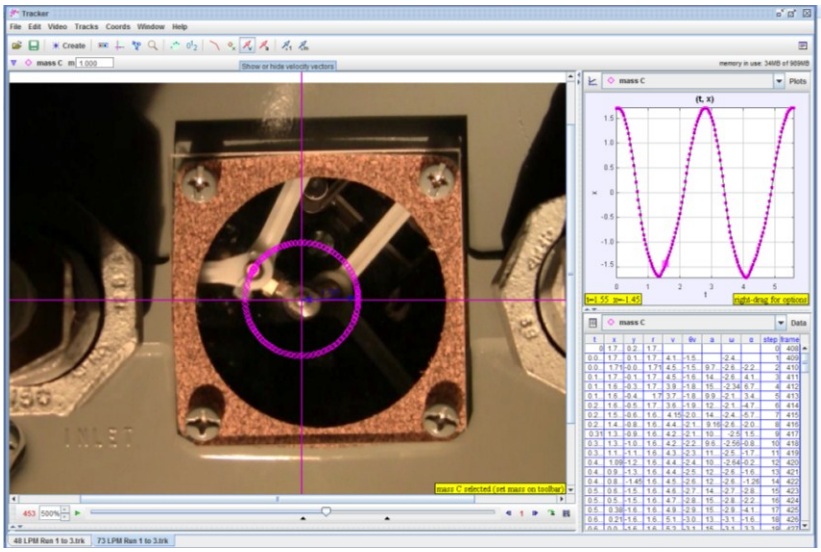


Fig. 3-1 – Video analysis software for collecting internal meter motion

These measurements also provide timing information for the motion of the internal shaft in order to see if there are any variations at constant flow rate. Table 3-1 summarizes the data taken from three complete revolutions that were taken at 8 different flow rates.

Table 3-1 – Summary of internal meter motion data from video measurements

Flow Rate (L/min)	Time for one full revolution [sec]			Average Velocity [rad/sec]				Min. Velocity [rad/sec]				Max. Velocity [rad/sec]			
	Run			Run			% Error	Run			% Error	Run			% Error
	1	2	3	1	2	3		1	2	3		1	2	3	
105	1.98	1.96	1.98	3.16	3.19	3.16	0.87	2.26	2.30	2.28	1.21	4.72	4.64	5.14	15.51
73	2.75	2.75	2.75	2.29	2.28	2.29	0.36	1.61	1.60	1.57	1.17	3.32	3.38	3.27	3.09
64	3.07	3.07	3.07	2.03	2.03	2.03	0.12	1.43	1.40	1.43	0.95	3.01	3.01	3.04	1.04
58	3.44	3.44	3.44	1.82	1.82	1.82	0.09	1.25	1.20	1.29	2.69	3.13	3.03	3.04	3.10
48	4.1	4.1	4.1	1.54	1.53	1.53	0.26	1.07	1.06	1.06	0.32	2.28	2.26	2.32	1.91
28	6.45	6.45	6.45	0.98	0.98	0.98	0.03	0.68	0.68	0.67	0.20	1.49	1.46	1.47	0.93
9	16.9	16.9	16.9	0.37	0.37	0.37	0	0.26	0.25	0.26	0.10	0.57	0.57	0.55	0.44

From Table 3-1, it can be seen that even though the velocity of the output crank is not constant, the time it takes to complete one full revolution is the same for any given constant flow

rate. This is expected as the motion of the crank directly translates the filling and emptying of the diaphragms. It was also found that the average velocity is the same for subsequent rotations. The readings are more accurate for lower flow rates with variations less than 0.5% measured. The range of variation between the maximum and minimum velocities is also determined. For example, at 58 L/min, the velocity has a range of 1.2–3.1 rad/s in one revolution. There is also a 2.69% error in the minimum velocity between the three readings. At 105 L/min, there is a 15.5% error between subsequent measurements for the maximum velocity. For high speed motions, the results are not very accurate. It should be noted that these variations increase with increasing flow rate. This difference can be attributed to the video recorder whose sampling rate is fixed, therefore, has larger error for higher speeds. In addition, the velocity and acceleration are calculated using first-order difference equations. However, it is sufficient to qualitatively observe the motion profile of the internal meter motion.

Fig. 3-2 shows a plot of the angular velocity with the angular position that was generated for one full revolution for all the different flow rates. It shows that the velocity of the mechanism goes through four cycles of speeding up and slowing down motion in one full revolution. There are four locations of high speed for each revolution, which correspond to the points where the output crank is being driven by one input crank. However, since the time for one complete revolution is constant, these speed variations do not have any effect on the average values. The motion is observed to have a similar pattern for all the flow rates.

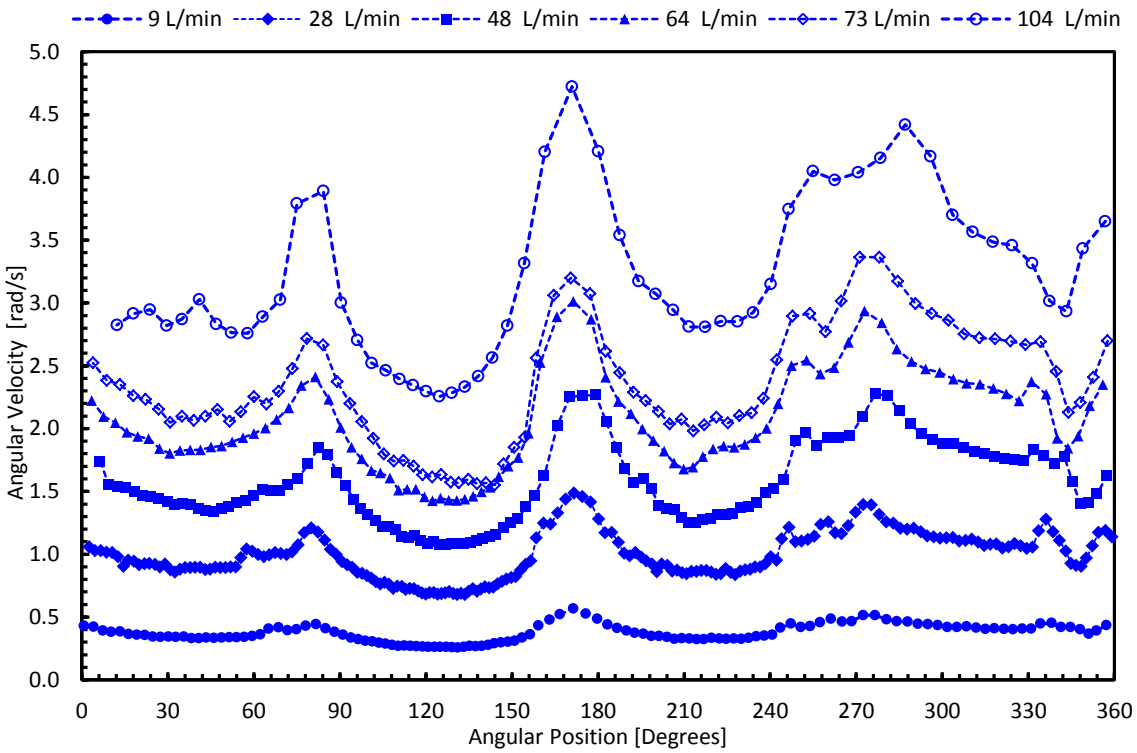


Fig. 3-2 – Angular velocity vs. position for the internal meter mechanism for representative flow rates

The calculated acceleration plots are also given in Fig. 3-3. It also shows that there are four distinct locations where the mechanism accelerates and then quickly decelerates before moving again with a constant velocity. These points of very low speed are approximately 90° apart and correspond to one distinct position on the diaphragm movements. The motion of the output shaft should be similar to that of the internal mechanism, scaled down by a factor of 18 as it is transferred through a worm gear.

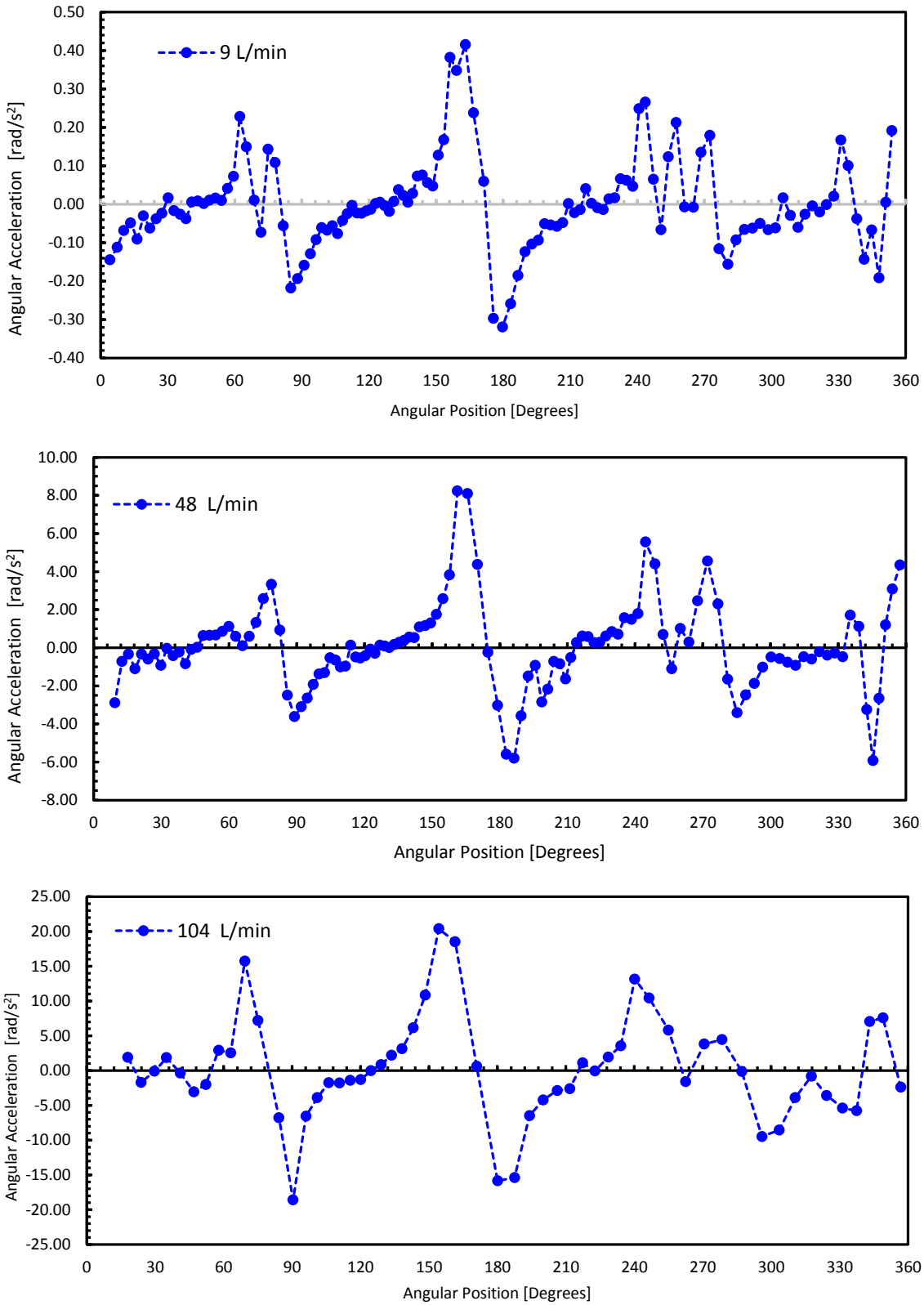


Fig. 3-3 – Angular acceleration vs. position for the internal meter mechanism for (a) 9 L\min (b) 48 L\min (c) 104 L\min

3.2 Encoder Measurements

Obtaining measurements from the encoder is straightforward. A C program is written to generate an interrupt every time a pulse is received from the encoder. The measurement starts by waiting for the encoder zero pulse, after which the elapsed time between two successive pulses is continuously recorded. The encoder zero pulse is used to check if no pulses have been dropped in one revolution of the meter. The angular distance between the two pulses is 0.5°(0.0087 rad). The instantaneous angular velocity at the i^{th} encoder position, ω_i (rad/s), of the index shaft is approximated by a first order difference equation

$$\omega_i = \frac{\Delta\theta}{t_i - t_{i-1}} = \frac{0.0087 \text{ rad}}{\Delta t} \qquad \textbf{3-1}$$

A plot of the instantaneous angular velocity as a function of time is given in Fig. 3-4. It shows the velocity profile of the index at 49 L/min and clearly shows the 18 cycles of the internal meter mechanism corresponding to the 18 turn ratios of the worm gear connected to the output shaft. The fluctuations in the times are due to the linkage motion whose four cycles are also visible.

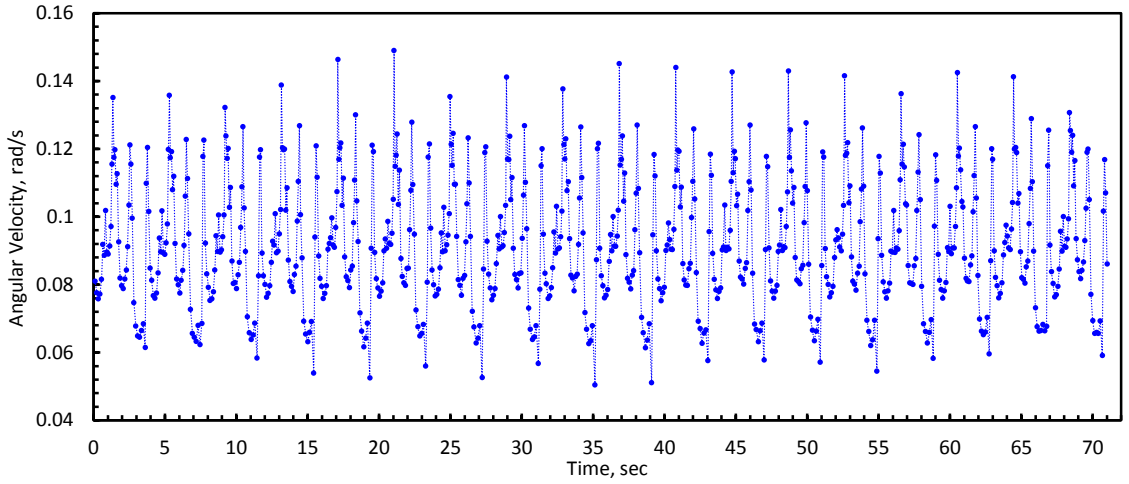


Fig. 3-4 – Angular velocity vs. time for the index shaft at 49 L/min

The results from the encoder measurements are used to characterize mechanical motion of the meter as a function of gas flow rate. Ten different representative gas flow rates ranging from 3

L/Min (7 ft^3/hr) to 99 L/Min (210 ft^3/hr) are established through the meter using the different valve combinations. The Omega flow meter is used as a reference. A summary of the average, maximum, and minimum rotation rates and time for one complete revolution of the index shaft (720 pulses) is shown in in Table 3-2. As can be seen, the maximum and minimum shaft velocity can vary by nearly a factor of three similar to the results from video measurements.

Table 3-2 – Summary of encoder measurements of meter index

Flow Rate (L/min)	Avg. index shaft velocity (rad/s)	Min. index shaft velocity (rad/s)	Max. index shaft velocity (rad/s)	Time for one index shaft rotation (s)
99	0.1803	0.1063	0.2765	36.33 ± 0.03
87	0.1630	0.0958	0.2513	40.20 ± 0.03
75	0.1392	0.0821	0.2170	47.02 ± 0.06
64	0.1217	0.0722	0.1937	53.82 ± 0.01
51	0.0960	0.0572	0.1518	68.23 ± 0.01
38	0.0769	0.0453	0.1189	85.13 ± 0.03
27	0.0594	0.0349	0.0911	110.18 ± 0.05
15	0.0382	0.0224	0.0582	171.3 ± 0.1
7	0.0144	0.0085	0.0220	454.4 ± 0.1
3	0.0045	0.0027	0.0073	1447.7 ± 0.1

Increasing the flow rate decreases the average time between pulses, because the mechanical meter mechanism moves faster to accommodate the higher volume flow rate. The instantaneous angular velocity for two complete revolutions of the internal mechanism (80 pulses total) is shown in Fig. 3-5. Each group of 40 pulses corresponds to one complete revolution of the internal meter mechanism. This data range corresponds to 80° rotation of the index shaft. As a consequence of the non-linear mechanical motion of the internal meter mechanism, the time between pulses for a single rotation of the internal meter mechanism (40 pulses) can vary significantly.

