2019

Development of a Sensing System for Underground Optic Fiber Cable Conduit Mapping

Sherif Bakr
Minnesota State University, Mankato

Development of a Sensing System for Underground Optic Fiber Cable Conduit Mapping

Minnesota State University, Mankato

Department of Mechanical & Civil Engineering

This thesis is submitted in partial fulfillment of the requirements for the Degree of

Master of Science

in

Mechanical Engineering

Submitted by:

Sherif Bakr

Thesis adviser:

Dr. Min Li

Committee Members:

Dr. Min Li (Chair)

Dr. Patrick Tebbe

Dr. Qun Zhang

07/19/2019

WE, THE UNDERSIGNED MEMBERS OF THE COMMITTEE,

HAVE APPROVED THIS THESIS

Development of a Sensing System for Underground Optic Fiber Cable Conduit Mapping

BY

SHERIF BAKR

07/19/2019

COMMITTEE MEMBERS

Min Li, Ph.D. (Chair)

Patrick Tebbe, Ph.D.

Qun Zhang, Ph.D.

ACCEPTED AND APPROVED ON BEHALF OF THE UNIVERSITY

Minnesota State University, Mankato

July 2019

Acknowledgments:

This research was supported by Condux International, Inc. where I have worked as a Design Engineer intern for about 2 years. I would like to thank Mike Brooks, the former Engineering Manager at Condux, for giving me the opportunity to join such an eminent firm and be a part of the Design Engineering team.

I would also like to thank Dr. Min Li for his continuous assistance and support throughout the past 11 months. Dr. Li has provided his perception and expertise that critically assisted this research. I also thank my committee members, Dr. Patrick Tebbe and Dr. Qun Zhang, for their advice and research tips.

I dedicate this thesis to my late grandma, Aziza, who I have spent my childhood and teenage years with. She will always be on my mind and forever in my heart.

I would like to show my boundless appreciation to my mum and dad, Azza and Salah, for their continuous guidance and encouragement ever since I came to the United States. I would finally like to thank my siblings and my fiancé, Mido, May, and Aya, for their endless love and support.

Abstract

The motivation of this research is to obtain an accurate three-dimensional (3D) layout of an underground conduit, which may be beneficial to optic fiber cable installers and engineers. A newly designed algorithm for 3D position tracking with the help of an inertial sensor and an encoder has been developed. Two types of representations (Euler angle and Quaternion) for orientation and rotation are also introduced, followed by several data pre-processing procedures. A sensing fusion method is utilized to overcome the accumulated errors introduced by the sensor drifting. Considering the application of 3D underground duct mapping in this research, a sensing system using the newly designed algorithm was designed and analyzed. Additional information, such as the orientation and position of the starting and ending points, are integrated into the algorithm to correct the sensing drifting and refine the position estimation. To verify and demonstrate the design of the algorithm and sensing system for 3D underground duct mapping, an experimental test-bed based on the sensing system design, which consists of an IMU, a duct rodder and a fiber blower, was developed. Experiments on three different layouts of the conduit were conducted and analyzed to demonstrate the feasibility and efficiency of the newly developed algorithm and the sensing system design.

Table of Contents

1. Introduction

1.1 Motivation

As the release of the 5G technology approaches, the market for optic fiber cables and their associated installing machines and accessories is expected to thrive. Fiber cables' capabilities are continuously increasing. They will have the ability to carry more data or information in a shorter period of time and therefore, they are expected to be more expensive. Those cables are subject to damage if they are blown through a conduit of a high bend radius. Avoiding the damage of an optic fiber cable, which costs thousands of US dollars, is one of the main goals of this research project. There are millions of ducts or conduits that have been laid underground for decades. Those ducts have experienced drifts, bends, or inconsistent expansions based on the nature of their surroundings. For example, cold temperatures may cause a duct to highly bend. Furthermore, poor soil exploration of the location where those ducts are installed is a reason for optic fiber cable damage over time. This research area is challenging due to the uncertain underground conditions and the complexity of the duct layouts. The concern is that the precise depth and orientation of those ducts are unknown and inconsistent over years. This means that if duct installers want to change or to fix a piece of duct underground, they could dig meters far from where those ducts are located, which will result in a waste of time and money. In addition, over years, when roads are paved, the depth of the underground ducts relative to the surface of the road changes. It is very difficult to keep track of the ducts' depth relative to the surface of the road. Shown below in [1, Fig. 1] is the process of blowing optical fiber underground. The optical fiber is rolled around a spool, shown on the left side of the figure. The tip of the

fiber is then fed into the machine from one side. It is then pushed by the use of two rubber belts, or chains, driven by a motor or two (pneumatic or hydraulic). The air compressor also helps in pushing the cable through the duct. The duct has to be pressurized before the cable is fed into the machine to ensure that there are no any leaks or dust in the duct [1].

Manufacturers of fiber optic installation machines spend a lot of money and time to develop a machine that is capable of installing the fiber cable from a designated start to an end location. It would be beneficial if they were able to view, through a computer software, how the underground duct is laid out. The information, such as the location and the magnitude of the maximum bend radii, the maximum the elevation or depth, could be very advantageous to the engineers who design the optic fiber cable installation equipment. In addition, it will save time to the machine operators who will be able to know, prior to blowing the fiber cable, whether or not their machine will be capable of performing the task. It will also save large amounts of money by protecting the optical fiber cables from potential damage. The goal of this project is to develop an intelligent sensing system that is capable of reconstructing a precise 3D plot of a layout of an underground duct.

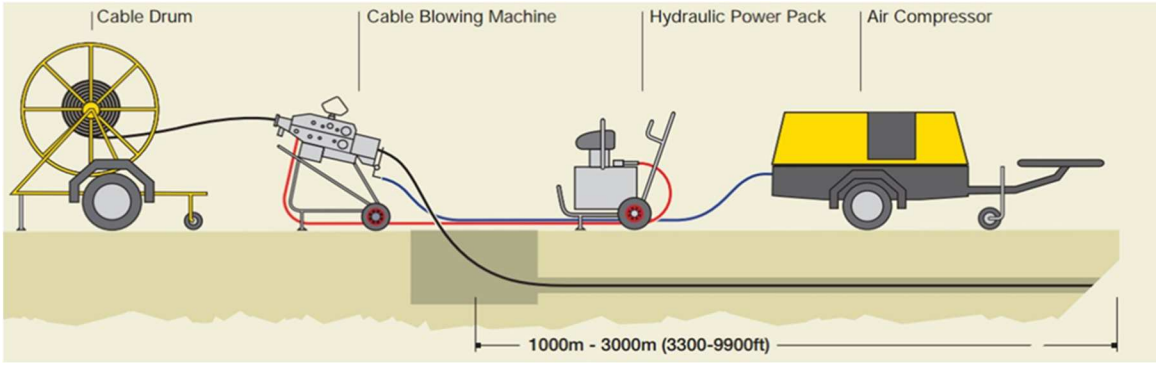


Figure 1. Optical fiber cable installation in an underground duct [1]

1.2 Introduction to Condux International, Inc.

Condux International has been manufacturing underground and overhead cable installation tools and equipment for the telecoms and electric industries. Condux has been in the optic fiber cable installation field for over 30 years with two main production lines: cable blowers and cable pullers. Condux also manufacturers all the tools required by contractors to perform a complete installation process. Due to the uncertain underground conditions, mentioned in the previous section, Condux has directed its focus towards duct mapping. It is very crucial for the engineers at Condux to determine the layout or the shape of the underground conduit before blowing the optic fiber cable in it. This layout or map of the underground conduit would allow the engineers to determine whether or not the fiber installing machine is capable of blowing the fiber through the conduit. In addition, the USA government requires the cables installers to provide a map that shows where those underground conduit are located underground prior to installing the fiber. This research assists Condux, and any other company in the same field, to obtain a 3D map of an underground optic fiber cable conduit.

1.3 Introduction to Inertial Sensors

Inertial sensors are sensors based on inertia and relevant measuring principles, which mainly includes accelerometers and gyroscopes, denoted as Inertial Measuring Units (IMUs). Their most familiar applications include smart phones, tablets, and Virtual Reality (VR) headsets. For instance, accelerometers are used in smart devices to ensure that images on screens are always displayed upright. They are also widely used in military, robotics, and aerospace-related applications. When IMUs are combined with other sensors, such as

the Global Positioning System (GPS) receivers, more complex outcomes could be obtained. The term that describes the combination of accelerometers with other sensors is called “sensor fusion”. Many researches and experiments have been done on sensors’ fusion due to the very useful outcomes that could be obtained by combining different kinds of sensors together besides their cheap cost. The most common and useful type of inertial sensor is the Micro electro-mechanical system (MEMS) sensor. The usage of the MEMS sensors in the medical and military sectors has widely increased during the past decade.

1.3.1 Accelerometers

Accelerometers could be used for three purposes; the first of which is orientation detection, as mentioned earlier, the second is velocity and displacement determination, and the third is vibration detection in a mechanical system. Accelerometers convert a mechanical property, which is the force acting on the sensor, to a corresponding acceleration value, usually expressed in the units of gravitational acceleration, “*g*”. The use of accelerometers to determine displacement, is usually undesired due to the numerous types of errors that may result, which will be further elaborated in the next sections. It is very challenging to completely eliminate the errors associated with accelerometers due to their very random and unpredictable nature.

There are different types of accelerometers. The accelerometers are chosen based on the required functions or the type of measurement to be made. There are three technologies that are used to measure acceleration. The first of which is Piezoelectric (PE) accelerometers. Piezoelectric accelerometers offer a wide measurement frequency range (up to 30 kHz) and are sold in a wide range of sensitivities, weights, sizes, and forms. PE accelerometers are suitable for shock or vibration measurements. Piezoelectric accelerometers have a piezoelectric element mounted by a screw between a dummy weight and the base. Measurements of large shock or vibration are possible due to the accelerometer's high mechanical capabilities (ultimate strength). Shown below in [2, Fig. 2] is a schematic of a piezoelectric accelerometer. [2]

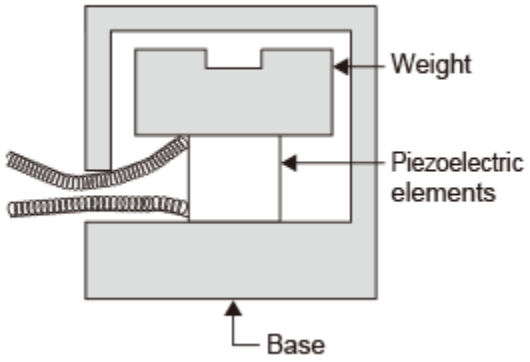


Figure 2. Piezoelectric accelerometer [2]

The second type of accelerometers is Piezoresistive (PR) accelerometers. PR accelerometers usually have low sensitivity which limits their usage for measuring vibrations. They are commonly used, however, to measure shock. PR accelerometers usually have a wide bandwidth (higher than 130 kHz) and their lowest frequency response could go down to 0 Hz. This property allows PR accelerometers to measure transient states. Shown below in [3, Fig. 3] is a schematic of a Piezoresistive accelerometer. [3]

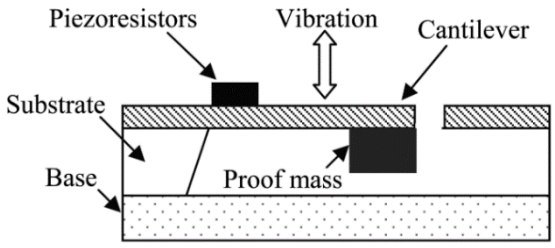


Figure 3. Piezoresistive accelerometer using cantilever design [3]

The third type of accelerometers is the Variable capacitance (VC) accelerometer. This is the newest and most common type of accelerometer. Its high sensitivity and narrow bandwidth help in measuring low-frequency vibration, motion, and steady-state acceleration. Thermal zero and sensitivity shifts can be as low as 1.5% over a temperature range of 180 °C. VC accelerometers rely on the change in capacitance in response to acceleration. Shown in [4, Fig. 4] is a schematic of a variable capacitance accelerometer, or a capacitive accelerometer. [4]

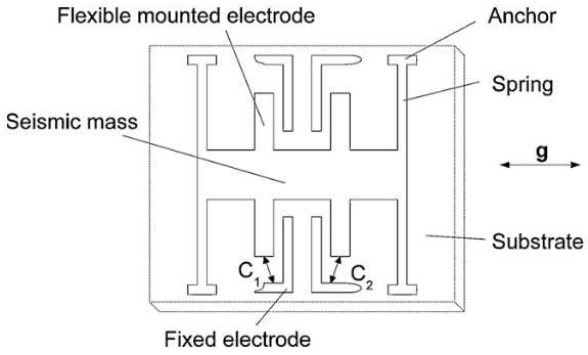


Figure 4. Capacitive accelerometer [4]

All three types of accelerometers mentioned above are used for various applications. Each has its advantages and disadvantages. They are chosen based on the type of measurement required. The frequency response of the accelerometer that corresponds to