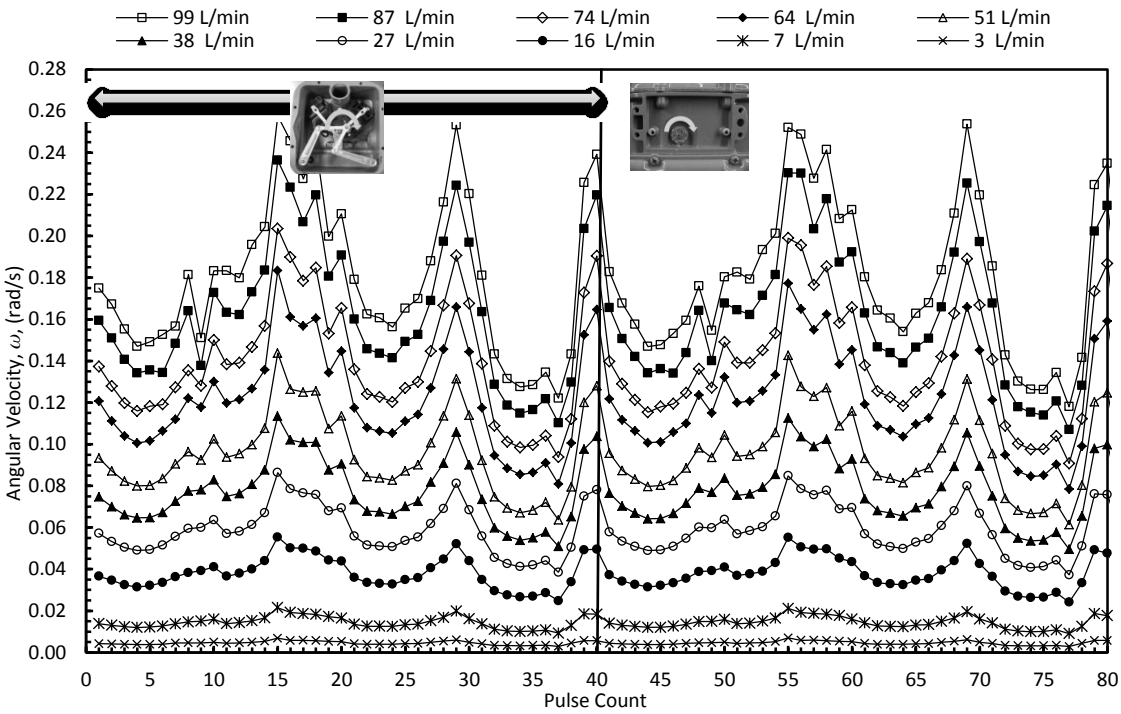


Fig. 3-5 – Instantaneous angular velocity for ten different flow rates

The setup is also used to perform accurate time measurements for the output shaft allows with sub-microsecond resolution. Table 3-3 summarizes the timing data that was collected at different flow rates over several runs. These results show an average of 36 measurements with the 95% confidence interval. The time intervals between these different measurements are accurate to within 0.04 seconds at high speeds and 0.2 seconds at lower speeds. These deviations are well within the range of meter uncertainty. The time period of internal meter mechanisms is also measured as it is the motion of the output shaft that transmits it through the worm gear. The results show that the time intervals for one complete cycle.

Table 3-3 – Results from timing measurements over several runs

Flow rate, L/min	Average time to complete one revolution, S	Time for one revolution of meter internal meter mechanism, S
9	293.4 ± 0.2	16.3 ± 0.2
16	169.74 ± 0.2	9.43 ± 0.1
26	112.32 ± 0.1	6.24 ± 0.08
49	71.1 ± 0.1	3.95 ± 0.04
58	59.76 ± 0.1	3.32 ± 0.04
64	53.64 ± 0.1	2.98 ± 0.04
73	47.52 ± 0.1	2.64 ± 0.04

3.3 Analysis of Experimental Results

The results presented thus far indicate that motion profile of the internal meter mechanism transmitted to the output shaft is similar. The ability to resolve these readings into meaningful usage rates depends on this property. Therefore, the statistics of these readings must be established with high fidelity. In this section, the analysis of the measurements for the accuracy, repeatability and linearity of the meter we will be presented.

3.3.1 Accuracy

As stated earlier, it is important to verify that meter is operating within the stated manufacturers' accuracy. The Omega mass flow meter is used a proving scale to do this by comparing it with the meter index measurement. The encoder is can be used to keep track of the number of revolutions the index performs within a given period and the total gas can be found using 56.6 L/rev (2 ft^3/rev). The Omega flow meter is not accurate for very low flow rates, less than 20% for the meter range. To measure low flow rates accurately, the gate valve is used. Careful measurements are made to reference the position of the hand wheel on the gate valve with specific flow rate values recorded by the flow meter. While performing these measurements, air flowing through the meter was also collected for a fixed duration and its

volume was recorded. This flow rate calculated from this additional measurement provided a means of verifying the readings from the Omega mass flow meter. From these measurements, it was established the gate valve can effectively be used to measure flow rates of 3 to 8 L/min in increments of ~1 L/min with each ¼ turn of the hand wheel.

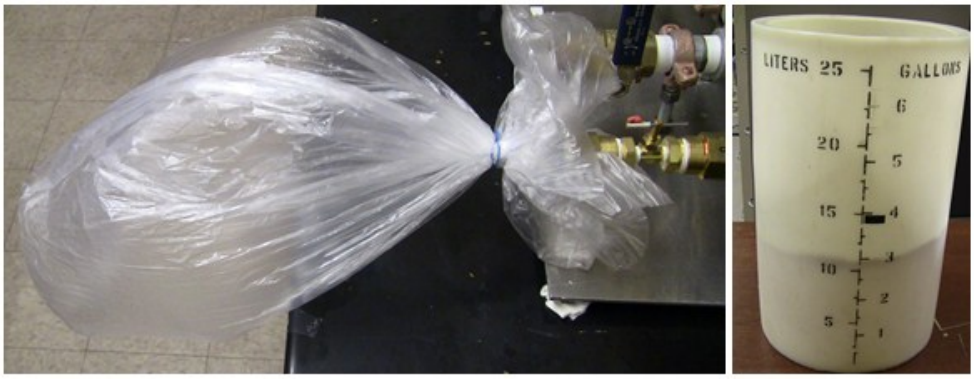


Fig. 3-6 – Method of proofing meter for low flow rates

In this way, the meter is proofed for the entire flow range. Table 3-4 shows the results of the measurements taken and %proof and % errors. The plot of the %proof vs. flow rates is shown in Fig. 3-6. The results show that the meter is operating within the manufacturers' accuracy.

Table 3-4 – Meter proof curve validation

Flow Rate (L/min)	No of revolutions	Total Volume, Meter index (56.6 L/rev)	Total Volume, L	% Proof	% Error
95	4	227	228	101%	-0.65%
74	3	170	171	101%	-0.65%
64	3	170	171	101%	-0.65%
58	4	227	228	101%	-0.65%
49	3	170	170	100%	-0.06%
26	3	170	170	100%	-0.06%
16	4	227	226	100%	0.23%
9	4	227	225	99%	0.67%
4	0.25	14.2	15	99%	1.11%

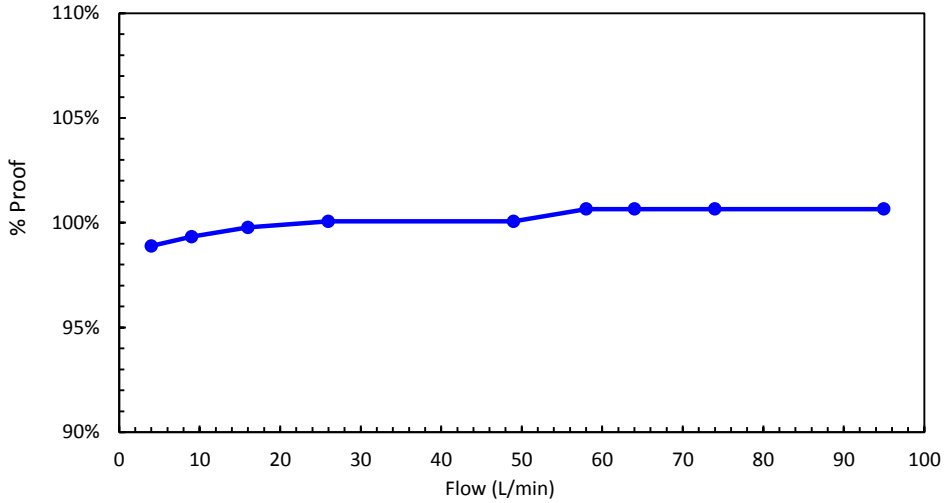


Fig. 3-7 – Meter proof curve validation

The pressure drop readings across the meter are also taken to verify if the meter is operating properly. This involves comparing the curve to the standard supplied by the manufacturers' specifications. If there is a large pressure drop, this would indicate friction in the mechanism, leakage in the meter or other abnormal operation. The results from the measurements are given in Table 3-5 and Fig. 3-6 and show that the meter is operating as expected.

Table 3-5 – Meter differential pressure measurements

Flow Rate, L/min	Inlet Pressure, in w.g.	Outlet Pressure, in w.g.	Differential Pressure, in w.g,
4	10.44	10.41	0.03
8	9.81	9.78	0.04
16	9.50	9.44	0.06
26	9.06	8.95	0.11
48	8.63	8.43	0.20
58	8.16	7.89	0.27
65	7.94	7.66	0.28
73	7.69	7.38	0.31
114	1.25	0.75	0.50

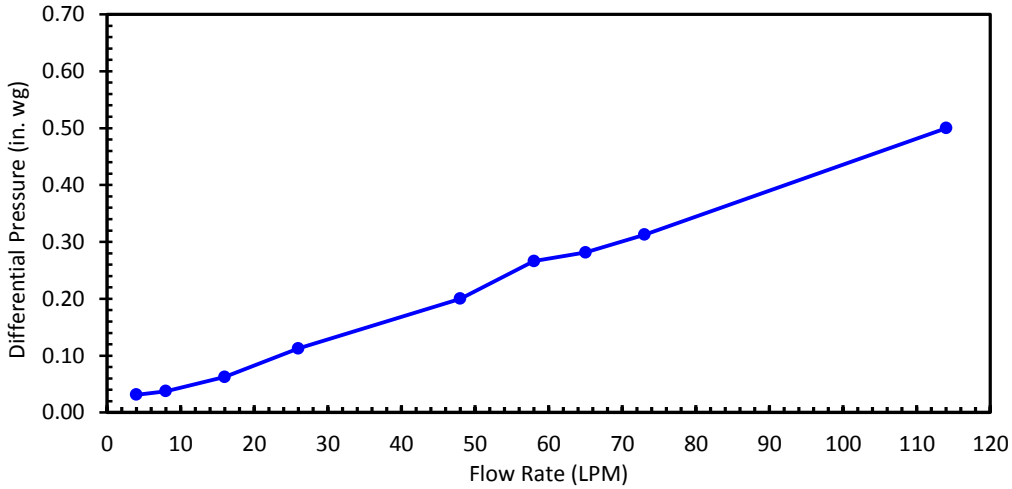


Fig. 3-8 – Measured pressure drop across meter

3.3.2 Repeatability

The next step is to calculate repeatability of the measurements. This is tested over one full rotation of the internal meter mechanism and one full rotation of the index. Repeatability is verified by statistical analysis of the measurements to see how much difference there is between two similar encoder positions readings taken under similar conditions. A student's t-test gives the 95% confidence that any subsequent measurements will fall within that interval. The test is conducted for the entire flow range of the meter. Meter repeatability can be used in order to determine the meter flow rate from a single encoder data points. Therefore, it is the main property that would allow the high-resolution meter reading concept to work.

To test the repeatability of the output shaft over one full rotation of the internal meter mechanism, the time between encoder pulses for each of the first 40 pulses (one complete internal meter mechanism revolution) are taken and the mean, maximum, minimum and a standard deviations are calculated for each pulse point. Table 3-6 summarizes the results of this calculation performed over 54 revolution of the internal meter mechanism taken at constant flow rate of 49 L/min.

Table 3-6 – Repeatability analysis of the internal meters mechanism at 49 L/Min

Pulse Point	Sample Size (*N*)	Mean (*x-bar*)	Min.	Max.	Std. dev. (σ)	Pulse Point	Sample Size (*N*)	Mean (*x-bar*)	Min.	Max.	Std. dev. (σ)
1	54	109.69	107.26	112.10	1.13	21	54	108.05	103.42	110.85	1.48
2	54	114.65	112.35	116.84	0.91	22	54	110.23	106.60	114.65	1.87
3	54	113.05	111.07	115.92	1.20	23	54	104.27	101.52	108.02	1.76
4	54	110.13	107.83	113.31	1.24	24	54	102.05	97.95	107.03	2.03
5	54	99.87	94.66	107.72	3.49	25	54	90.12	85.48	95.02	3.19
6	54	94.09	90.87	97.93	1.92	26	54	80.72	76.17	84.92	2.21
7	54	93.10	84.05	98.85	5.07	27	54	69.38	67.22	72.71	1.33
8	54	89.65	83.90	98.19	4.61	28	54	80.83	72.77	88.17	3.23
9	54	96.21	94.38	98.26	0.71	29	54	97.41	83.87	111.05	7.22
10	54	96.34	94.71	97.94	0.78	30	54	122.71	113.54	128.19	3.61
11	54	91.92	89.41	95.89	1.70	31	54	130.65	126.18	135.04	1.83
12	54	83.87	77.17	89.97	3.33	32	54	137.04	130.78	142.70	2.75
13	54	63.29	58.75	75.27	4.13	33	54	134.14	131.11	139.13	2.17
14	54	73.05	63.95	77.79	3.65	34	54	127.78	123.69	132.34	2.20
15	54	73.58	71.71	76.19	1.05	35	54	152.18	125.02	174.24	13.52
16	54	72.45	69.78	76.79	1.92	36	54	106.23	88.42	147.60	17.14
17	54	83.16	79.17	86.43	2.02	37	54	74.43	70.95	85.35	2.97
18	54	79.55	74.61	85.86	2.75	38	54	74.41	69.30	78.83	2.66
19	54	98.11	92.18	102.41	2.69	39	54	95.00	81.40	99.33	4.50
20	54	106.56	100.58	109.32	1.79	40	54	105.77	101.08	108.77	1.77

All values in this table are time differences measured in milliseconds.

From Table 3-6, we can see that meters internal meter mechanism executes a very similar motion profile for a constant flow rate with a standard deviation less than 7 ms for most of the flow range. The highest range of uncertainty is with pulses 35–37, which corresponds to an abrupt change in velocity in the meter internal mechanism, where the variation is ~3%. From Fig. 3-9, it can be see that at this point, variations between different readings can be as high as 17ms. This point was observed during the video measurements conducted in the as a point where

the crank motion undergoes a quick deceleration. This point affects the repeatability of the internal meter mechanism, however, the variations are still very small compared to the level of accuracy need for high resolution metering concept to be fully implemented.

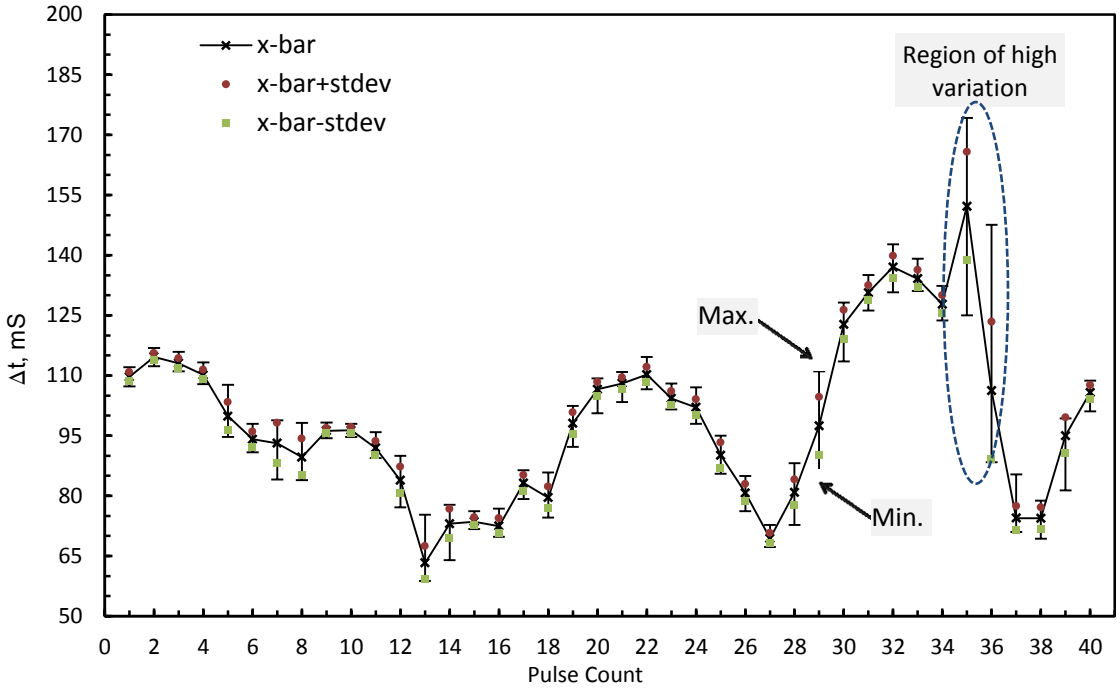
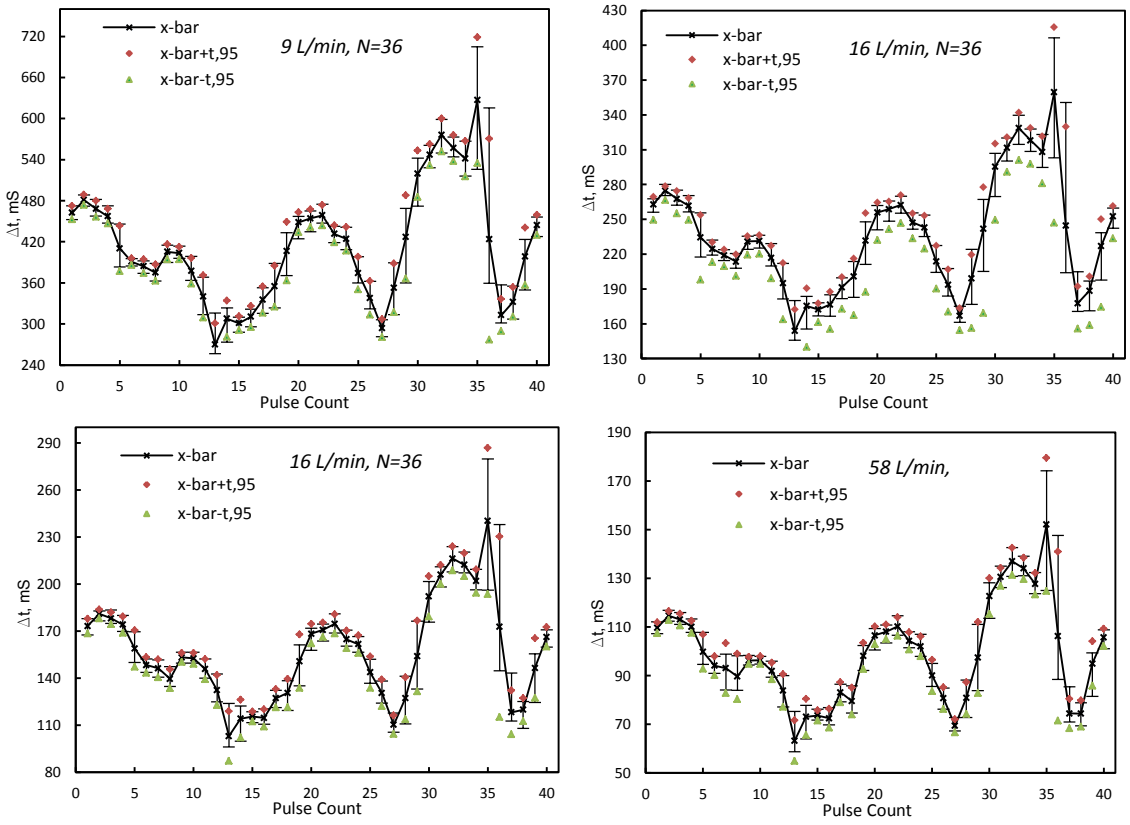