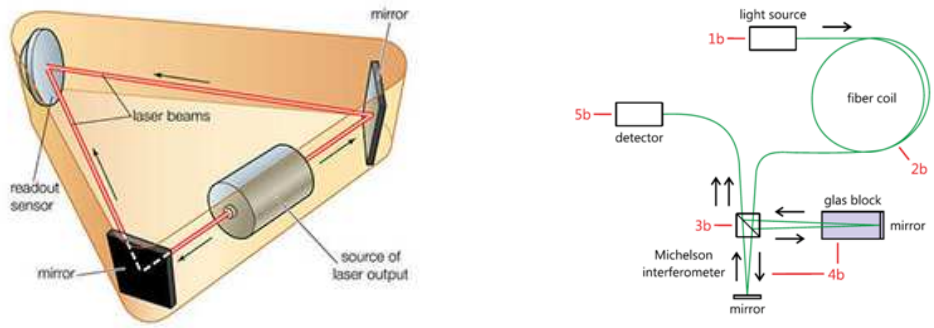
the mechanical stimulation is considered the most critical parameter an accelerometer is chosen upon. [5]

MEMS accelerometers could be utilized in this research to obtain velocity and displacement by direct integration. This method results in a large random accumulative errors, however. The accelerometer could also be used to estimate the roll and pitch rotation angles. This will be further elaborated in the next section.

1.3.2 Gyroscopes

Gyroscopes, in their simplest form, are a spinning wheel or a disk that are allowed to rotate freely about an axis. The orientation of the axis is not affected by tilting of the mounting. Gyroscopes are the most important part of an inertial navigation system. The use of gyroscopes has significantly increased over the past decade. Their use has evolved to monitoring the orientation of an aircraft or guiding an unmanned aircraft during flight. In addition to numerous medical, military, and aerospace applications. Gyroscopes measure angular velocity, usually in degrees/second. The most common types of gyroscopes are the Ring Laser Gyroscope (RLG) and Fiber Optics Gyroscope (FOG). Both operate based on the same principle, the Sagnac effect. The Sagnac effect results from splitting a light beam in two and directing them through different paths. Those two light beams are then allowed to interfere. The fringe pattern is observed and the change in rotational angle is calculated based on the fringe data. The difference between FOG and RLG is that one propagates in a cavity and the other propagates in the fiber, see Figure 5. [6]

The third and most important type of gyroscopes is the MEMS gyroscope. The use of MEMS gyroscopes has exponentially increased over the past five decades. The main advantage of MEMS gyroscopes over FOG and RLG gyroscopes is their low cost. In addition, MEMS gyroscopes are very small in size, which allows their usage in various applications where FOG and RLG cannot be used. Also, they do not have any moving components, which means that they are less likely to be damaged. They are also completely maintenance free. Shown below in [7, Fig. 5] are the ring laser and fiber optic gyros. [7]



a) Ring laser gyro *b) Fiber optic gyro*

Figure 5. Ring laser and Fiber optic gyroscopes [7]

1.3.3 Utilizing Accelerometers and Gyroscopes to Obtain Velocity, Displacement, and Orientation

Since the IMU outputs the time intervals along with the accelerometer and gyroscope values, direct integration could be done to obtain velocity data from the acceleration data along each of the three axes. Another integral could also be done to obtain the displacement data from the velocity data. The same theory applies to the gyroscope. Integration of the angular velocity data could be done to obtain the orientation or the Euler angles roll, pitch, and yaw. It is worth mentioning that direct integration of the data results in high random

accumulative error over time. This accumulative error drastically decreases the quality of the data and outputs false results.

A trapezoidal numerical integration method is used to estimate velocity and displacement with acceleration data. The mathematical equations are shown below.

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_0^t \mathbf{a}(t)\,dt \tag{1}$$

$$\mathbf{p}(t) = \mathbf{p}_0 + \int_0^t \mathbf{v}(t)dt \tag{2}$$

Where $\mathbf{v}(t)$ and $\mathbf{p}(t)$ are the estimated velocity and displacement of the sensor, $\mathbf{v}_0$ and $\mathbf{p}_0$ are the initial velocity and position, respectively.

Note that the same equations, Eqs. (1) and (2), could be used to determine the Euler angles from the angular velocities (or the gyroscope data).

The accelerometer's data could also be used to determine the roll and pitch angles using the following relationships shown in Eqs. (3a) and (3b).

$$Roll = \tan^{-1}\left(\left(a_{nb}^b\right)_y \Big/ \left(a_{nb}^b\right)_z\right) \tag{3a}$$

$$Pitch = \tan^{-1}\left(\left(a_{nb}^b\right)_x \Big/ \sqrt{\left(a_{nb}^b\right)_y^2 + \left(a_{nb}^b\right)_z^2}\right) \tag{3b}$$

Roll angle is calculated from the y and z components of the accelerometer's data whereas pitch angle is calculated from the y, z, and x components of the acceleration.

1.4 Review of Prior and Related Work

This research addresses a problem that has been discussed by many researchers and engineers over the past decades. Remote tracking of the location and the orientation with

alternative approaches has been the concern. Researches on precise real-time tracking of acceleration, velocity, and displacement are widely growing. Tracking, locating, and mapping are commonly used in applications of the fields of military, engineering, robotics, and automotive industries. Sometimes the sole use of GPS in determining the location is not accurate and reliable and hence the Inertial Measuring Units (IMUs) are also utilized. Researches on IMUs have been significantly growing in the past decade due to their relatively cheap price and great advantages. They are also usually fused with other sensors for accuracy purposes. An example of a commonly used inertial sensor is the Xbox 360 Kinect Sensor, which is capable of determining the orientation of a human body. Although the principle of orientation and location estimation is very straightforward, to obtain accurate and reliable estimation is still challenging.

Numerous filters and data rectification methods are developed to improve the estimation. The Kalman filter, extended Kalman filter, Madgwick filter, and the complementary filter are commonly used for data refinement, each of which has its own advantages and disadvantages. The output from each filter was compared against the other using the same data set and algorithm. It is very challenging to state that one filter is better than the other two. The filter type is chosen based on the function and the design intent of the application. For example, some applications require immediate analysis and processing of data. On the other hand, offline data analysis and processing may be acceptable in other applications, and hence the processing time will not be a constraint.

1.4.1 Orientation and Position Estimation Methods

Various researches have been made to track the human body movement. Orientation estimation researches were conducted using magnetic, angular rate, and gravity (MARG) sensors. With the help of accelerometers, the position, relative to a starting point in the XYZ-plane, was computed [8]. The nature of the human body movement was utilized in the developed algorithm. Calibration of the accelerometers, gyroscopes, and magnetometers was performed during the stance phase of the human movement. Although it may seem that very accurate results could be obtained, there was still an error of about 3-5%, as the sensing system completely stops every step the human takes [8]. In another research, a self-calibration IMU was utilized to model the errors of the accelerometer, gyroscope, and magnetometer simultaneously. A post-processed least-squares framework was used to reduce the accumulative errors. However, the errors propagated over longer time periods. This limits the versatility of the research [9]. Additional researches were performed to map tendon ducts on pre-stressed concrete bridges by means of Ground Penetrating Radar (GPR) [10]. The use of a radar-based solution would be very expensive compared to the use of an IMU-based system, as the inertial measuring units are relatively cheap. The GPR technology was only precise in the XY-plane; however, the results were not accurate enough on the Z-axis, the axis perpendicular to the surface of the ground [10].

A mobile battery-powered robot was designed for air duct exploration in recent research. The purpose of the research was to 3D map air ventilation ducts by the use of a wheeled robot. The relatively big size of the air ventilation ducts worked as an advantage to the researchers. The research was focused mainly on less complicated environments. The 3D

mapping was only applicable for ducts that had minimal to no obstacles. However, a high growth rate of errors occurred, which limited the robot's capability to function on short ducts only. The high errors resulted from the unavailability of a global localization reference [11]. It is possible to obtain a bias-free displacement estimation only if the motion is sinusoidal. Weighted-Frequency Fourier Linear Combiner (WFLC) and Band-Limited Multiple Fourier Linear Combiner (BMFLC) methods were used to find the displacement only from accelerometer data. However, those methods could only be used if the motion is periodic. In other words, the WFLC and the BMFLC could be utilized only if the acceleration can be modeled by a series of sine and/or cosine components. This method could be useful in other applications [12]. A drift-free displacement estimation method using a periodic or a quasi-periodic signal was proposed in a research paper. The paper utilized the WFLC and the BMFLC methods mentioned above. However, the only difference is that it could handle quasi-periodic signals, unlike the method proposed in [12]. The position of an actual periodic or a quasi-periodic motion is estimated through an attenuated and phase-shifted position of periodic or quasi-periodic motion. The results had a low-frequency drift that was filtered out using a high-pass filter [13].

It would have been helpful if the Global Positioning System (GPS) could be used in this project. However, the currently available GPS technology cannot penetrate the ground and reach several meters underneath. The use of a signal transmitting technology underground was not considered for the same reason. Another research was conducted to determine the position and the orientation of a *Tissue Imaging Probe System* (TIPS) with the use of an IMU and a microcontroller. The Finite Impulse Response filtering (FIR), which is a high

pass filter, was used to eliminate the errors resulted from the double integration of the accelerometer data. Increasing the filtering capabilities of the FIR caused a slower response time, which was not desirable in the research. The experimental displacement was off compared to the actual displacement the probe was moved [14]. A 2-D location tracking method was proposed in [15]. The method was proposed to track a land vehicle using a 6-dof IMU. The calibration method proposed in the research did not allow usage over long time periods. The device had to be mounted on top of the vehicle with the z-axis orthogonal to the roof of the car. That was the only method to compensate for the acceleration due to gravity and prevent the interference between the gravitational acceleration and the vehicle's translational acceleration.

1.4.2 Kalman Filter for Orientation and Location Estimation

The Kalman filter is a very complicated set of recursive mathematical equations used to provide estimates of a state of a process. It functions in a way that minimizes the mean of squared errors. It can be used to make estimations about future states even if the past and the present states are imprecise. The Kalman filter uses a set of measurements that are observed over a defined period of time. The more data points per unit time, the better the results.

A research was conducted to track a 3-dimensional orientation of a chip that consists of an accelerometer and a magnetometer. The algorithm utilized The Kalman filter to obtain an orientation estimation with respect to the gravity's coordinate system and the chip's local magnetic field vectors. Since accelerometers measure acceleration and not orientation, other input parameters were used in the Kalman filter to correct the accelerometer's errors.

Those additional filters were dynamically varied depending on the state of the system. The filter input parameters were calibrated using MATLAB simulations. A rotation and a vibration movement were performed to adjust the parameters. The chip was used in a human posture tracking device [16]. In another research [17], various inertial sensors were fused together to obtain accurate position and orientation estimation. The research focused on the signal processing aspects of position and orientation estimation. Different modeling methods were used to test the proposed algorithm. The algorithm included the use of the Extended Kalman Filter as well as the Complementary Filter. Several probabilistic models were done for the gyroscope and the accelerometer. The obtained data of the orientations were smoothed using an optimization method; Gauss-Newton. Kalman filter's results were compared against the Complementary filter. Two calibration methods were utilized in the model; Maximum a Posteriori, and Maximum likelihood. Both of those methods estimate a variable from a probabilistic distribution model. The only difference between both methods is that the Maximum a Postetiori method works on a posterior distribution, and not only repeatability.

Another research was conducted to build a map and compute its location using a mobile robot. The mobile robot was only implemented on a 2-dimensional layout. A 3-dimensional test of the system involved added complexity and increased sensing and complexed modeling. The algorithm used involved the estimation of a joint state that had a robot pose as well as a defined position of an observed landmark. State space matrices were used to define the robot's location relative to the horizontal land. The Extended Kalman filter (EKF) was used to improve the resulted map [18]. The optimization formulation of the

Kalman filtering was analyzed in different research. Initial classic Kalman smoothing as a least squares problem was first analyzed. Extensions of Kalman smoothing into systems with nonlinear processes and models was then performed. The Maximum a Posteriori formulation was performed under linear and Gaussian assumptions. A smooth and non-smooth signal was tested with the proposed method. The results showed higher accuracy in smooth signals. Two novel approaches to Kalman smoothing of sparse systems were also performed [19].

Another research was conducted to study the noise of the inertial navigation sensors; an accelerometer and a gyroscope. Simple average filtering algorithm was performed to the gyroscope data at rest and during an identified rotation. The Kalman filter was then used due to the complexity of the sensors' errors. The Kalman filter was used to tie the accelerometer and the gyroscope together to generate more accurate plots of the orientation. The Kalman filter method was proven to produce better results compared with simple filters [20]. Another research implemented the Kalman filter on accelerometer data for three state estimation of a dynamic system [21]. The dynamic system was assumed to be in constant acceleration. The research implements the Kalman filter on a model of the system. The algorithm obtained in the research could be extended for usage in gyroscopes, and a combination of the two; an accelerometer and a gyroscope. Other researches have been conducted to determine the orientation of a rigid body. Research has been conducted to determine the shape and angle of a rigid body in full 3-dimensional, 360 degrees on each axis. The Kalman filter (KF) was used to get a more precise estimation of a current state, based on historical or statistical data [22].