Fig. 3-9 – Repeatability analysis of the internal meter mechanism at a flow rate of 49 L/min

Next, the uncertainty of the mean of these measurements is calculated for the other flow rates spanning the operating range of the meter. At least 36 measurements were taken for the internal revolution of the meter at 40 pulses per revolution (PPR). These values were randomly selected from several trails that taken at different times. Because the sample size is small, a student-t test is used to calculate the 95% confidence intervals. The results are displayed on Fig. 3-10. This shows the plot of the average of the reading indicated in the figures as '*x-bar'*. The error bars represent the minimum and maximum times recorded in the readings and the data points are the 95% confidence interval that pulse width will fall within those intervals. These results clearly

demonstrate internal revolution of the meter is repeatable over a major portion of the motion with high accuracy i.e., the motion pattern observed are similar. It also demonstrates that this behavior is similar over the entire range of flows. This result is important and is useful in developing the meter reading algorithms discussed in the next section. This is due to the fact that the output shaft goes through the 18 teeth of the worm gear once in a full revolution. The worm gear responds differently to the quick deceleration depending on where it is. This may be due to slippage of friction. None the less, it can be used to obtain high resolution metering data from the meter.



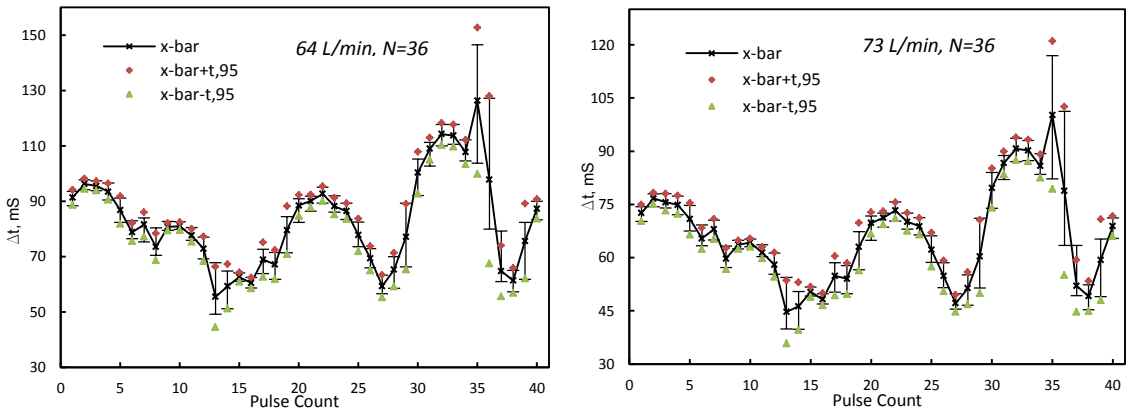


Fig. 3-10 – Student-t test results for meter repeatability over 40 pulse points.

To assess the repeatability of the meter/encoder setup, timing data were collected for 15 consecutive revolutions of the meter index at a flow rate of 49 L/min. Then, similar to the previous calculation, the mean, minimum, maximum and 95% confidence intervals are calculated. This is shown in Fig. 3-11. The meter mechanism is repeatable with variations less than 1% measured.

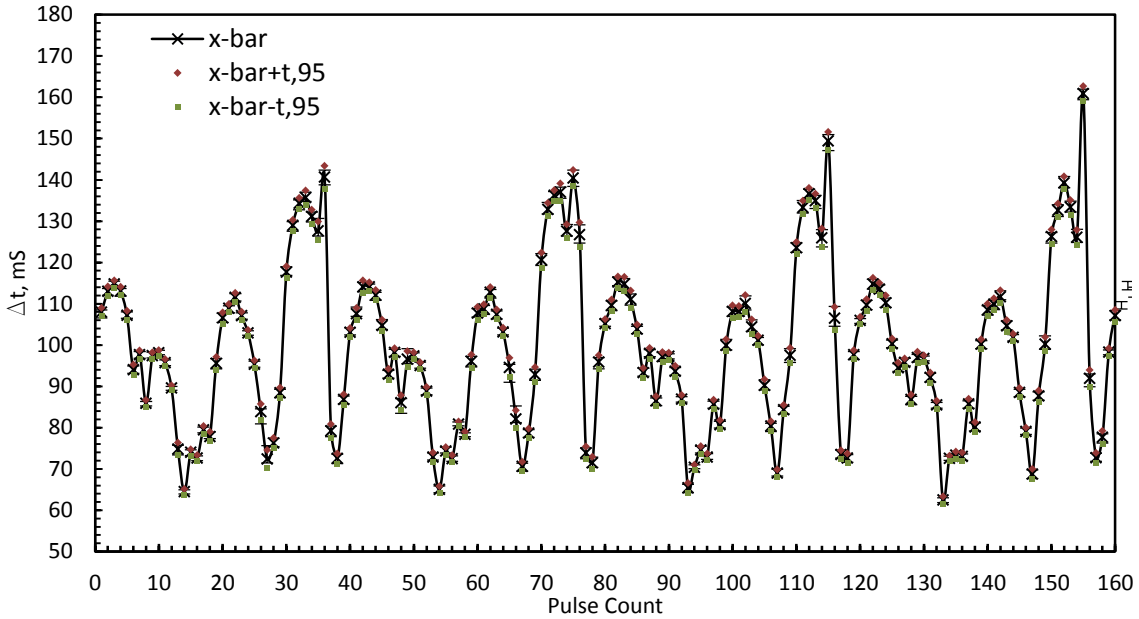


Fig. 3-11 – Repeatability analysis of output shaft at 49 L/min

Finally, a similar test is performed for the other flow rates and the results are shown in Fig. 3-12. The results show similar repeatability across the entire flow range with variations less the 1% measured.

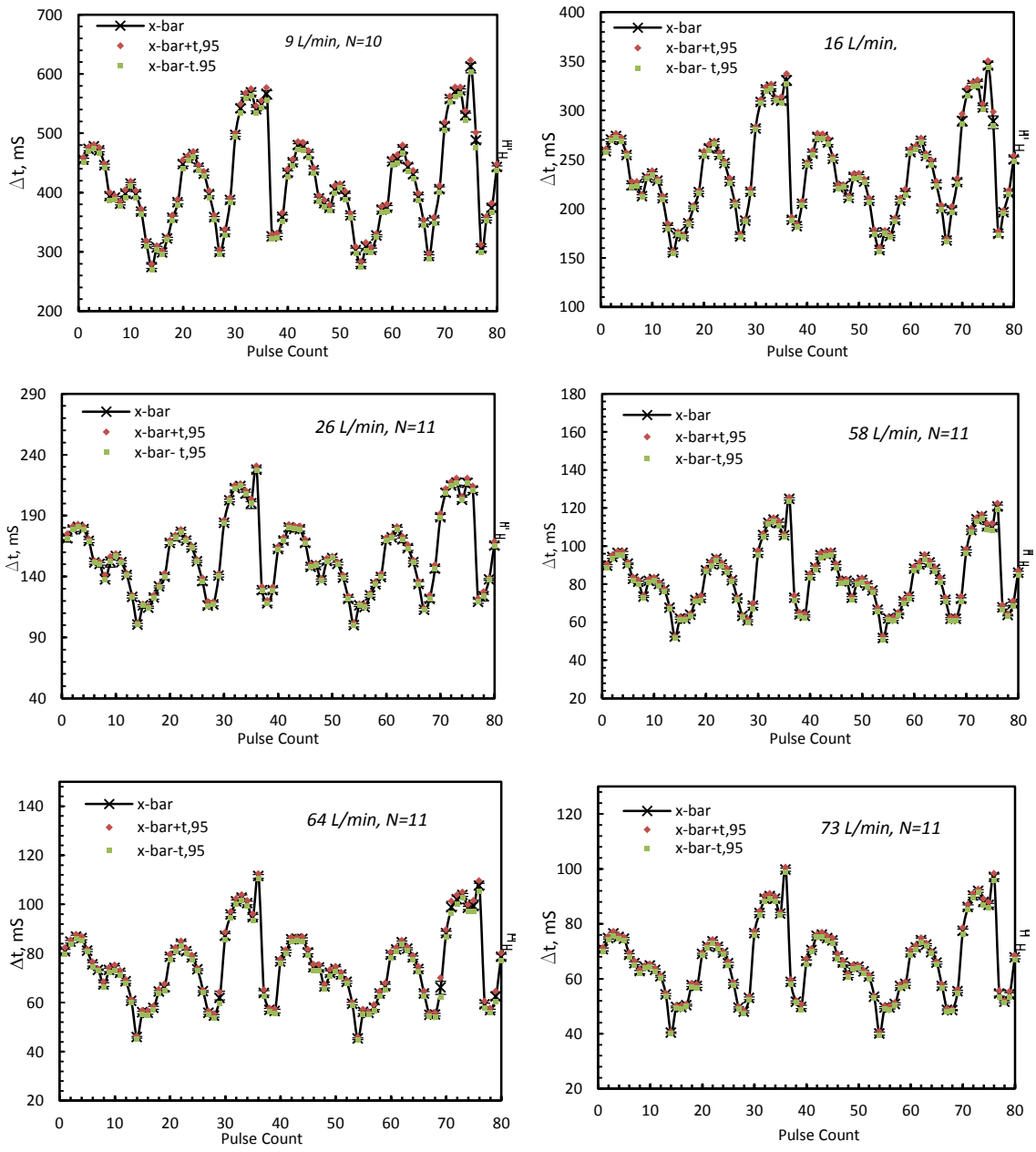


Fig. 3-12 – Student-t test results for output shaft over 80 pulse points.

3.3.3 Scaling Analysis

The fact the motion is repeatable over different flow ranges indicate that for different flow rates, the meter is executing the same motion profile but at increased or reduced speed. This indicates the meter velocity can be normalized by encoder position. In order to verify this, a scaling analysis is performed on the data collected. An empirical scaling factor is sought to see if the curves fall on top of each other. A constant flow rate of 51 L\min was used as base line. Then, all the velocity values for each flow rate are scaled by a constant factor until all the curves are coincident. The scale factors determined from this process are shown in Table 3-7 and the scaled curves are shown in Fig. 3-13. As can be seen, all flow rate data collapse reasonably well onto a single curve, confirming that the meter mechanism motion scales up or down with flow rate.

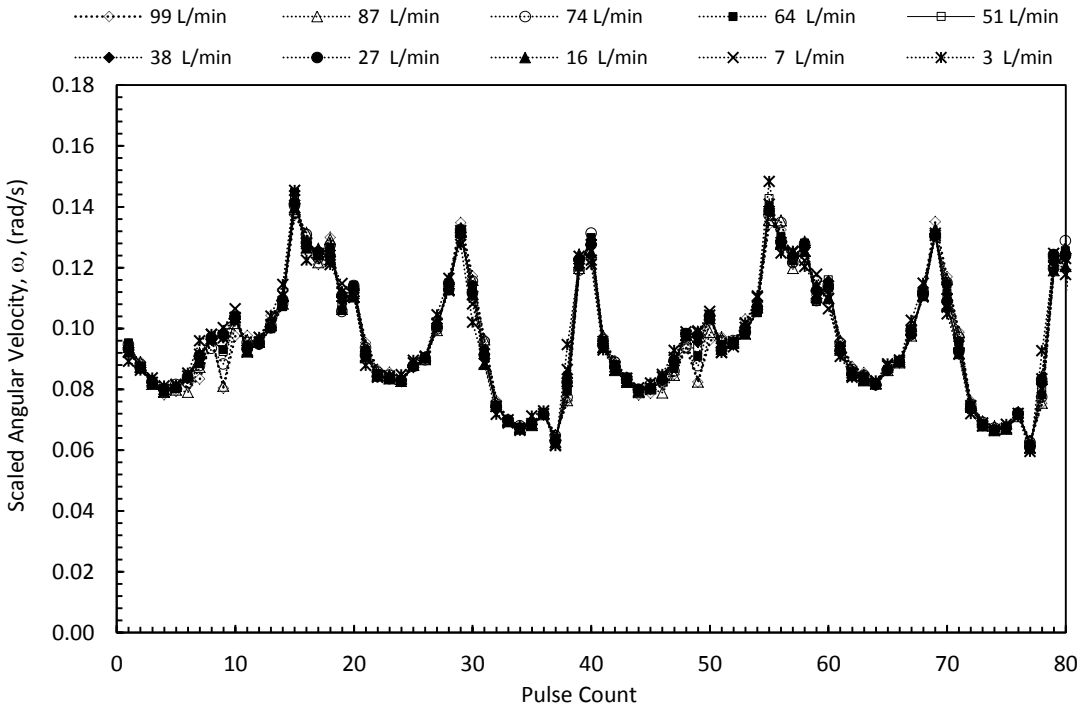


Fig. 3-13 – Scaling of all flow rates to 49 L/min

Table 3-7 – Scale factors for Fig. 3-13

Flow rate (L/min)	Scale factors
99	1.88
87	1.70
74	1.45
64	1.27
51	1.00
38	0.80
27	0.62
16	0.40
7	0.15
3	0.05

The fact that the motion is repeatable also indicates that the measurements can be separated into distinct regions by using a reference curve. This is shown in Fig. 3-14, where the measured velocity measurements recorded from the meter shown at different positions. By using the mean of several points for a specific encoder position as a reference, all other flow rates can be normalized by it, separating the curves from each other. This can be done without any loss of information because the repeatability of the meter has been confirmed. The result of this operation applied on the original data from the meter (Fig. 3-13) is shown in Fig 3-14.