Another research paper combined both, a train of IMUs and MARG sensor arrays, to get more accurate and efficient results about the orientation of a rigid body [23]. A "novel orientation filter" was designed to reduce the accumulative errors. The novel orientation filter has a quaternion representation allowing the data to be used in an analytically derived gradient-descent algorithm [24]. The novel orientation filter was compared against the Kalman filter and the results were very comparable. Kalman filters (KF) or extended Kalman filters (EKF) are used to correct bias when two sensors or more are fused together, such as a gyroscope and a magnetometer. Two researches were completed to track the human body movement and a small handheld microsurgical instrument in [25] and [26], respectively. Both researches utilized the EK and the EKF filters; however, there was a minimal dependency on the magnetometer data in this research due to the unknown underground conditions. Any steel foundations or metal bodies could drastically affect the magnetometer readings. A research was conducted to examine the disadvantages of the Kalman filter. It was proven, via simulations and experiments that the Kalman filter fails if multiple states satisfy the steady-state measurement. Also, the future estimator is very imprecise when the initial guess of the state is inaccurate [26].

1.5 Problem Description and Objectives

The cost of the optical fiber cable is extremely expensive due to its high capabilities. It would be a waste of time and money if that cable was incorrectly installed or damaged during an installation process. The main reason for damaging the cable is trying to push the cable, via an installation machine, through a high bend radius in the conduit. One possible solution is to make an accurate estimation of how the underground conduit looks

like. Prior knowledge of the conduit layout will allow engineers to calculate the bend radii in the conduit and determine their corresponding locations. Accordingly, decisions about the fiber cable capability of passing through the conduit safely will be made. The ultimate goal of this research is to develop an intelligent sensing system that is capable of obtaining an accurate 3-dimensional layout of an underground conduit with the least possible percentage error. Three objectives are needed to be achieved by this work.

- The *first* objective is to develop a position tracking method based on inertial sensor and encoder measurements. This method should be efficient, robust, accurate, and insensitive to different types of errors introduced by measurements.
- The *second* objective is to design a sensing system for 3D underground conduit mapping, optimize the design based on the previously developed method and make it suitable for the corresponding application.
- The *final* objective is to design and build an experimental test-bed to implement the design of the sensing system for 3D underground conduit mapping and consequently verify and evaluate its performance.

1.6 Outline and Organization of Thesis

The remainder of this dissertation is outlined as follows:

Chapter 2 presents a newly designed algorithm for 3D position tracking with the measurements of the inertial sensor and encoder. Two types of representations (Euler angle and Quaternion) for orientation and rotation are also introduced, which is followed by several data pre-processing procedures. A sensing fusion method is utilized to overcome

the accumulated errors introduced by the sensor drifting. Considering the application of 3D underground duct mapping, a sensing system using the algorithm presented in Chapter 2 is designed and analyzed in Chapter 3. Additional information, such as orientation and position of the starting and ending points, are integrated into the algorithm to correct the sensing drifting and refine the position estimation.

To verify and demonstrate the design of the algorithm and sensing system for 3D underground duct mapping presented in Chapter 2 and 3, several experiments are conducted and analyzed in Chapter 4. Finally, Chapter 5 summarizes the contributions of this research as well as the future works that can refine and extend these studies.

2. Self-constrained Position Tracking Using Inertial Sensor and Encoder

To develop an intelligent sensing system that can obtain an accurate 3-dimensional layout of an underground conduit, a self-constrained position tracking method using the inertial sensor and encoder has been established. Several methods are integrated, which consists of two parts: 1) orientation estimation; 2) position estimation. The orientation provides a critical basis for the position estimation, which results in more attention on orientation estimation. See Figure 6.

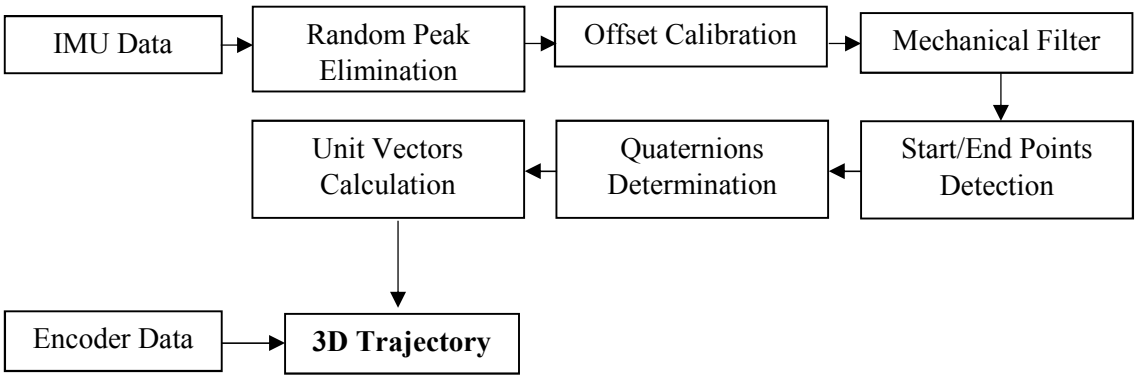


Figure 6. Self-constrained position tracking algorithm

Ideally, the location and orientation information can be obtained by integrating the measurements of inertial sensors (acceleration and angular velocity). However, sensor readings are normally affected by measurement errors. Manipulation of integrals would accumulate the error and dramatically affect estimation results. Thereby, the critical part of position tracking using inertial sensors, is to eliminate the effects of errors on the position estimation. Before introducing the tracking method, the errors of inertial sensors are analyzed, which provide a basis for tracking method design.

$$a_m = a_t + e \text{, where } e = e_b + e_n \tag{1}$$

The sensor measurement a_m is contributed by true value a_t and error e, which consists of slowly time-varying bias e_b and random error e_n. The time-varying bias e_b is due to the system characters and environment such as temperature whereas the random error e_n is caused by the system noise and uncertainty. Normally, the random error has Gaussian distribution with zero mean, which leads to no influence on integration.

In order to conduct position tracking, a number of coordinate frames need to be introduced. To simplify this problem, we assume the earth is stationary and the earth frame is an inertial frame.

- *Sensor frame* (*s*) is the coordinate frame of inertial sensors with its origin located at the center of sensors. All sensor readings are with respect to this frame.
- *Navigation frame* (*n*) is a local coordinate frame that is defined stationary with respect to the earth. The results of position tracking are the estimation of the location and orientation of the sensor frame with respect to the navigation frame.
- *Inertial/earth frame* (*i*) is a stationary frame with the origin located at the center of the earth. The inertial sensors measure the linear acceleration and angular velocity with respect to the inertial frame.

Based on the simplified coordinate frame defined above, there is no relative motion between navigation frame and inertial frame and the inertial sensor can be assumed to measure the linear acceleration and angular velocity with respect to navigation frame.

2.1 Orientation and Rotation Representation

Since the inertial sensor is attached to the moving object, the task of position tracking is to estimation the trajectory of the sensor frame. The inertial sensor measures the inertial parameters of the sensor frame with respect to the inertial frame (or navigation frame) in the sensor frame. A method to represent the orientation and rotation of the moving frame is desired, which can be mainly achieved with two representations, Euler angle and Quaternion.

1) Euler Angle Representation

Any rotation can be accomplished with three rotation about 3 axes of a coordination system. Those three element rotations define Euler angle, which is typically denoted as φ, θ, ψ. There are two types of rotation, extrinsic rotation and intrinsic rotation. The former defines the rotation about the axes of the fixed original coordinate system whereas the later defines the rotation about the moving coordination system attached to the moving body. The Euler angle with extrinsic rotation following the sequence of *z*-*y*-*x* are utilized in this thesis.

A 3×3 rotation matrix is introduced to perform the rotation. All three-element rotation of Euler angle defined above are represented with the following 3 basic rotation matrices:

$$\mathbf{R}_x(\varphi)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\varphi & -\sin\varphi\\ 0 & \sin\varphi & \cos\varphi\end{bmatrix},\mathbf{R}_y(\theta)=\begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{bmatrix},\mathbf{R}_z(\psi)=\begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix} \quad (2a,b)$$

$$\mathbf{R}=\mathbf{R}_z(\psi)\cdot\mathbf{R}_y(\theta)\cdot\mathbf{R}_x(\varphi)$$

The orientation's parameters φ, θ, and ψ are the roll, pitch, and yaw about the *x*-axis, *y*-axis, and *z*-axis respectively. The overall rotation **R** can be represented by a 3×3 matrix constructed with the multiplication of 3 basic rotation matrices. Thereby, to convert the sensor readings to the navigation frame, the following equations are utilized:

$$\boldsymbol{\omega}_{ns}^{n}=\mathbf{R}^{ns}\boldsymbol{\omega}_{ns}^{s}, \text{ where } \mathbf{R}^{ns}=\mathbf{R}_z(\psi)\cdot\mathbf{R}_y(\theta)\cdot\mathbf{R}_x(\varphi) \quad (3a, b)$$

ω_{ns}^{s} are inertial sensor measurements, which are the angular velocity of the sensor frame with respect to the navigation frame in the sensor frame. By multiplying the rotation matrix

R^{ns} from the sensor frame to the navigation frame, the corresponding angular velocity (ω_{ns}^{n}) in the navigation system is derived. The orientation's parameters φ, θ, and ψ are the roll, pitch, and yaw about the *x*-axis, *y*-axis, and *z*-axis respectively, which are estimated with previous measurements of inertial sensors.

2) Quaternion Representation

Besides the Euler angle, any rotation or sequence of rotation of a moving coordinate *B* about a fixed coordinate *A* can also be interpreted as a single rotation by an angle α about a fixed axis (called Euler axis), which is represented with a unit vector $\vec{\mathbf{u}} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^{\mathrm{T}}$. Thereby, a combination of a unit vector representing the axis and a scalar angle can uniquely determine a 3D rotation or the orientation of coordinate *B* relative to coordinate *A*. A number system, quaternions, consisting of four numbers are introduced to mathematically represent this angle-axis rotation or orientation as below:

$$\mathbf{q}^{BA} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} = \begin{bmatrix} \cos\alpha/2 & -u_x \sin\alpha/2 & -u_y \sin\alpha/2 & -u_z \sin\alpha/2 \end{bmatrix} \quad (4a, b)$$

The inverse (denoted by subscript $^{-1}$) or conjugate (denoted by subscript *) of the rotation quaternion are introduced to represent the opposite rotation or swapped relative orientation, which is mathematically expressed in Eq. (5).

$$\left(\mathbf{q}^{BA}\right)^{-1} = \left(\mathbf{q}^{BA}\right)^{*} = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \end{bmatrix} = \mathbf{q}^{AB} \quad (5a, b)$$

Where $\mathbf{q}^{AB}$ represent the orientation of coordinate *A* with respect coordinate *B*.

To represent a sequential orientation and coordinate transformation, the Hamilton product (denoted by $\otimes$) of the quaternion is introduced in Eq. (6). This product is not commutative, which is expressed as $\mathbf{m} \otimes \mathbf{n} \neq \mathbf{n} \otimes \mathbf{m}$.

$$\begin{aligned} \mathbf{m} \otimes \mathbf{n} &= \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \end{bmatrix} \otimes \begin{bmatrix} n_1 & n_2 & n_3 & n_4 \end{bmatrix} \\ &= \begin{bmatrix} m_1 n_1 - m_2 n_2 - m_3 n_3 - m_4 n_4 \\ m_1 n_2 + m_2 n_1 + m_3 n_4 - m_4 n_3 \\ m_1 n_3 - m_2 n_4 + m_3 n_1 + m_4 n_2 \\ m_1 n_4 + m_2 n_3 - m_3 n_2 + m_4 n_1 \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{6}$$

Assume another coordinate C is introduced and its orientation $\mathbf{q}^{CB}$ with respect to coordinate B is given. The orientation of C relative to A is represented with the quaternion product in Eq. (7).

$$\mathbf{q}^{CA} = \mathbf{q}^{CB} \otimes \mathbf{q}^{BA} \tag{7}$$

Assume $\mathbf{u}^A$ is a vector described in coordinate A. A 0 is inserted to this vector to make it a row vector containing 4 elements. Given the relative orientation of coordinate B represented with $\mathbf{q}^{AB}$, the same vector described in coordination B is expressed in Eq. (8).

$$\mathbf{u}^B = \mathbf{q}^{BA} \otimes \mathbf{u}^A \otimes \left(\mathbf{q}^{BA}\right)^* \tag{8}$$

2.2 Pre-process of Raw Data

The quality of the raw data directly affects the estimation results, especially when the integration manipulation is involved. Thereby, the raw data should be pre-processed to filter the noise/error before being applied with the position tracking algorithm. There are several steps listed as below:

1) *Offset calibration*

As illustrated in Eq. (1), there is a slowly time-varying bias e_b in inertial sensor measurement, which can be roughly eliminated by subtracting the mean of the data points when the sensor is stationary. This should be done each time when the sensor is powered up.