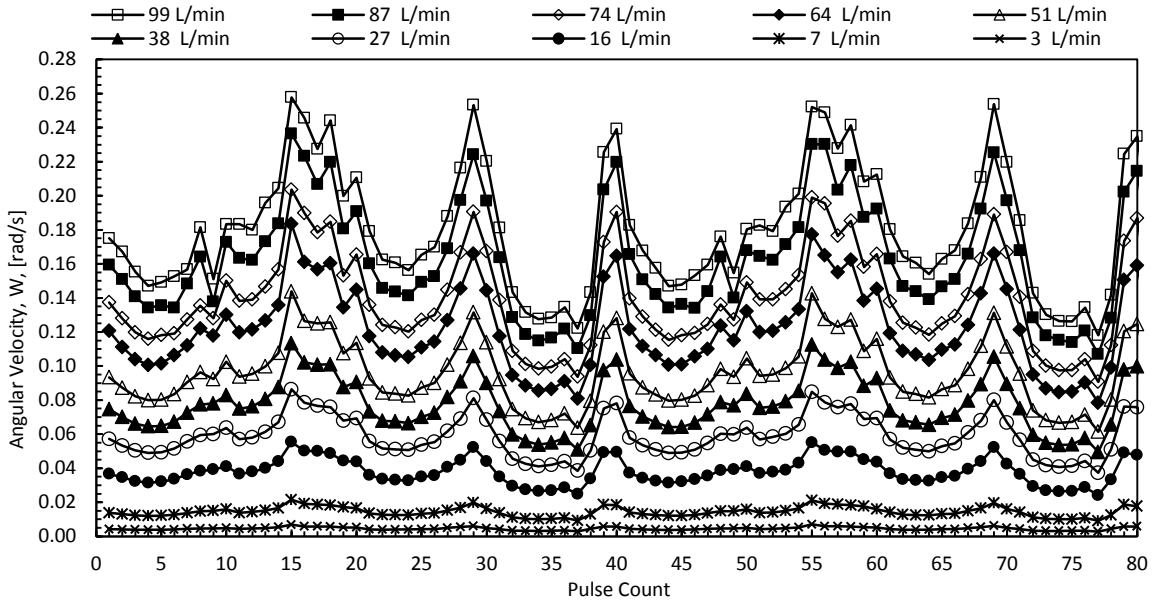


Fig. 3-14 – Angular velocity vs. encoder position

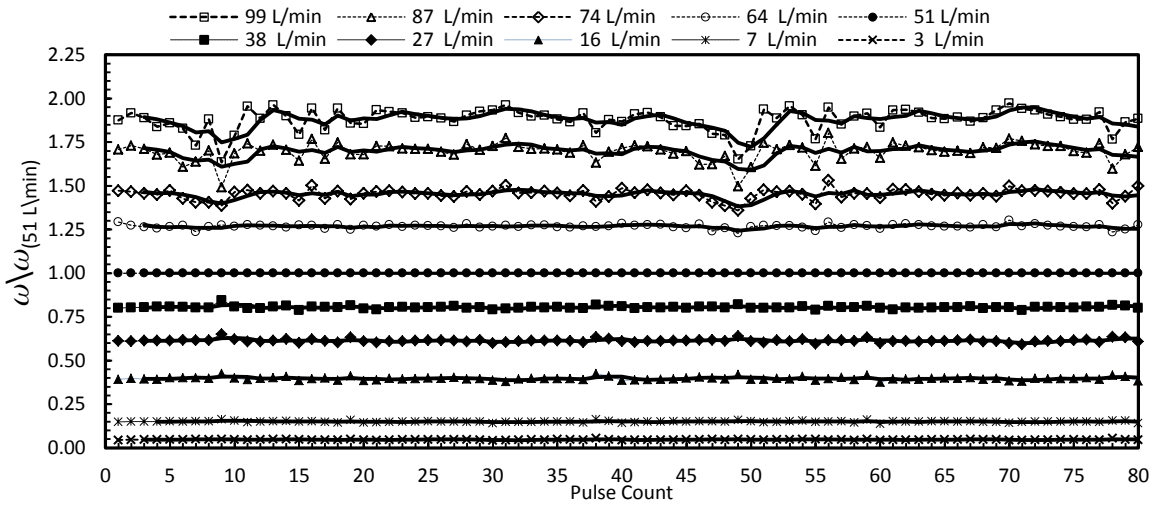


Fig. 3-15 – Normalized angular velocity vs. encoder position

This type of scaling can be done over a full rotation of the internal meter mechanism or for the entire rotation of the output shaft. Using one revolution of the output shaft gives much better results and a middle range can be used as a reference. Finally, it is interesting to see from Fig. 3-13 that this procedure reduces the measurement to the values that were determined earlier for the scaling of the values spatially. This means that these two procedures are complimentary, and they produce a scaled flow rate reading of the final values. The second procedure produces scale factors that are ratios of flow rates and the first procedure shows values that are empirically determined. The successful accounting of the nonlinear motion of the meter mechanism therefore depends on one the linearity of the meter.

3.3.4 Linearity

As it was indicated in section 2.3, the linearity of the meter indicates the deviation of the responses of the flow with respect to a best-fit linear function relating the actual flow output values. The meter is a positive-displacement-type, thus the mechanism speed is expected to scale linearly with the volume flow rate through the meter. To confirm this, the average velocity for one rotation of the index shaft was plotted as function of flow rate, as shown in Fig. 3-14.

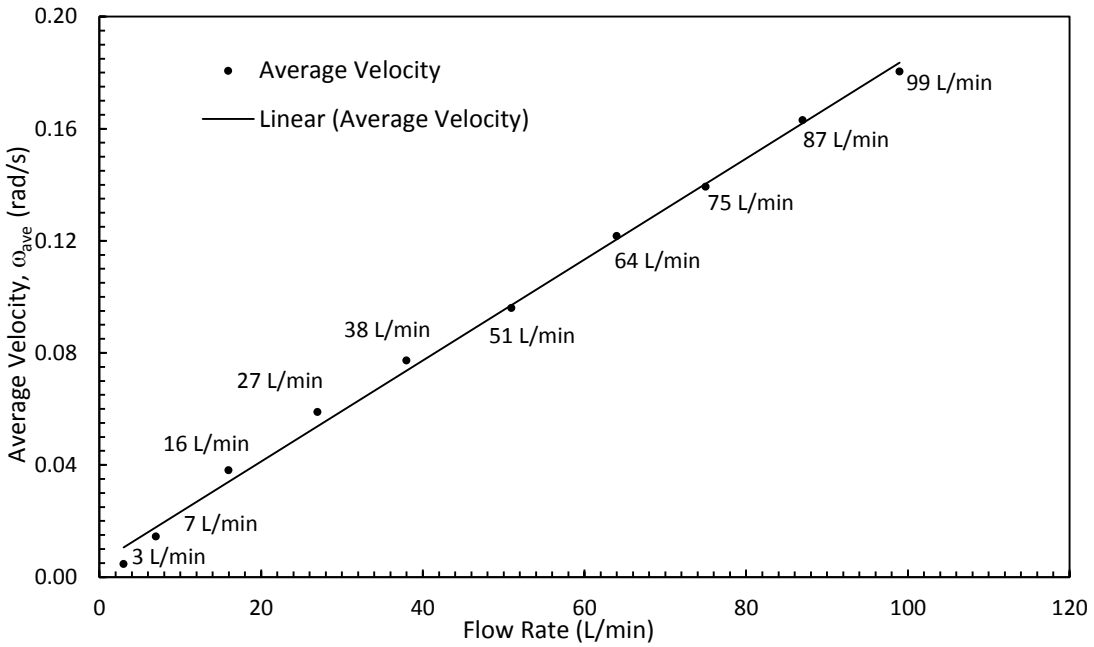


Fig. 3-16 – Average velocity of meter index vs. flow rate

The meter being perfectly linear means that the scale factors using the scaling Fig. 3-16 are just ratios of the flow rate to the reference flow rate. They can be compared to the actual flow rates as shown below. The relationship between the scale factors was found to be linear at the higher flow rates but not for the lower values.

Table 3-8 – Comparison of scale factors

Flow rate (L/min)	Actual scale factor	Estimated Scale Factor (as flow rate ratio)
99	1.88	1.92
87	1.70	1.71
74	1.45	1.45
64	1.27	1.25
51	1.00	1.00
38	0.80	0.76
27	0.62	0.55
16	0.40	0.33
7	0.15	0.14
3	0.05	0.06

If the meter were perfectly linear, the scaling factor should simply be the ratio of the actual flow rate to that of the reference, e.g., the scaling factor for the 73 L/min curve would be 74/51 = 1.45. The estimated scale factors from this approach are also shown in Table 3-8. The measured and estimated scale factors are generally in good agreement, especially in the high flow rate regime of the meter. There is a slight difference in some of the lower flow rates. This may be due friction in the mechanism as a function of mechanism speed or intrinsic variations in the meter accuracy at low flow rates. To compensate for the internal mechanism velocity variation, data similar to that in Fig. 3-14 can be stored in the meter firmware as a table of correction factors specific for each of the individual pulses. This requires that the meter firmware know which pulse is currently being measured, either by having an encoder that has a unique output for each position, or by internal counting with respect to some reference, e.g., the separate zero pulse from the encoder.

CHAPTER IV

ALGORITHM DEVELOPMENT

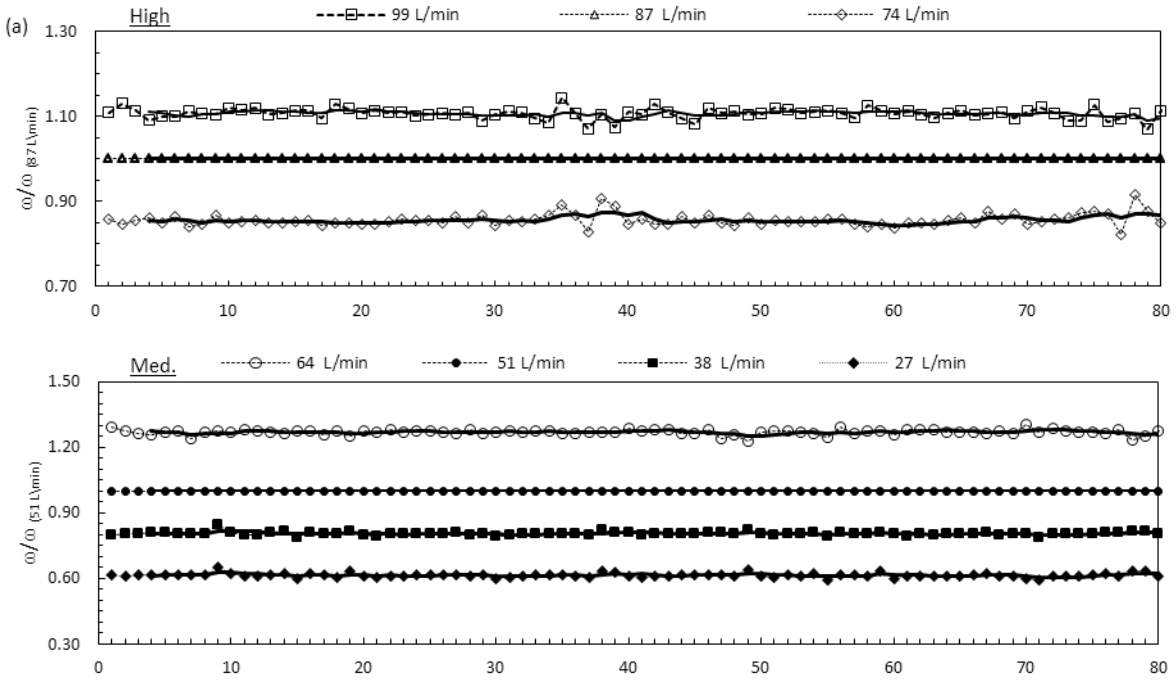
This section describes the development of the algorithms that are used to perform high-resolution meter reading and decomposition of total consumption into individual appliance loads. It also includes a method to teach the meter to automatically recognize the number of appliances inside a home. These algorithms were developed on the Atmel 2560 μC and STK600 development platform using the optical encoder. A three-step approach is utilized to transforming the collected data into gas flow estimates and determining appliance level activity. First, instantaneous velocities are calculated and normalized based on a reference curve. Second, the flow rate is determined using its relationship with the normalized velocity. Finally, the calculated flow rate is used to classify the appliance source.

The meter reading algorithms are first described. An automatic learning mode is that can determine the usage rate of appliances is also presented. Finally, an appliance event detection technique and a performance metric presented.

4.1 High Resolution Meter Reading Algorithm

The development of high-resolution meter reading algorithm is based on the fact the meter motion is repeatable and varies linearly with the flow rate. It involves accounting for the nonlinear motion of the internal meter mechanism. To do this, a reference curve must be loaded into the program. A lookup table representative of a specified flow rate for one full revolution of the output shaft can be used to account for any flow rate by using an appropriate scaling factor. The scale factor can be assumed to be ratio of the flow rates i.e. ratio of times or velocities because the meter flow rate varies linearly with speed.

It can be seen from Fig. 3-15 that using a flow rate of 51 L/min as a reference introduces some fluctuation at the higher flow rates. For more accurate readings, the flow range of the meter is broken up into three different regions. In order to account for a correct flow rate, three lookup tables are used for a low, medium and high flow rate region. This has an added benefit as the flow rate scales more accurately with the mechanism speed in these regions. Then, if a new pulse point falls within any of these regions, a correct scale factor can be applied by linear interpolation to calculate the flow rate. Utilizing three different reference curves for the three regimes gives excellent results. Fig. 4-1 shows these three flow regimes and curves normalized velocities in this range.



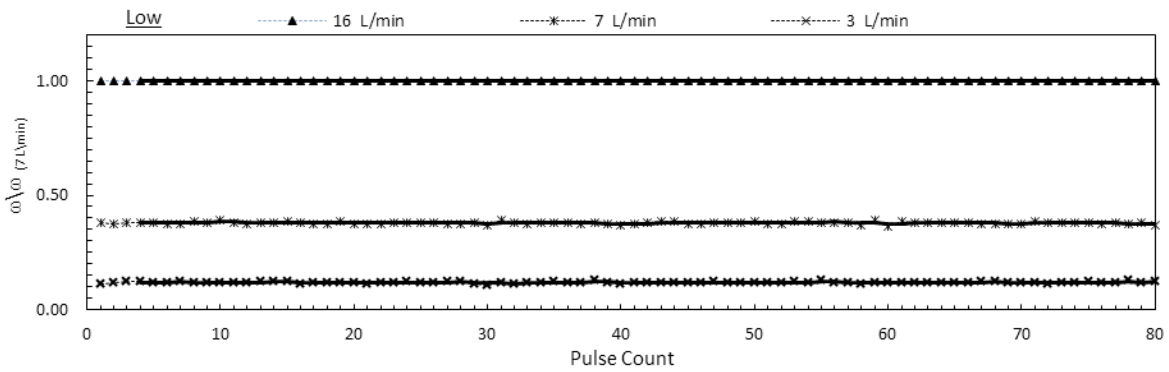


Fig. 4-1 – Normalized velocity vs. angular position at the three different flow regions (a) 87 L\min, (b) 51 L\min, (c) 16 L\min

Since the meter consumes 56.63 L (2 ft^3) per revolution of the index shaft, it is possible to obtain these reference curves by running the meter at a constant flow rate. Once the index shaft reaches its home position, the flow rate can be calculated by,

$$\dot{Q}_{ref} = \left(\frac{1}{18}\right)\frac{56.63\ L}{\sum_{i=1}^{PPR}(t_i)_{ref}} \qquad \textbf{4-1}$$

where, $(t_1,\ t_2\ , t_3, \ldots t_{\mathrm{PPR}})_{ref}$ represent the reference times that to be stored in a lookup table. The following procedure summarizes a simple way to calculate the instantaneous flow rate based on this approach:

a. Wait until encoder reaches zero reference position

b. record the arrival time of an encoder pulse, (t_i)

c. record the arrival time of next encoder pulse, (t_{i+1})

d. calculate the current position of the internal meter mechanism based on encoder position and take the ratio of the time interval between the two most recent pulses with the reference data point corresponding to that position to compensate for the crankshaft variation,

$$\Delta t_i^* = \frac{(t_{i+1} - t_i)}{(t_{i+1} - t_i)_{ref}} \qquad 4\text{-}2$$

e. find the scale factor (*S.F.*) from the appropriate reference curve and multiply the flow rate of the reference curve $\left(\dot{Q}_{ref}\right)$ used in 4 and report instantaneous flow rate, $\dot{Q}(t_i)$

$$\dot{Q}(t_i) = \Delta t_i^* * \dot{Q}_{ref} \qquad 4\text{-}3$$

f. Go to step c and repeat

The code that is used perform these steps is given in Appendix A. Several error checking routines are also implemented. First, an error condition is set to check if the meter is recording the correct number of pulses per revolution. This is checked by running a counter that counts the number of pulses starting from the zero pulse. When a second zero signal is received, this sum is checked against the base PPR of the system to check if the agree. If there is a mismatch, an error flag is raised and the watchdog timer is used to restart the program. A second error check is implemented to check if the sum of the time difference between individual pulses is equal to the time it takes the meter to do a complete full revolution.

4.2 Decomposition of total load

This section describes the determination of the individual appliance usage from the total load. One of the benefits of having high-resolution flow estimates is the ability to decompose the total load into its constituent appliances. Load decomposition requires knowledge of the number and consumption rate of appliances inside a home. This is supplied as an appliances reference table (ART). This table lists the appliance usages rates for each appliance and all the possible combinations. Once the current flow is determined, it is compared to the preexisting flow rates

compiled for the various appliances to classify the sources as individual appliances or combinations of them. In the program, an 8-bit flag is used to store the appliance status, which indicates which appliances are on. Every time a change in flow rate is detected, this flag is updated by referencing the ART.

The ART can be manually programmed during initial setup. There are several ways to do this. The first involves uses the readings from the home's gas meter itself. The second involves reading the flow ratings listed on the appliances, such as a water heater or furnace. Large appliances typically list their gas consumption. A third method is to use the measurements reported on the gas bill. Since the duration of gas usage and its relative flow can be recorded, it can be used calculate the total gas consumed over a period of time. In this way, the first gas bill (or even a set of sparse measurements from the meter) can be used to calibrate the system to appliance consumption rates.

The ART can also be automatically determined by running the meter in learning mode and detecting the changes in appliance activity. This enables the installation this system in the least intrusive manner. In addition, it also eliminates the need to manually update the program every time there are any changes in number and type of appliances in a home.

4.2.1 Learning Mode

The automatic learning mode requires only the number of appliances to get started. It works by first recording the total usage rates and binning similar flow rates together. This is done over a sufficiently long period of time until all possible appliance combinations have been detected and no new flow rate values are registered. Then, by using the existing information, the algorithm identifies the main appliances. From this, the ART can be generated to be used for real-time decomposition of the total load consumption load to the constituent appliance loads.

4.2.2 Appliance Event Detection

Appliance event detection refers to successfully classifying a change in consumption rate. To do this, an exponential moving average is implemented using a single pair of averaging limits (denoted by δ and $-\delta$) The size of this limit (δ) is based on the maximum variation expected from the meter which was obtained through detailed statistical analysis (Section 3.3). If a reading falls within these limits, new data points are averages as they are. If a new reading lies outside area ($\mu \pm \delta$), the averaging criteria is relaxed further. If this continues for two or more consecutive pulses, it is a clear indication that the consumption rate has changed and can be registered as a legitimate appliance event. It is time stamped using a real-time clock and the flow rate is also saved. The averaging parameter is also reset for the next consumption event.