2) *Random Peak elimination*

In the measurements, there are lots of random peaks which are much greater than peripheral data. This peak can be eliminated as expressed in Eq. (9)

$$\hat{a}(i)=\frac{1}{2}\left[a(i-1)+a(i+1)\right] \text{ if } \left|a(i)-a(i-1)\right|>h_1 \text{ and } \left|a(i)-a(i+1)\right|>h_1 \tag{9}$$

$a(i)$ is the measurement, $\hat{a}(i)$ is the updated result after manipulation, and h_1 is the threshold.

3) *Mechanical and Moving average Filter*

In the measurements, there are also random error e_n as introduced in Eq. 1. Part of these errors is relatively small and isolate around zero, which is a filter with mechanical filter expressed with Eq. (10a). Meanwhile, due to the character of zero mean, moving filter defined in Eq. (10b) is utilized.

$$\hat{a}(i)=0 \text{ if } \left|a(i)\right|<h_2 \tag{10a}$$

$$\hat{a}(i)=\frac{1}{M}\sum_{j=0}^{M-1}a(i+j) \tag{10b}$$

Where h_2 is the threshold, M is the size of the filter.

2.3 Orientation Estimation

Theoretically, given an IMU, the orientation of the moving sensor can be easily estimated by integrating the angular rate (gyroscope reading), detecting the direction of gravitational acceleration or geomagnetic field. However, it becomes challenging when measurement errors and underground magnetic anomalies are introduced. A reliable and accurate filter of orientation estimation which fuses two estimations from the IMU similarly to the complementary filter is introduced. Because of the unknown and complicated underground condition, the magnetometer which is normally used to determine the heading will not be utilized.

2.3.1 Orientation Estimation with Angular Velocity

Ideally, the orientation can be estimated by integrating the angular velocity. Even though a gyroscope can provide the angular velocity, these measurements are with respect to the sensor coordinate, which needs to be converted to the navigation coordinate. Quaternion representation is utilized for this part.

With the measurements of the gyroscope $\boldsymbol{\omega}_{ns}^{s}$ represented in quaternion form shown in Eq. (11a), the time rate of the orientation of the navigation frame relative to the sensor frame expressed with quaternion can be calculated using Eq. (11b).

$$\boldsymbol{\omega}_{ns}^{s} = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \end{bmatrix}, \dot{\mathbf{q}}^{ns} = \frac{1}{2}\mathbf{q}^{ns} \otimes \boldsymbol{\omega}_{ns}^{s} \qquad \text{(11a, b)}$$

By numerically integrating the quaternion derivative $\dot{\mathbf{q}}^{ns}$, the orientation of the navigation frame relative to the sensor frame $\mathbf{q}_t^{ns}$ can be calculated with Eq. (12).

$$\dot{\mathbf{q}}_{\omega,t}^{ns} = \frac{1}{2}\mathbf{q}_{est,t-1}^{ns} \otimes \boldsymbol{\omega}_{ns,t}^{s}, \ \mathbf{q}_{\omega,t}^{ns} = \mathbf{q}_{est,t-1}^{ns} + \dot{\mathbf{q}}_{\omega,t}^{ns}\Delta t \qquad (12a, b)$$

Where Δt is the time interval, $\mathbf{q}_{est,t-1}^{ns}$ is the previous orientation estimation.

2.3.2 Orientation Estimation with Gravitational Acceleration

An accelerometer measures the acceleration of the moving sensor frame relative to the inertial frame with respect to the sensor frame, which consists of two parts, gravitational acceleration and linear acceleration due to the motion. Normally, the latter is much smaller than the former, which make it possible to simply assume the accelerometer only measure the gravitational acceleration.

The task to estimate orientation with gravitational acceleration can be converted to a typical optimization problem expressed in Eq. (13).

$$\min_{\mathbf{q}^{ns} \to \square^4} f\left(\mathbf{q}^{ns}, \mathbf{a}_{ns}^{n}, \mathbf{a}_{ns}^{s}\right),$$

$$\text{Objective function: } f\left(\mathbf{q}^{ns}, \mathbf{a}_{ns}^{n}, \mathbf{a}_{ns}^{s}\right) = \left(\mathbf{q}^{ns}\right)^{*} \otimes \mathbf{a}_{ns}^{n} \otimes \mathbf{q}^{ns} - \mathbf{a}_{ns}^{s} \qquad (13a, b, c)$$

$$\mathbf{q}^{ns} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}, \text{ constraint: } \left\|\mathbf{q}^{ns}\right\| = 1$$

Where $\mathbf{a}_{ns}^{n} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$ and $\mathbf{a}_{ns}^{s} = \begin{bmatrix} 0 & a_x & a_y & a_z \end{bmatrix}$ is the gravitational acceleration with respect to the inertial frame (navigation frame) and sensor frame. $\mathbf{a}_{ns}^{s}$ is the measurement of the accelerometer.

The gradient descent algorithm is applied to obtain the sensor frame orientation represented with $\mathbf{q}^{ns}$, which is mathematically represented in Eq. (14).

$$\mathbf{q}_{k+1}^{ns} = \mathbf{q}_{k}^{ns} - \mu \frac{\nabla f\left(\mathbf{q}_{k}^{ns}, \mathbf{a}_{ns}^{n}, \mathbf{a}_{ns}^{s}\right)}{\left\|\nabla f\left(\mathbf{q}_{k}^{ns}, \mathbf{a}_{ns}^{n}, \mathbf{a}_{ns}^{s}\right)\right\|}, k = 0,1,2,...n$$

$$\nabla f\left(\mathbf{q}_{k}^{ns}, \mathbf{a}_{ns}^{n}, \mathbf{a}_{ns}^{s}\right) = \mathbf{J}^{T}\left(\mathbf{q}_{k}^{ns}, \mathbf{a}_{ns}^{n}\right) f\left(\mathbf{q}_{k}^{ns}, \mathbf{a}_{ns}^{n}, \mathbf{a}_{ns}^{s}\right) \quad \text{(14a, b, c, d)}$$

$$f\left(\mathbf{q}_{k}^{ns}, \mathbf{a}_{ns}^{n}, \mathbf{a}_{ns}^{s}\right) = \begin{bmatrix} 2\left(q_2 q_4 - q_1 q_3\right) - a_x \\ 2\left(q_1 q_2 + q_3 q_4\right) - a_y \\ 2\left(0.5 - q_2^2 - q_3^2\right) - a_z \end{bmatrix}, \mathbf{J}^{T}\left(\mathbf{q}_{k}^{ns}, \mathbf{a}_{ns}^{n}\right) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix}$$

$\mathbf{q}_{n+1}^{ns}$ represents the orientation of the sensor which will be calculated after n iteration with the initial guess of $\mathbf{q}_{0}^{ns}$ and step size μ. The Jacobian $\mathbf{J}$ of the objective function f is involved to calculate the gradient of f.