4.2.3 Performance Metric

Since the system relies in the precise encoder positioning of the current encoder pulse with the reference curve, a way should be check if the reference pulses are coincident with the received pulses. This is done by defining a performance metric P,

$$P \stackrel{\text{def}}{=} \left| \frac{\sum_{i=1}^{PPR} \dot{Q}_i(t)\Delta t_i - \mathrm{V}(t)}{\mathrm{V}(t)} \right| \qquad 4\text{-}4$$

where $\dot{Q}_i(t)$ is the current flow rate and $\mathrm{V}(t)$ is the total volume calculated from the meter index for one revolution. If this value is close to 0, the system is performing well, since the sum of the estimated flow volumes is close to the total volume. If this value exceeds a set threshold e, the system needs recalibration and the reference curves need to be reprogrammed.

CHAPTER V

RESULTS AND DISCUSSION

In this chapter, results from different tests that were conducted to test the performance of the meter reading algorithm are presented. Features which are essential for total load decomposition such as learning mode and appliance detection are evaluated for several scenarios and the results from these tests are also presented. In the discussion section, the practical application of this method using a low cost encoder module for real residential gas metering and appliance load monitoring is considered in detail. The overall accuracy of the system is also discussed including potential sources of error that could influence the readings. Installation, calibration and additional benefits such appliance and meter diagnosis are also discussed in detail.

5.1 Results

5.1.1 Meter Reading Algorithm Performance

The high resolution meter reading algorithm is tested for constant on-off (2 state) appliance loads, different constant appliance combinations and constant appliance loads with variable appliances present. The appliances are simulated using the different valve combinations in the experimental setup.

For single, constant appliance loads, only one valve is opened and the flow rate through the meter was collected. The simulated appliances represent a boiler, a water heater and a dryer. The flow rate readings are outputted to screen and also saved to a file. The results for these three appliance loads are plotted in Fig. 5-1. The results show that the program has correctly compensated for the meter mechanism motion and accounted for the scaling of the flow with the proper reference stored in the program. These tests were also conducted for much longer periods (>3 hrs.) to test for drift of the flow rate estimates. It was found to be very stable.

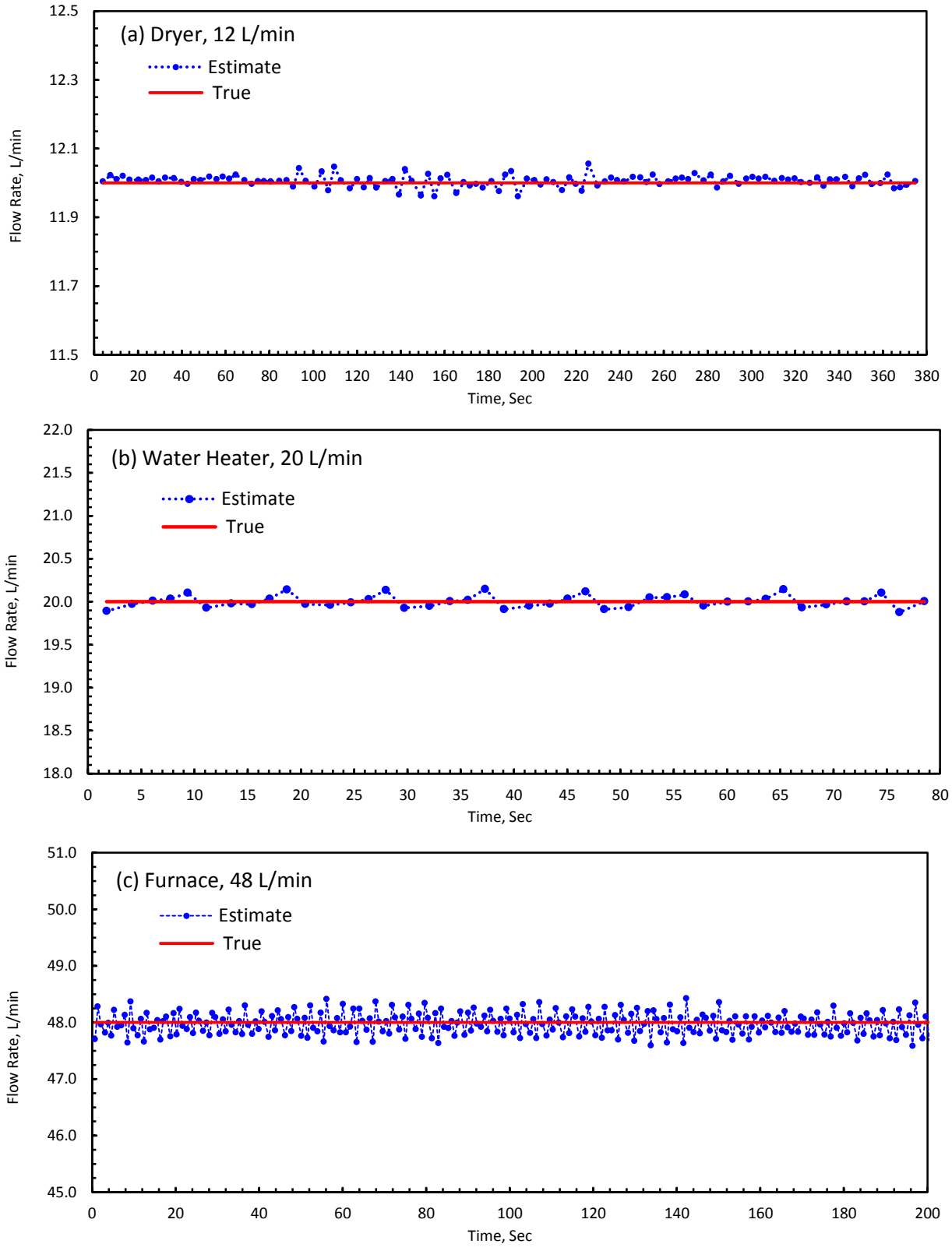


Fig. 5-1 – Results from algorithm at constant appliance loads (a) dryer (b) furnace and (c) water heater

For the combined appliance loads, each valve is turned on and off consecutively. The result from a simple test performed is shown in Fig. 5-2. Each appliance event is annotated as it is turned on and off. Fig. 5-3 shows the total consumption profile.

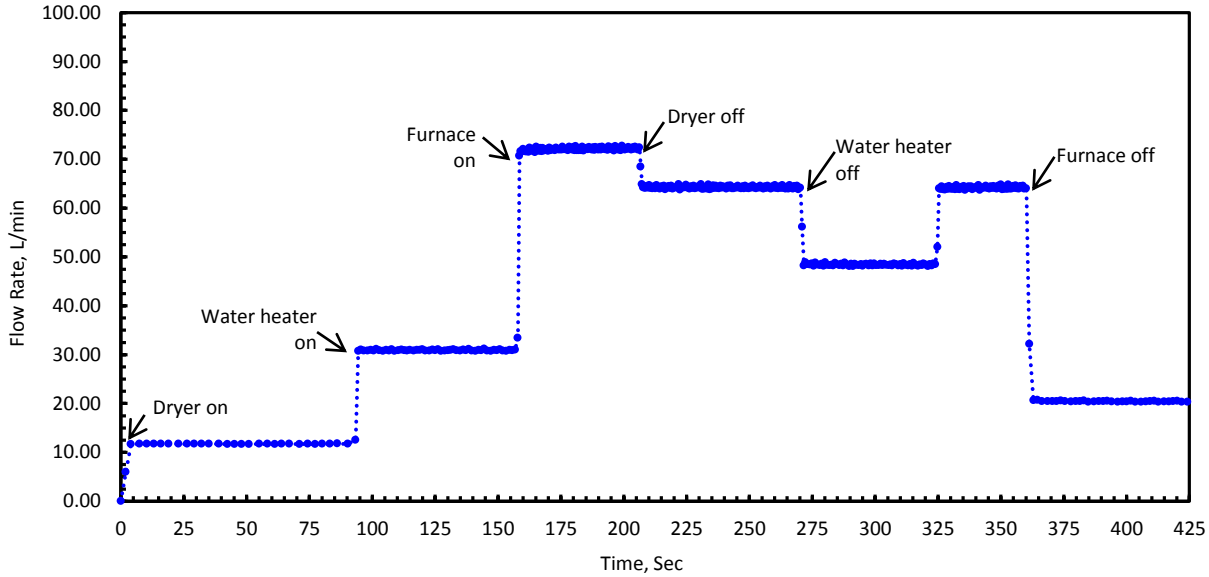


Fig. 5-2 – Results from high resolution meter reading algorithm for different appliance combinations

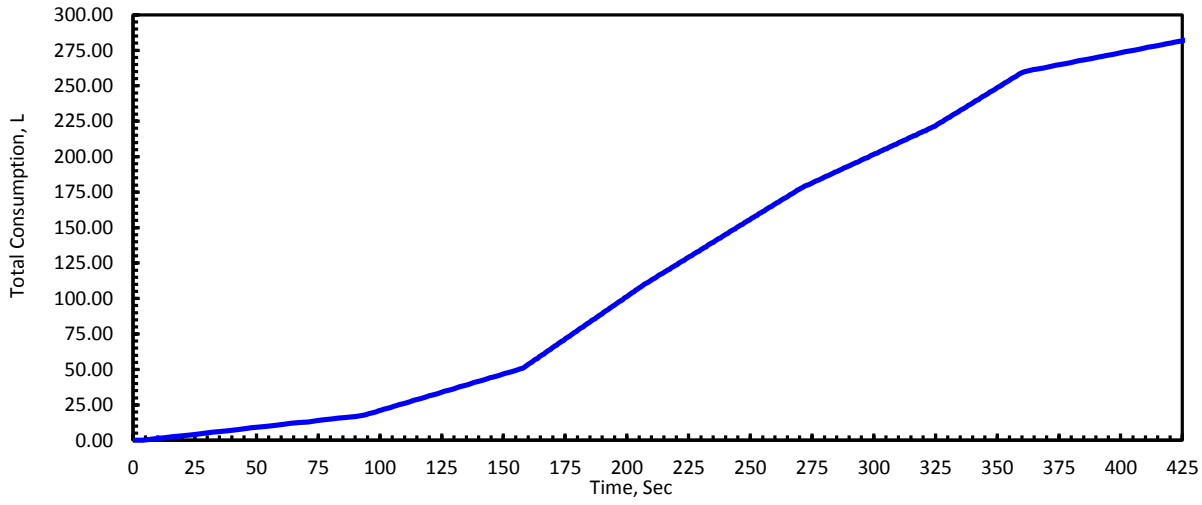


Fig. 5-3 – Total consumption profile for different appliance combinations

The performance of the algorithm in detecting single appliances with variable flow rates was also tested. To do this, the gate valve was used. The assumption is that the variable device will

have three operating loads, low/medium/high. The different flow rates are ramped up through these three stages one by one and also consecutively. Fig. 5-4 shows that the algorithm can effectively capture the effect of variable rate appliances.

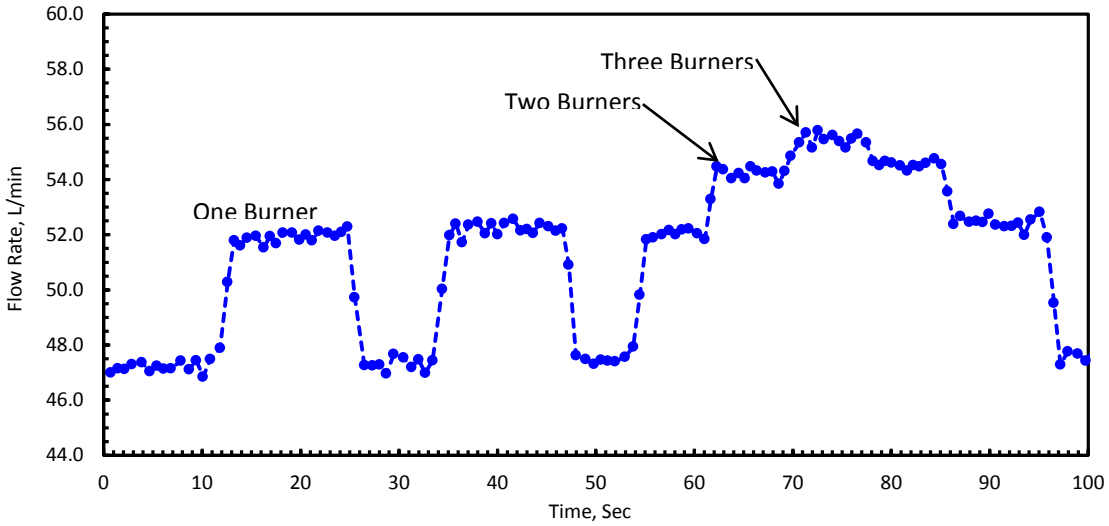


Fig. 5-4 – Results from high resolution meter reading algorithm for variable appliances

Finally, the overall performance is summarized by calculating the average, minimum, maximum and 95% confidence intervals of the results for all these cases. These numbers provide an upper bound of the errors for the flow rate reading of the algorithm. These results are shown in Table 5-1.

Table 5-1 – Summary of algorithm performance constant load

Actual Flow rate (L/min)	Average (L/min)	Minimum (L/min)	Maximum (L/min)	Standard deviation	95% confidence Interval	Omega Flow Meter (L/min)	% Error
12	12.01	11.96	12.06	0.02	±0.03	11	9%
20	20.01	19.88	20.15	0.07	±0.15	19	5%
30	30.73	30.50	30.99	0.11	±0.21	29	3%
48	47.96	47.53	48.38	0.20	±0.40	48	0%
58	57.39	56.84	57.91	0.25	±0.49	58	0%
64	63.98	63.39	64.50	0.21	±0.42	64	0%
72	71.91	71.24	72.57	0.22	±0.44	73	-1%

5.1.2 Total Load Decomposition

As described in section 4.2, decomposing the total load to its constituent appliances requires a learning mode to build the appliance reference table. To do this, an automatic teaching routine was written, which relies on appliance detection based on total flow rate. To evaluate its performance, it was tested for a typical load profile which was simulated by turning on turning the valves on and off in different combinations. The flow rates are visualized in the form of a histogram, with flow rates in the horizontal axis, and number of times detected on the vertical axis as shown in Fig 5-5. The appliances with their known flow rates can be detected from this chart if all possible appliance combinations are recorded. By running the learning mode for a sufficiently long enough period of time, the consumption rates of constant on-off appliances inside a home and can be found to build the ART. Detecting variable rate appliance automatically is difficult. If the primary appliance flow rates and their combinations are subtracted from this figure, the variable flow rates remain.

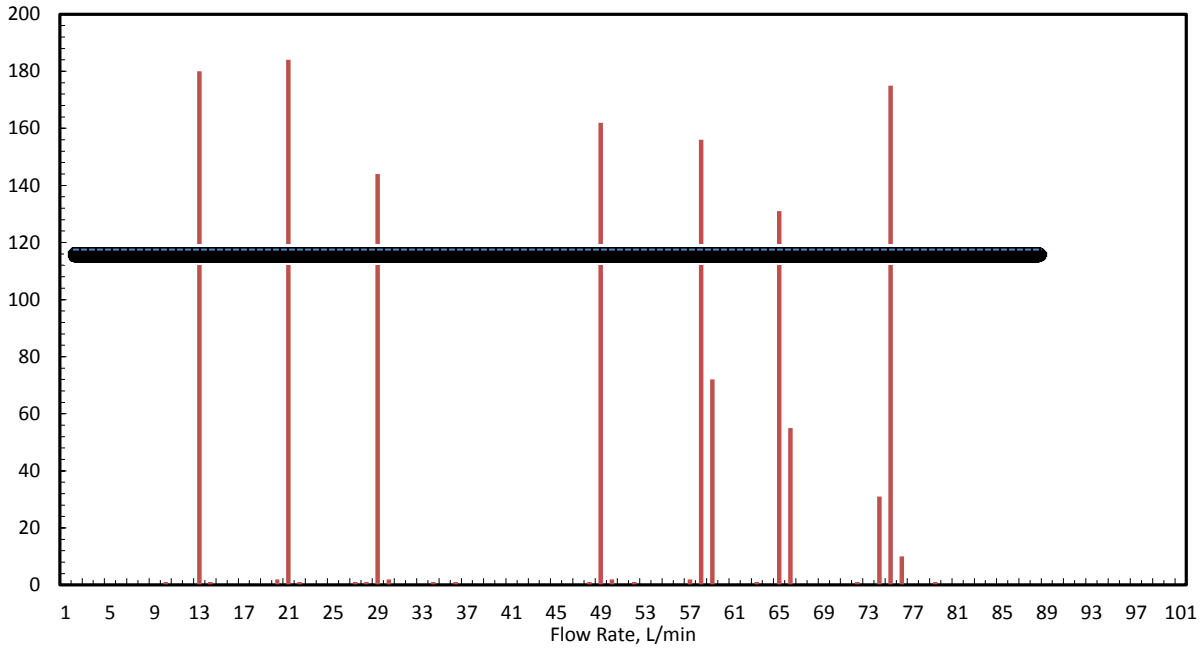


Fig. 5-5 – Evaluation of automatic teaching mode for constant load appliances

5.2 Application to Real Gas loads

5.2.1 Retrofit Module

The successful application of this method requires the use of a shaft encoder that is low in cost and has minimal power consumption to maximize battery life. The encoder must also be integrated into a module that is accommodating for retrofit installations and perform reliably in exterior environments under extremes of temperature, humidity, and precipitation. These requirements rule out optical encoders, which are expensive, power intensive and have a limited temperature and humidity range for proper operation. Alternatives include reed switch or mechanical encoder designs in which negligible electrical power is required.

A prototype of a low cost encoder was designed and built another student working on this project. The design is developed using magnetically actuated reed switches to generate an interrupt for the microcontroller when the switch is closed. Due to size and interaction between switches, they cannot be packed to give a high resolution like an optical encoder. Therefore, a compound gear train is necessary to increase the angular velocity of the meter output shaft to compensate for the lower resolution. The cost of this encoder depends on the design of this gear train. A high gear ratio creates problems in design, assembly and introduces friction in the output. To minimize the cost of this design, it is necessary to use the lowest gear ratio possible. Therefore, it is important to find the minimum PPR that must be used to perform all the functions developed in this method. Since this is an optimization problem, a load simulation is performed to find the minimum PPR for the mechanical encoder.

5.2.2 Load simulation

Finding the optimum PPR for the encoder design requires a real gas usage data of a residential home. This data can be obtained by installing sensors on all the gas consuming appliances in a

residential home. While this would provide good experimental data, it is very difficult to perform. As an alternative, an extensive survey of the existing literature was performed to find various studies involving sub-metering appliance gas use in residential settings. These studies included data of performance measurements of homes side by side, energy demand monitoring projects, hot water usage in domestic residential buildings, weather sensitivity studies on appliances etc. [39-43]. Based on a detailed analysis of the information obtained, a residential gas consumption profile is created to simulate the realistic metering requirements.

The validity of this information for simulating gas loads can be justified by considering the long term and short term appliance activities separately. Long term appliance usage patterns were available from numerous studies aimed at understanding the daily usage profiles of appliances and classifying their seasonal patterns. This coarse data displays the general usage pattern of major appliances on an hourly basis. For example, furnace loads have longer periods between cycles while water heaters loads are much more frequent. Using these patterns, a usage profile is generated for different appliances. For short term appliance patterns, high frequency usage measurements were also available from several studies. Some examples include furnace operation data from a residential home that was taken every 5 minutes and dryer loads taken every second. The consumption rate of the appliances intervals more than the simulation is assumed to be constant.

The information from these studies is compiled in order to construct a gas consumption profile of appliances and this is shown in Fig. 5-5 (a) with some appliance activity annotations. The constructed usage profile is for 24 hours starting from *12:01* am to *11:59* pm and the frequency is every second. The entire data set includes 86,400 data points of total consumption and shown in Fig. 5-5(b). The total load profile is a combination of individual appliance profiles.

The combinations of the different loads were also multiplied by a pressure loss factor to reflect the reduction of a combined load from the independent sum of the two different loads. The details of this procedure are given in Appendix B. This profile is used to determine the minimum PPR necessary for the mechanical encoder design. This is done by simulating the meter and using this load profile to find at what minimum PPR it will be possible to capture all the usage details.

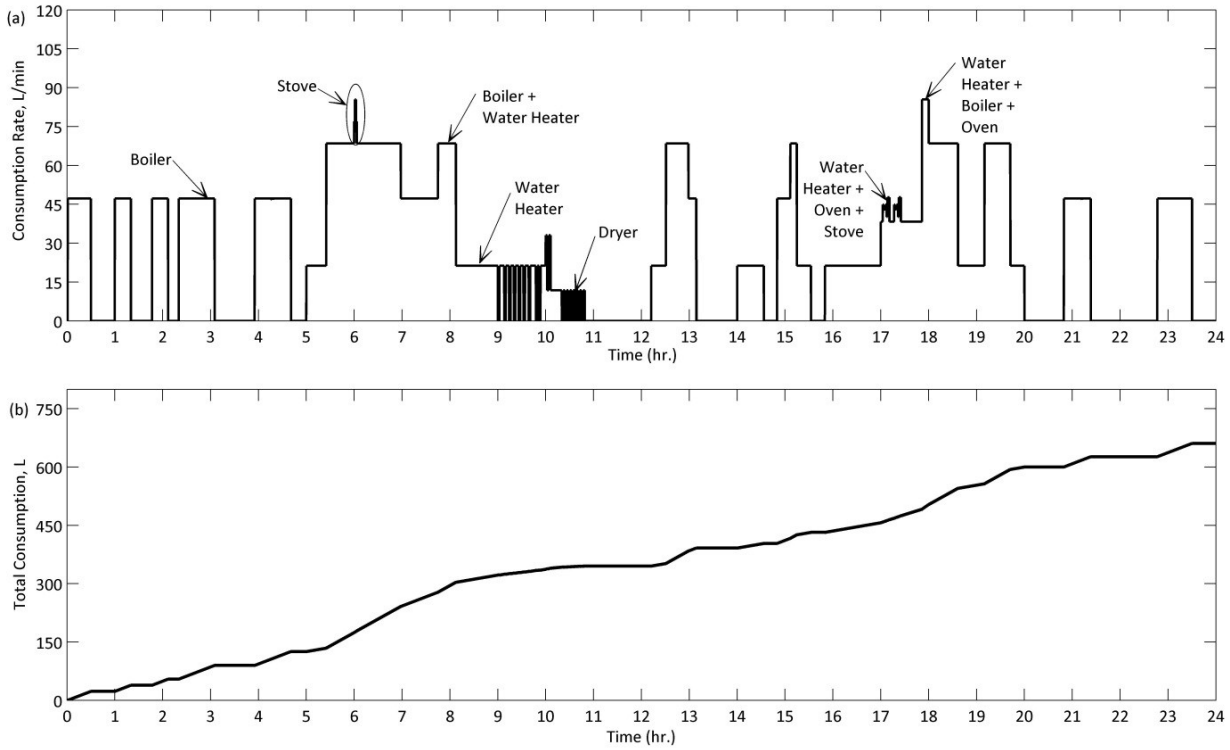


Fig. 5-6 – (a) Usage data for different appliances and (b) total usage over a 24 hour period

The steps of the simulation are shown in Fig 5-6. The real usage data is first run through a model of the meter. The model generates pulses for a 24 hour period at different number of pulses per revolution of the mechanical encoder. These pulses are then used as input to the meter reading algorithm to calculate the usage data. Finally, it is compared with the original data to pick the optimum PPR for the encoder.

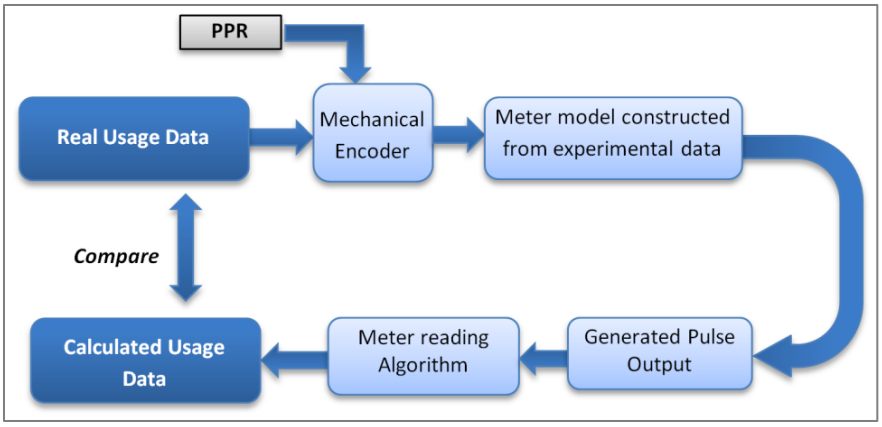


Fig. 5-7 – Schematic of simulation steps

A MATLAB® model of the meter was built as function of encoder PPR. The complete procedure is given in Appendix B. The model is based on the basic motion profile of the internal meter mechanism and the relationship between meter flow rate and shaft velocity. A flow rate of 49 L/min is used as the reference point. The input flow rates, the meter shaft velocity for at the ith pulse is estimated by $\omega(i) \cong Q(i) * \frac{\omega_{ref}}{Q_{ref}}$ or using the time interval between the two most recent pulses, $t(i) \cong Q(i) * \frac{Q_{ref}}{t_{ref}}$, where $Q(i)$ is the flow rate at that instant. In this way, the model generates output pulses from the meter model corresponding to the usage profile.

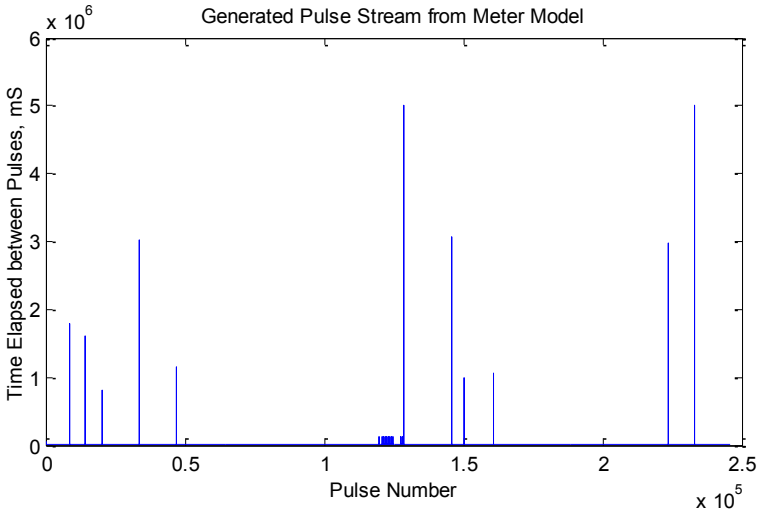
The model takes into account the variations in time measurements due to the motion of internal meter mechanism. To cover the entire range of uncertainty, the maximum, minimum and average values for the measured time differences are used. They were incorporated into the model by adding a uniformly distributed component bounded between the minimum and maximum time intervals between two consecutive pulses. This added random component to the pulses incorporates all of the uncertainty from the actual meter into our model.

The meter model is independent of the PPR at which these experimental measurements that are obtained. Instead of taking measurements at each desired PPR, the measurements at 720PPR

are down sampled to 18 PPR in multiples of 18. This corresponds to 1 to 40 points per revolution of the internal meter mechanism. The maximum used was 720. A minimum of 18 PPR must be used for this high resolution metering technique to work.

For periods when a device was on, the model generates the pulses that would occur during that time depending on the PPR used. If the PPR values require timing between pulses to be longer than 1 sec, a branch is used to increment the time period until the minimum time is found. Pulses are then generated for that time period and the remainder is saved for the next time step. The times differences between each pulse counts is stored for later processing.

The number of pulses that are generated from the model depends on the PPR value that is used. The pulse generation was performed for a 24 hour period for different values of PPR. For example, Fig. 5-8(a) shows the pulses generated by the simulation for a 360 PPR. In this case, 250,000 individual pulses have been generated from the model for a 24 hour period. Periods with no appliance activity produce very long pulses. These periods can be removed from the plot to get a clear picture of the generated pulses Fig. 5-8 (b).



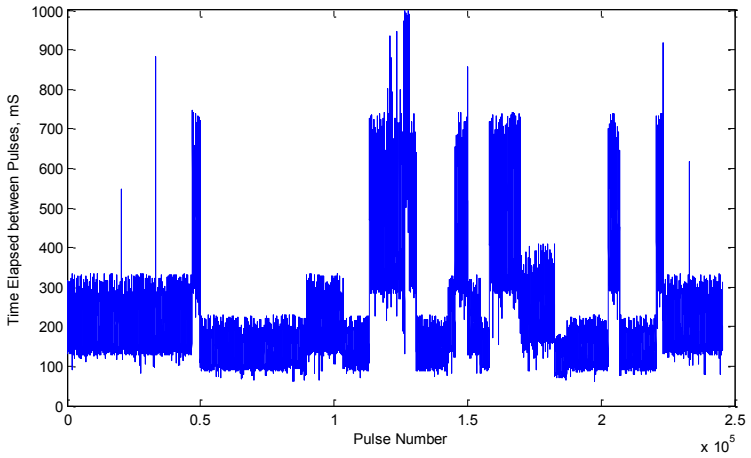


Fig. 5-8 – (a) generated pulse stream and (b) pulse stream for active periods only from meter model at 360 PPR

In order to find the optimum encoder PPR, the generated pulses for each PPR are used as inputs for the meter reading algorithm. An encoder with a very low PPR can result in the loss information for usage events occurring within an interval that is less than the sampling period. Using an encoder with a very high PPR can increase the metering resolution. However, the resulting noise in the signal is increased significantly because more points are sampled during the motion of the internal meter mechanism. In addition, if averaging methods are used to smooth the data simply, that is similar to reducing the resolution. By comparing the output data with the real usage data to find the minimum relative error between them, it was found that 90 PPR is an optimum value to capture all the usage data and is shown in Fig. 5-9 and Fig. 5-10. This value was also chosen for the design of mechanical encoder.

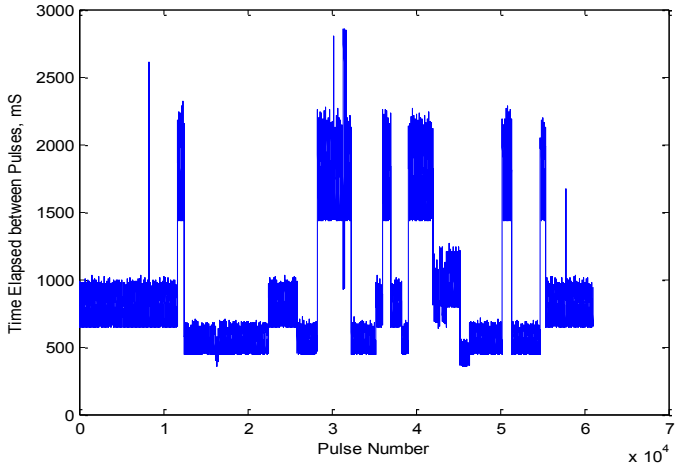


Fig. 5-9 – Generated pulse output at 90 PPR

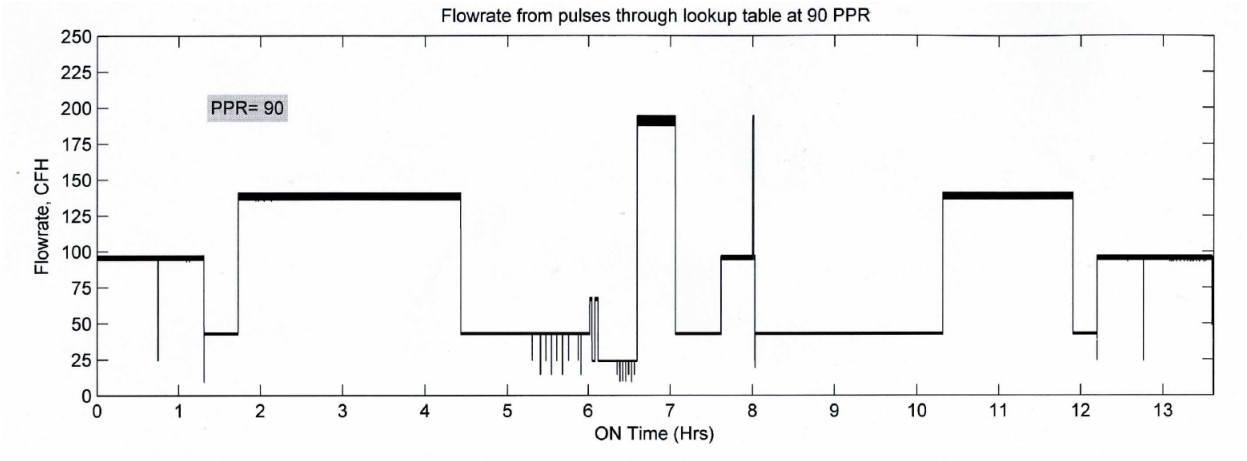


Fig. 5-10 – Simulated load for load profile at 90 PPR

5.3 Overall System Accuracy

The overall uncertainty of the total gas volume reported by the meter retains the uncertainty of the meter itself (± 1%, per the manufacturer), since the encoder system only serves to track the rotation of the meter output shaft and does not otherwise alter the metering mechanism in any way. Uncertainty sources from encoder mounted can be avoided by carefully mounting the encoder to the output shaft as discussed in section 2.5.2. The uncertainty in the instantaneous reported flow rate from this system rate includes uncertainty from the internal meter mechanism as well as that from the flow rate correction and internal meter position corrections. Flow rate

uncertainty will determine the smallest flow difference that can be resolved, which becomes important for two appliances that have similar flow rates.

Another important factor to consider is the sampling rate. Because the shaft velocity is linearly related to the flow rate, the sampling rates also depend on the flow rate. This is essential at very low flow rates, because the sampling rates may be too small detect appliance events. The number of pulses per revolution as a function of flow rate can be easily calculated:

$$X\ \text{cu.ft.} / \text{hr} * 1\ \text{rev} / 2\ \text{cu.ft.} * 1\ \text{hr}/3600\ \text{sec} * Y\ \text{pulses}/\text{rev} = Z\ \text{pulses/sec} \qquad 5\text{-}1$$

This shows that the pulses per revolutions required is directly related to the number of pulses/sec that are available at a specified flow rate. Therefore, the PPR must be chosen appropriately to be able to capture the lowest flow rate that needs to be detected. This value sets the lower bound in the selection and design of the encoder module.

Table 5-2 – Sampling Rate at 90 PPR

Flow Rate, $ft^3/hr.$	Flow Rate, L/min	Frequency, Hz	Time Period, S
250	118	3.125	0.3
200	94	2.5	0.4
150	71	1.875	0.5
100	47	1.25	0.8
75	35	0.9375	1.1
50	24	0.625	1.6
25	12	0.3125	3.2
15	7	0.1875	5.3
10	5	0.125	8.0
5	2.5	0.0625	16.0
1	0.5	0.0125	80.0

For the mechanical encoder, at the chosen 90 PPR, based on 5-1, it provides 0.0125 pulse/sec for a flow rate of 1 ft^3/hr. Table 5-2 shows the sampling frequency and time between consecutive pulses as function of flow rate. For example, for a flow rate of (0.5 L/min) 1 ft^3/hr. it would take 80 sec to get one pulse from the meter. If this represents a leak in the system, it can detect easily. For a period 1hr, 45 pulses will be collected from the meter. With current AMR solutions, this event would not be registered at all. With this system, not only can the event be registered, its flow rate can be calculated to within 1%.

CHAPTER VI

CONCLUSIONS AND FUTURE DIRECTIONS

6.1 Conclusions

This work presents the development of a high-resolution meter reading system that uses a retrofit concept to convert the mechanical motion of the meter output shaft into actual flow rates. The method utilizes the signature of constant on-off appliances in order to decompose the total load into individual appliance usage. The basic concept of the system is presented in Chapter 1

A detailed description of the diaphragm type positive displacement gas meters and its measurement properties is given in Chapter 2. The diaphragm gas meter used in this work is also described. The meter principle of operation and the internal meter mechanism is also described in detail as well as parameters that are important to measure, such as meter such as uncertainty, repeatability and linearity. The experimental setup that is used to simulate a typical gas supply system inside a home and all the various components are presented. The appliances that are simulated are a stove, a water heater, a boiler and a dryer. The mounting of high resolution encoder to the output shaft of the meter is also discussed in detail. Meter modifications in order to perform video measurements of the internal meter mechanism are provided.

The experimental results are presented in chapter 3. The proper operation of the meter is verified by proofing the meter with a separate flow meter and measuring the pressure drop across the meter and comparing the results with the manufacturers' specification. The internal meter mechanism motion profile recorded from the video measurements is also given. The results show that the internal meter mechanism goes through a repeatable motion with four cycles of speeding up and slowing down as a result of the movement of diaphragms. This is identified as the primary cause of the non-linearity of the meter mechanism. Next, results from the encoder data

are given. Since the encoder measurements are made at very higher resolution (720 PPR), it is used to visualize the velocity profile of the internal meter mechanism from the motion of the output shaft. The results also show that the motion profiles are similar with the meter mechanism executing the same motion at a speed that increases linearly with increasing flow rate. The timing data shows that the average period of the meter mechanism and the output shaft varies by a maximum of 1% at lowest flow rate. In order to compensate for the non-linear motion of the meter mechanism, the experimental data is analyzed for repeatability and linearity. This analysis is carried out by performing a student's t-test on data collected for a single flow rate and at different times. The analysis shows that the repeatability of the internal meter mechanism is ~±3% and the repeatability of output shaft is ~±1%. The analysis of linearity of the meter shows that higher flow rates scale linearly to meter flow rate with slight variation at low flow rates.

Details of the algorithm that are developed using the experimental results are given in chapter 4. Three reference curves stored in firmware are utilized to compensate the nonlinear motion of the meter. The linearity of the meter within these three flow rate ranges is used to perform linear interpolation of the instantaneous velocity to determine the instantaneous flow rates. This can be done to any desired resolution depending on the encoder PPR used. Using a very high resolution encoder can result is larger variation of the final result as more points of the internal meter mechanism are sampled.

In order to implement this method as a low cost encoder module, a mechanical encoder with reed switches and a gear train is utilized. The cost of a mechanical encoder module is directly related to the PPR needed. In order to minimize the cost, load simulation is carried out in order to determine the minimum PPR required. The analysis is done on real usage data collected from literature. The simulation results show that 90 PPR is an optimum PRR.

In Chapter 5, the results from algorithm performance test at 90 PPR are presented. The algorithm that is developed that can perform high-resolution measurements with the same uncertainty as the meter (~±1.0%). The method is also analyzed for the development of low cost encoder utilizing this concept. Finally, the sampling rates at 90 PPR and the potential sources of error for the system are described.

6.2 Future Work

One potential issue that remains to be addressed is how the mechanical behavior of the meter changes with age and use. In this work the meters were new, however the meter behavior after one, two, or three decades of continuous use may drift from its as-new value. Provisions for recalibration or accounting for meter aging would need to be incorporated into a final device design.

Variable-flow gas-consuming devices, such as a gas cooktop or gas grill, also present a challenge for decomposition, because their usage signatures do not follow a simple binary pattern. Finally, if only a single variable-flow appliance exists in the house, then the contribution from this device can be determined from the total volume less the contributions from all other devices. However, if more than one variable rate appliance is inside a home, it is difficult to decompose. A sensor could also be outfitted to the appliance, but this detracts from the simplicity and non-intrusive nature this concept.

Finally, if two appliances have similar flow rates, e.g., within 1–2 L/min, this method may be unable to distinguish between them. One approach would be to incorporate time-of-day or patterns of usage data to attempt to differentiate which appliance is on. A wireless sensor could also be placed on one of the appliances to indicate when it is consuming gas, although this represents an additional cost and complexity. Another approach would be to simply forgo the

desire to separate the usage for each appliance in this case and report their combined usage as a single number. Large or intermittent gas leaks could potentially alter the usage rates enough to confuse the detection algorithms, which would result in higher read errors. Future opportunities for research include the long-term performance studies.

REFERENCES

[1] V. Chandra. (2006). *Fundamentals of Natural Gas*.

[2] EIA, "Natural gas annual 2009," United States DOE/EIA-0131(09), December 2010.

[3] R. A. Kerr. (2010, 25 June 2010) Natural Gas From Shale Bursts Onto the Scene. *Science* [News Focus]. pp. 1624-1626

[4] T. D. Tamarkin. (1992, October) Automatic Meter Reading. *Public Power Magazine*.

[5] J. L. Fischer, "Automatic meter reading module," United States Patent, Nov. 7, 2002.

[6] R. K. Payne and W. A. Lien, "Automated meter reader direct mount endpoint module," United States Patent, Feb. 17, 2011.

[7] S. Cumeralto and R. DeVries, "Method and apparatus for collecting and displaying consumption data from a meter reading system," United States Patent, 2008.

[8] Younghun Kim, Thomas Schmid, Zainul M. Charbiwala, and M. B. Srivastava, "ViridiScope - Design and implementation of a Fine Grained Power Monitoring System for Homes," presented at the UbiComp 2009, Orlando, Florida, 2009.

[9] D. H. Wilson and C. Atkeson, "Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors," presented at the Proceedings of the Third international conference on Pervasive Computing, Munich, Germany, 2005.

[10] Alice M. Agogino, Jessica Granderson, and S. Qiu, "Sensor Validation and Fusion With Distributed 'Smart Dust' Motes for Monitoring and Enabling Efficient Energy Use," in *AAAI Technical Report SS-02-03*, Canada, 2002.

[11] M. L. Marceau and R. Zmeureanu, "Nonintrusive load disaggregation computer program to estimate the energy consumption of major end uses in residential buildings," *Energy Conversion and Management,* vol. 41, pp. 1389-1403, 2000.

[12] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE,* vol. 80, pp. 1870-1891, 1992.

[13] G. W. Hart, "Residential energy monitoring and computerized surveillance via utility power flows," *Technology and Society Magazine, IEEE,* vol. 8, pp. 12-16, 1989.

[14] G. W. Hart, E. C. Kern Jr., and F. C. Schweppe, "Non-intrusive appliance monitor apparatus," United States Patent, 1989.

[15] S. B. Leeb, S. R. Shaw, and J. L. Kirtley, Jr., "Transient event detection in spectral envelope estimates for nonintrusive load monitoring," *Power Delivery, IEEE Transactions on,* vol. 10, pp. 1200-1210, 1995.

[16] L. Farinaccio and R. Zmeureanu, "Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses," *Energy and Buildings,* vol. 30, pp. 245-259, 1999.

[17] C. Laughman, L. Kwangduk, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Power signature analysis," *Power and Energy Magazine, IEEE,* vol. 1, pp. 56-63, 2003.

[18] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, 2011, pp. 239-240.

[19] J. Lei, L. Jiaming, L. Suhuai, J. Jin, and S. West, "Literature review of power disaggregation," in *Modelling, Identification and Control (ICMIC), Proceedings of 2011 International Conference on*, 2011, pp. 38-42.

[20] J. Fogarty, C. Au, and S. E. Hudson, "Sensing from the basement: a feasibility study of unobtrusive and low-cost home activity recognition," presented at the Proceedings of the

19th annual ACM symposium on User interface software and technology, Montreux, Switzerland, 2006.
[21] Y. Kim, T. Schmid, Z. M. Charbiwala, J. Friedman, and M. B. Srivastava, "NAWMS: nonintrusive autonomous water monitoring system," presented at the Proceedings of the 6th ACM conference on Embedded network sensor systems, Raleigh, NC, USA, 2008.
[22] E. Larson, J. Froehlich, T. Campbell, C. Haggerty, L. Atlas, J. Fogarty, and S. N. Patel, "Disaggregated water sensing from a single, pressure-based sensor: An extended analysis of HydroSense using staged experiments," *Pervasive and Mobile Computing,* vol. In Press, Corrected Proof, 2010.
[23] S. N. Yamagami, H, "Non-Intrusive submetering of residential gas appliances," *Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings,* pp. pp. 265–273, 1996.
[24] G. Cohn, S. Gupta, J. Froehlich, E. Larson, and S. Patel, "GasSense: Appliance-Level, Single-Point Sensing of Gas Activity in the Home," in *Pervasive Computing*. vol. 6030, P. Floréen, A. Krüger, and M. Spasojevic, Eds., ed: Springer Berlin / Heidelberg, 2010, pp. 265-282-282.
[25] S. Marvin, H. Chappells, and S. Guy, "Pathways of smart metering development: shaping environmental innovation," *Computers, Environment and Urban Systems,* vol. 23, pp. 109-126, 1999.
[26] B. Neenan and R. C. Hemphill, "Societal Benefits of Smart Metering Investments," *The Electricity Journal,* vol. 21, pp. 32-45, 2008.
[27] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?," *Energy Efficiency,* vol. 1, pp. 79-104, 2008.
[28] D. Mountain, "The impact of real-time feedback on residential electricity consumption: the Hydro One pilot," *Mountain Economic Consulting and Associates Inc., Ontario,* 2006.
[29] F. Geraldine. (2009) Technology-Enabled Feedback on Domestic Energy Consumption: Articulating a Set of Design Concerns. 37-44. Available: http://doi.ieeecomputersociety.org/10.1109/MPRV.2009.17
[30] G. Wood and M. Newborough, "Influencing user behaviour with energy information display systems for intelligent homes," *International Journal of Energy Research,* vol. 31, pp. 56-78, 2007.
[31] J. G. Webster, *The Measurement, Instrumentation, and Sensors Handbook*: CRC Press, 1999.
[32] I. Bluvshtein, "Uncertainties of Measurement Instruments," *Pipeline & Gas Journal,* 2007.
[33] N. Ford, "Fundamental principle of Diaphargm Meters," 2005.
[34] R. C. Baker, *Flow Measurement Handbook*: Cambrige University Press, 2000.
[35] U. R. C. Nilsson and J. Delsing, "In situ detection of inaccurate gas flow meters using a fingerprint technique," *Flow Measurement and Instrumentation,* vol. 9, pp. 143-152, 1998.
[36] L. Peksa, T. Gronych, M. Vičar, M. Jeřáb, P. Řepa, J. Tesař, D. Pražák, Z. Krajíček, and F. Staněk, "Method of measuring the change in volume of a diaphragm bellows used in a volume displacer of a constant-pressure gas flowmeter (with a practical guide)," *Measurement,* vol. 44, pp. 1143-1152, 2011.

[37] H. O. Johansen. (1958) How Your Gas Meter Works. *Popular Science*. 110-112. Available: http://books.google.com/books?id=SC0DAAAAMBAJ&pg=PA110&dq=popular+science+gas+meters&hl=en&ei=jWWYT7yHAY3G0AGVvdCBBw&sa=X&oi=book_result&ct=book-thumbnail&resnum=1&ved=0CEgQ6wEwAA#v=onepage&q=popular%20science%20gas%20meters&f=false

[38] D. Brown. (2009) Tracker Video Analysis and Modeling Tool. Available: http://www.compadre.org/Repository/document/ServeFile.cfm?ID=7365&DocID=517

[39] G. Wood and M. Newborough, "Dynamic Energy-Consumption Indicators for Domestic Appliances: Environment, Behaviour and Design," *Energy and Buildings,* vol. 35, pp. 821-841, 2003 2003.

[40] T. Ueno, F. Sano, O. Saeki, and K. Tsuji, "Effectiveness of an Energy-Consumption Information System on Energy Savings in Residential Houses Based on Monitored Data," *Applied Energy,* vol. 83, pp. 166-183, 2006 2006.

[41] K. W. Warren, "Determining the Impact of Residential Gas Furnaces on Utilities with Applications to Other End Uses," MS, Mechanical Engineering, University of Wisconson - Madison, Madison, 1993.

[42] T. J. Lui, W. Stirling, and H. O. Marcy, "Get Smart," *IEEE Power & Energy Magazine,* p. 13, June 2010 2010.

[43] J. Menkedick, "Metered Ranges, Cooktops and Ovens in the Northern Illinois Gas Residential Load Study Data Base," Gas Research Institute, Chicago, IL GRI-93/0204, 1993.

Appendix A

```
/*****************************************************************
This program was produced by the CodeWizardAVR V2.03.9 Evaluation
Automatic Program Generator
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project : METER READING SOFTWARE
Version : 1.5.0
Date    : 8/09/2010
Author  : Mahder Tewolde
Company : Stony Brook University
Comments: Collect pulse widths (time periods) from an optical
encoder and reports the instantaneous flowrate as well as the
total consumption.

Chip type                   : ATmega2560
Program type                : Application
AVR Core Clock frequency    : 18.436306  MHz
Memory model                : Small
External RAM size           : 0
Data Stack size             : 2048

*****************************************************************
MAJOR REVISION HISTORY

Version 1.00 8/09/2010:  Performs meter reading functionalities
                         Utilizes 2-Channels from Encoder for 720PPR
                         Error Reporting Capabilities
                         Formatted Printing to 'COM1' for debugging
Version 1.20 12/3/2010:  Added Teaching Mode Functionality
                         Averaging Techniques Introduced
Version 1.30 2/13/2011:  Added Real Time Clock Functionality
                         Automatic Teaching Mode Implemented
Version 1.40 3/17/2011:  Reconfigured to 360 PPR
                         Removed unused functions
Version 1.50 7/17/2011:  Updated Project Name for AVR Studio 5 compatability
                         Added Event Based Usage Information
*****************************************************************
KNOWN ISSUES
1. Timer date is incorrect and needs to be restored when chip is reset
2. Automatic Teaching Mode not complete
*****************************************************************
WIRED CONNECTIONS TO STK600

LED's connected to PORT A of the STK600
Channel_GZ of encoder is connected to PIND.0
Channel_B+ of encoder is connected to PIND.1
IO_Pins (SDI/SDO) from DS1305 is connected to PINB.0
SCLK Pin from DS1305 is connected to PINB.2
CE Pin from DS1305 is connected to PINB.4
*****************************************************************/
#asm
   .equ __ds1302_port=0x05      ;PORTB
   .equ __ds1302_io=2           ;DS1305 PINS 12 & 13: SDO/SDI
   .equ __ds1302_sclk=1         ;DS1305 PIN 11: CLK
   .equ __ds1302_rst=4          ;DS1305 PIN 10: CE
#endasm

/*System Wide Header Files*/
#include <mega2560.h>
#include <sleep.h>
#include <stdlib.h>
#include <math.h>
#include <delay.h>
#include "meter.c"
#include "error_handling.c"
#include "ds1305.c"
#include "appliances.c"
```

```
/*System Wide Constants*/
#define LED_STATE PORTA
#define CPU_FREQ_MHZ 18.436306
#define roundf_10(x) ((int)(x*10+0.5))/10.0 //round to 10th
#define roundf_100(x) ((int)(x*100+0.5))/100.0 //round to 100th
#define min(a,b) a>b?b:a
#define max(a,b) a>b?a:b

#define DEBUG_ON
/*Including the DEBUG_ON on setting will cause the debug
flag to be enabled. This allows the compiler to include the
additional library and lines of code required for debugging*/
#ifdef DEBUG_ON
#include <STDIO.H>
unsigned char int_num=0;
#define TEST_PIN PORTF.0
#define DPRINTF 1
#else
#define DPRINTF 0
#endif
#define debug_print(...) if (DPRINTF) printf(__VA_ARGS__)

//System Wide Variables
enum pulse_type {LONG_PULSE,SHORT_PULSE} get_type ;
struct usage_profile {
            unsigned long time_stamp;
            unsigned char f_rate;
            unsigned char status;
            } usage_profile_array[500] = {0};

static bit  print_OK_1=0,print_OK_2=0,just_started=1,first_run=1,event_detected=0;  //16 bits
available
float app_totals[8]; //app1,app2,app3,app4
float app_flowrates[8] = {10.0,16.0,48.0,0,0,0,0,0}; //app1,app2,app3,app4
unsigned long loop_count = 0;
extern char print_error[MAX_ERROR][40];
float alpha = 0.1; //exponential averaging factor
int ref_num = 0;

unsigned int error_code = 0,index = 0, row_num = 0, ISR_skip_count =0, event_no=0;
volatile unsigned int ov_ctr_short,rev_count,pulse_count,master_count,ov_ctr_3;  //counters
volatile unsigned long ov_ctr_long;
volatile unsigned int start_time,time_out_short,time_out_long,timer_3_cnt;       //time storage
volatile float ratio,short_pulse_width,long_pulse_width, flowrate;
float exp_averaged_flow, cumsum,running_total;
unsigned long meter_rdg, ave_count;        // consumption in LPM
float flowprev;  float temp;
unsigned char h,m,s;   // hour, min, sec
unsigned char d,mon,y; // day month and year          //For debugging rtc_init
unsigned long initial_unix_time = 1311297000;
unsigned long initial_time[2];         //initial unix time and alarm0 setting for rtc init.
unsigned char *app_status;                     //first 4 bits indicate status

//Funtion Prototypes Declarations
float get_pulse_width(enum pulse_type);
float lookup(float);
float calculate_flow_rate(float,int);
float get_EMA(float,float,float);
float get_SMA(float,float,int);
void display_led(char *);
int is_flow_stable(void);
//void event_detection(void);
//void update_meter_rdg(char *);
//int isinrange(float, Apps_ON);

//*****************************************************************************
//DATA PROCESSING FUNCTIONS
//*****************************************************************************
```

```
//Function returns the time elapsed b/n 2 pulses based on get_type(long-s/short-ms)
    float get_pulse_width(enum pulse_type get_type){
       unsigned char sreg;
       unsigned long utime_diff;
       unsigned int time_in;
       float time_diff;
        sreg = SREG;                        // Save global interrupt flag
        #asm("cli");                        // Disable interrupts
        time_in =  TCNT1L + (TCNT1H << 8);  //16-bit register access
        switch(get_type) {
           case SHORT_PULSE:
                utime_diff = (unsigned long)time_in
                            +((unsigned long)ov_ctr_short*65536)
                             -(unsigned long)time_out_short;
                ov_ctr_short=0;
                time_out_short = time_in;
                time_diff = utime_diff/(1000.*CPU_FREQ_MHZ);    //in milli-seconds
            break;
           case LONG_PULSE:
                 utime_diff = (unsigned long)time_in
                              +((unsigned long)ov_ctr_long*65536)
                              -(unsigned long)time_out_long;
                ov_ctr_long=0;
                time_out_long = time_in;
                time_diff = utime_diff/(1000000.*CPU_FREQ_MHZ);    //in seconds
            break;
            }
        SREG = sreg;                  // Restore global interrupt flag
        return time_diff;
    }

/*This function takes the raw times and returns the scaled time ratios and sets
 the global variable row num to indicate the scaling factor to be used*/
     float lookup(float raw_times)  {
            index = pulse_count % ENC_PPR;
            if (raw_times < (lu_tbl[0][index+1] + lu_tbl[1][index+1])/2 ) {
            ref_num = 0;//use the high ref table
            return  (*(*(lu_tbl+0)+(index+1)))/(raw_times);
            }
            else if (raw_times <= (lu_tbl[1][index+1] + lu_tbl[2][index+1])/2 &&
            raw_times >= (lu_tbl[0][index+1] + lu_tbl[1][index+1])/2) {
            ref_num = 1;
            return  (*(*(lu_tbl+ 1)+(index+1)))/(raw_times);
            }
            else  {
            ref_num = 2;
             return  (*(*(lu_tbl+2)+(index+1)))/(raw_times);
            }
     }

//Get flowrate from table using the scaled times
    float calculate_flow_rate(float norm_pulse_widths, int ref_num)  {
                flowrate = (norm_pulse_widths * (*(*(lu_tbl+ref_num) + 0)));
                return  flowrate;
    }

//This function calculates the exponential moving average
    float get_EMA(float x_n, float s_n, float a){
         float average;
         if (s_n == 0) average = x_n;
         average = a*x_n + (1-a)*s_n;
         return average;
    }

//This function calculates the cumulative sum
    float cum_SUM(float x_n, float s_n){
            s_n = x_n + s_n;
            return s_n;
    }
//This routine checks if the flow is stable in regards to the average
    int is_flow_stable(void) {
```

```
            return (fabs(flowrate-exp_averaged_flow) <= MAX_RES);
    }
//This routine adjusts the averaging window based on flow stabililty
    void adjust_averaging_period (void){
        if (is_flow_stable() && (alpha > .10))
             alpha -= 0.1 ;
        else if (!is_flow_stable() && (alpha < .50))
             alpha += 0.1;
    }
//This routine reports the final output based on stability of the flow and
//adjusts the reported flow to reflect any changes instantenous flow
    float report_flow_rate(void){
         if (is_flow_stable())
              {
                return(exp_averaged_flow);
              }
          else if (!is_flow_stable())
              {
                return(flowrate);
                exp_averaged_flow = 0;
              }
    }
//Update meter reading and individual appliances based on collected data
  void update_meter_rdg(char *app_status) {
    int i;
     for (i = 0;app_status;i++) {       //number of appliances
        if (*app_status & 1)
            app_totals[i] = app_flowrates[i]*running_total*60;
            *app_status>>= 1;
        }
     meter_rdg =  report_flow_rate()*running_total*60;
}

//Event Detection
    void event_detection(void){
         if (is_flow_stable())
                event_detected = 0;
        else if (!is_flow_stable()){
                event_detected = 1;
                event_no++;
                usage_profile_array[event_no].time_stamp = get_utime();
                usage_profile_array[event_no].f_rate = exp_averaged_flow;
                usage_profile_array[event_no].status = *app_status;
        }
    }
//This routine checks if the reported flowate is within the appliances deteremined
    int isinrange(float flowrate, struct Apps_ON app) {
            return (flowrate >= app.avg_frate - app.stdev
                    &&  flowrate <= app.avg_frate + app.stdev);
    }
//updates the appliance status based on flowrate and which appliances are on
    void update_status(float flowrate) {
        if (flowrate == 0.0)    //all appliances are off
            app_status=0x00;                    //0b---0000
            else if (isinrange(flowrate,app_1))
            *app_status=0x01;                    //0b---0001
            else if (isinrange(flowrate,app_2))
            *app_status=0x02;                    //0b---0010
            else if (isinrange(flowrate,app_3))
            *app_status=0x04;                    //0b---0100
            else if (isinrange(flowrate,app_12))
            *app_status=0x03;                    //0b---0011
            else if (isinrange(flowrate,app_13))
            *app_status=0x05;                    //0b---0101
            else if (isinrange(flowrate,app_23))
            *app_status=0x06;                    //0b---0110
            else if (isinrange(flowrate,app_123))
            *app_status=0x07;                    //0b---0111
    }
//update the LED to display the most recent appliance status
    void display_led(char *app_status){
```

```
             LED_STATE = ~(*app_status);
    }

//*********************************************************************
//END OF DATA PROCESSING FUNCTIONS
//*********************************************************************



//*********************************************************************
//INTERRUPT HANDLING
//*********************************************************************

/*External Interrupt 0 service routine (CH_GZ at PD.0),rising edge*/
    interrupt [INT0] void ext_int0_isr(void)
    {
            if (just_started == 1) { /*enable INTs 1 & start reading*/
                #asm("cli");
                EICRA=0x0F; //0xFF
                EICRB=0x00; //0x03
                EIMSK=0x03; //0x1F
                EIFR=0x03;  //0x1F
                TIMSK3=0x01;
                start_time = (unsigned long)(TCNT1L + (TCNT1H << 8))
                            +((unsigned long)ov_ctr_short*65536);
                ov_ctr_short = 0;
                just_started = 0;
                #asm("sei");
            }

            long_pulse_width = get_pulse_width(LONG_PULSE);

            while (first_run==0){
                if (pulse_count != ENC_PPR)
                    error_code = 2;
                if (master_count/pulse_count != rev_count)
                    error_code = 3;
                if (fabs(running_total - long_pulse_width) > 10.0)
                    error_code = 4;  //make error condition laxer
            break;
            }

            if (SET_METER&&(!first_run))collecting = 0;
            pulse_count = 0;
            rev_count++;
            print_OK_1= 1;
            first_run =0;
    }

/*Timer 1 overflow interrupt service routine*/
    interrupt [TIM1_OVF] void timer1_ovf_isr(void){
        ++ov_ctr_short;
        ++ov_ctr_long;
    }

/*External Interrupt 1 service routine (CH_B+ at PD.1)*/
    interrupt [INT1] void ext_int1_isr(void){
        TEST_PIN = 1;
        ISR_skip_count++;
        if (SET_METER && (ISR_skip_count == 4)){
            lu_table_eeprom[row][pulse_count] = get_pulse_width(SHORT_PULSE);
            }
        if ((!SET_METER)&&ISR_skip_count == 4)         //check every 4th pulse
        {
                ISR_skip_count = 0;
                short_pulse_width = get_pulse_width(SHORT_PULSE);
                cumsum = cum_SUM(short_pulse_width,cumsum);
                ratio = lookup(short_pulse_width);
                flowrate =  calculate_flow_rate(ratio,ref_num);
                exp_averaged_flow = get_EMA(flowrate,exp_averaged_flow,alpha);
                adjust_averaging_period();
```

```
                running_total = running_total + (short_pulse_width/1000);
                pulse_count++;
                master_count++;
                ave_count++;
                print_OK_2= 1;
                TEST_PIN = 0;

            if (TEACH_APPS)
             {
                  temp = roundf_10(exp_averaged_flow*2);
                  j = (int)temp;
                  data_list[j]++;
                      if (event_detected)
                      {
                      usage_profile_array[event_no].time_stamp = get_utime();
                      usage_profile_array[event_no].f_rate = exp_averaged_flow;
                     // save the time and the flowrate at that time
                      }
             }
        }
    }


//*********************************************************************
//MAIN FUNCTION
//*********************************************************************

    void main()
    {
        #include <avrconfig.h>
        printf("\rInitializing...\r ");
        restore_meter_data();  //restores last reading,setup status,etc.
        initial_time[0] = 1311236002;initial_time[1] = 0;
        rtc_init(initial_time);
        LED_STATE = 0x00;

        if (!INITIAL_METER_SETUP)
        {
            debug_print("Meter Reference Information Missing\n");
            debug_print("Now Running Initial Setup Mode..\r");
            get_meter_reference_data();
            update_lookup_table();
        }
        else


         if (!APPLIANCE_SETUP)
         {
              printf("No Appliances Reference Table.. PRESS 1 to teach.. \r ");
              //learn_appliances();
          }
          else

        printf("Meter Reading Mode..\r");
        printf("SETUP OK\r..waiting for zero pulse\r ");


        while (1)
        {
            #asm("wdr");
            //update_status(report_flow_rate());
            //display_led(app_status);
            get_time(&h,&m,&s);
            get_date(&d,&mon,&y);
            //display_led(report_flow_rate());

            if (error_code != 0)      //handle any errors
            {
                debug_print("ERROR %i: %s",error_code,
                (print_error[error_code-1]));
                 backup_meter_data();
```

```
            log_error(error_code);
        }

        //        Debug printing
        //        if (print_OK_2==1) {
        //         debug_print("%02d:%02d:%02d  %4u,%8.3f,%7.3f,%7.3f,%7.3f, %i\r",
        //        h,m,s,pulse_count,short_pulse_width,exp_averaged_flow,
        //        average_flow,report_flow_rate(),is_flow_stable());
        //        print_OK_2= 0;
        //        }
        //

        //            TEST PRINTING
        if (print_OK_2==1)
        {
        //  debug_print("%i, %7.3f\r", pulse_count,short_pulse_width);
        debug_print("%4u,%8.3f,%7.3f,%7.3f,%7.3f, %i, %i\r",
        pulse_count,short_pulse_width,flowrate,exp_averaged_flow,
        report_flow_rate(), ref_num, is_flow_stable());
        print_OK_2= 0;
        }

        if (print_OK_1==1)
        {
        debug_print("\n REV%i, %7.3f,%7.3f %02d/%02d/%02d %02d:%02d:%02d\r",
        rev_count,running_total,long_pulse_width,mon,d,y,h,m,s);
        running_total = 0;
        print_OK_1= 0;
        }
    }
}
```

Appendix B

```
%-------------------------------------------------------------------------
% Load Data into Matlab from csv files for processing
% Input - CSV Files that contain the compiled usage data for a
% residential home
% Output - Consumption profile for a 24 hour period (individual
% appliance and running total) plots and .mat format
%-------------------------------------------------------------------------
clear all;clc;
% Number of appliances
N = 4;
% Appliance Identifiers
sym('furnace');
sym('dryer');
sym('waterheater');
sym('stove');

%Load Consumption Profile from CSV files
furnace1 = csvread('furnace.csv');
waterheater = csvread('waterheater.csv');
dryer = csvread('dryer.csv');
stove = csvread('stove.csv');
loss_factor = 0.99;
%Combine all data into one (1x86400) array
usage_CFH = loss_factor*(dryer(:,2) + waterheater(:,2) + furnace1(:,2) + ...
            stove(:,2)+ stove(:,2));
%Convert data into Liters/min from Cu.Ft/Hr
usage_LPM = usage_CFH*0.472;
%Calculate total daily consumption
total_usage_LPM = cumsum(usage_LPM)/3600;
%save files a .mat files to use with Simulink
save usage_LPM.mat;
save total_usage_LPM.mat
%Consumption profile
subplot(2,1,1);plot(usage_LPM);
axis([0 86400 0 120 ])
set(gca,'YTick',0:15:120)
set(gca,'XTick',0:3600:86400)
set(gca,'XTickLabel',{'0','1','2','3','4','5','6','7',...
    '8','9','10','11','12','13','14','15','16','17',...
    '18','19','20','21','22','23','24'})
title('Typical Dialy Gas Usage')
xlabel('Time (hr.)')
ylabel('Consumption Rate, L/min')
subplot(2,1,2);plot(total_usage_LPM);
axis([0 86400 0 800])
set(gca,'YTick',0:150:800)
set(gca,'XTick',0:3600:86400)
set(gca,'XTickLabel',{'0','1','2','3','4','5','6','7',...
    '8','9','10','11','12','13','14','15','16','17',...
    '18','19','20','21','22','23','24'})
title('Total Consumption')
xlabel('Time (hr.)')
ylabel('Total, L')
%-------------------------------------------------------------------------
```

```
%-------------------------------------------------------------------------
%Pulse Generation Code for Simulating Meter Output for input loads
clear all;
%Load the Input Data
load usage_LPM;
load total_usage_LPM;
usage_LPM_REF = 49;

%Internal Ratio of the Meter
meter_ratio = 18;
for k = 2 %generate at (k*18) PPR
    PPR = k*meter_ratio;
    %METER MODEL BASED ON 40 PPIR
    s1 = ['load v' int2str(PPR/meter_ratio) '.mat']; eval(s1)
    s2 = ['xmean = v' int2str(PPR/meter_ratio) '(:,1);']; eval(s2)
    s3 = ['xmin = v' int2str(PPR/meter_ratio) '(:,2);']; eval(s3)
    s4 = ['xmax = v' int2str(PPR/meter_ratio) '(:,3);']; eval(s4)
    r = zeros(k,1);

    dt_NO_FLOW = 0;
    dt = 1; %in seconds
    dp = 0;
    pulse_count = 1; %pulse count
    index = 0;
    temp = mean(total_usage_LPM);   %Estimiate of array size to be
    %preallocated for storing pulses.
    z = ceil(temp*PPR/2);
    pulses = zeros(z,1);

    tic
    for t = 1:86400    %For each second of the flow rate values
        if index==0 %Add the random component to model
            r  = random('Uniform',xmin, xmax);
        end
        if usage_LPM(t) == 0
            dt_NO_FLOW = dt_NO_FLOW + 1;
            continue
        else
            p = dp;
            while (p < dt)
                index = mod(pulse_count,PPIR);
                if (dt_NO_FLOW~=0)
                    pulses(pulse_count) =  dt_NO_FLOW*1000;
                    dt_NO_FLOW = 0;
                else
                    pulses(pulse_count) = r(index+1)...
                        *usage_LPM_REF/usage_LPM(t);
                    p = pulses(pulse_count)/1000 + p;
                end
                pulse_count =  pulse_count + 1;
            end
        end
        dp = p-dt; %store the fraction of times that
        %added for timing procedure
    end
    toc
    %Clean up operations
    pulses = pulses(pulses~=0);
    s5 = ['pulses_' int2str(PPR/meter_ratio) ' = pulses;']; eval(s5);
    s6 = ['save pulses_' int2str(PPR/meter_ratio) '.mat;']; eval(s6);
    %save pulses.mat;
    err = length(total_usage_LPM) - sum(pulses)/1000;
    %Remove long times from pulses
    pulses_clean = pulses(pulses<=20000);
    %% Plots
    figure(PPR)
    plot(pulses);
    title('Generated Pulse Stream from Meter Model')
    xlabel('Pulse Number')
    ylabel('Time Elapsed between Pulses, mS')
    figure(2)
```

```
    plot(pulses_clean);
    title('Generated Pulse Stream from Meter Model')
    xlabel('Pulse Number')
    ylabel('Time Elapsed between Pulses, mS')
end
%-------------------------------------------------------------------------
```