In order to decrease the calculation for the optimization, instead of doing multiple iterations of calculation to calculate the orientation using each measurement, one iteration is conducted for each measurement and the result is used for the iteration of the next measurement as illustrated in Eq. (15a) with an assumption that the convergence rate μ_t is equal or greater than the physical rate of change of orientation expressed in Eq. (15c)

$$\mathbf{q}_{\nabla,t}^{ns} = \mathbf{q}_{est,t-1}^{ns} - \mu_t \frac{\nabla f}{\|\nabla f\|} \quad \text{(15a, b, c)}$$

where $\nabla f = \mathbf{J}^{T}\left(\mathbf{q}_{est,t-1}^{ns}, \mathbf{a}_{ns}^{n}\right) f\left(\mathbf{q}_{est,t-1}^{ns}, \mathbf{a}_{ns}^{n}\right), \quad \mu_t = \alpha \left\|\dot{\mathbf{q}}_{t}^{ns}\right\| \Delta t, \alpha > 1$

The orientation $\mathbf{q}_{t}^{ns}$ at time t is calculated with the previous orientation estimation $\mathbf{q}_{est,t-1}^{ns}$ and the gradient of the objective function ∇f which is calculated with Eq. 15(b).

2.3.3 Filter Fusion Algorithm

With the measurements of IMU, the orientation can be calculated with Eq. (12) and (15). Like the complementary filter, a fusion method for orientation estimation is described in Eq. (16) by introducing a weight constant γ_t.

$$\mathbf{q}_{est,t}^{ns} = \gamma_t \mathbf{q}_{\nabla,t}^{ns} + (1-\gamma_t)\mathbf{q}_{\omega,t}^{ns}, 0 \le \gamma_t \le 1 \tag{16}$$

The value of γ_t can be determined by ensuring the weighted divergence rate of $\mathbf{q}_{\omega,t}^{ns}$ (β) equal to the weighted convergence rate of $\mathbf{q}_{\nabla,t}^{ns}$ ($\mu_t/\Delta t$), which is represented in Eq. (17a) or rearranged in Eq. (17b) can be expressed as the magnitude of a quaternion derivative due to the measurement error.

$$\gamma_t \mu_t/\Delta t = (1-\gamma_t)\beta \text{ or } \gamma_t = \frac{\beta}{\mu_t/\Delta t + \beta} \tag{17a, b}$$

Since α defined in Eq. (15c) has no upper bound, μ_t could be very large and Eq. (17b) could be simplified as $\gamma_t = \beta\Delta t/\mu_t$, which is very small. Thereby, Eq. (16) can be simplified as below by substituting Eq. (12b) and (15a) and expressed in Eq. (18).

$$\mathbf{q}_{est,t}^{ns} = \frac{\beta\Delta t}{\mu_t}\left(-\mu_t \frac{\nabla f}{\|\nabla f\|}\right) + (1-0)\left(\mathbf{q}_{est,t-1}^{ns} + \dot{\mathbf{q}}_{\omega,t}^{ns}\Delta t\right) \tag{18}$$

By rearranging Eq. (18), a filter is derived as shown in Eq. (19a), where $\dot{\mathbf{q}}_{est,t}^{ns}$ is the estimated rate of change of orientation and $\dot{\mathbf{q}}_{e,t}^{ns}$ is the estimated error of $\dot{\mathbf{q}}_{est,t}^{ns}$.

$$\mathbf{q}_{est,t}^{ns} = \mathbf{q}_{est,t-1}^{ns} + \dot{\mathbf{q}}_{est,t}^{ns}\Delta t$$

$$\text{where } \dot{\mathbf{q}}_{est,t}^{ns} = \dot{\mathbf{q}}_{\omega,t}^{ns} - \beta\dot{\mathbf{q}}_{e,t}^{ns}, \quad \dot{\mathbf{q}}_{e,t}^{ns} = \nabla f/\|\nabla f\| \tag{19a, b, c}$$

The filter estimates the orientation by numerically integrating $\dot{\mathbf{q}}_{est,t}^{ns}$, which is estimated with Eq. (19b). The difference between (12a) and (19a) is that the error of the rate of change of orientation obtained from the gyroscope $\dot{\mathbf{q}}_{\omega,t}^{ns}$ is removed by involving the accelerometer measurements.

2.4 Position Estimation

The distance of travel of the sensor can be easily obtained by involving the encoder readings. The trajectory of the moving sensor can be easily estimated by combining the real-time encoder measurements and estimation of the sensor frame.

Assume the sensor always moves along the *x* axis of the sensor frame and distance of travel at time *t* is Δd_t provided by the encoder. Thereby, the change of position with respect to the sensor frame $\Delta\mathbf{r}_t^s$ at time *t* is represented in Eq. (20a), which can be transformed to the navigation frame using the orientation estimation $\mathbf{q}_{est,t}^{ns}$ with Eq. (20b). The estimation of the sensor position is calculated by adding the change of the position for each time period as illustrated in Eq. (20c). Originally the sensor frame is assumed to coincide with the navigation frame, thereby, $\mathbf{r}_{est,0}^n = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$ when $t = 0$.

$$\Delta\mathbf{r}_t^s = \begin{bmatrix} 0 & \Delta d_t & 0 & 0 \end{bmatrix},\ \Delta\mathbf{r}_{\Delta t}^n = \mathbf{q}_{est,t}^{ns} \otimes \Delta\mathbf{r}_t^s \otimes \left(\mathbf{q}_{est,t}^{ns}\right)^*,\ \mathbf{r}_{est,t}^n = \mathbf{r}_{est,t-1}^n + \Delta\mathbf{r}_t^n \qquad (20a, b, c)$$

Since the sensor moves along the *x* axis of the sensor frame, the rotation about the *x* axis (roll angle) will not affect the position estimation and the *x*-axis error of the gyroscope has less effect on the position estimation. The rotation about the y axis (pitch angle) will affect

the elevation (z component of the position vector) whereas the rotation about the z axis (yaw angle) will affect the heading (x and y component of the position vector).

3. Design of Sensing System for 3D Underground Conduit Mapping

With the method developed in the previous section, a sensing system for 3D underground conduit mapping is designed as manifested in Fig. 8. An IMU with power is packed in a designated IP68 metal housing. A rope is hooked onto the device and the other side of the rope is blown through the duct all the way to the other side using an optic fiber blower. The rope on the other side of the duct is hooked to a controllable motor with an encoder. The device is then inserted into one end of the duct, powered on, and left stationary for several seconds for offset calibration. A simplified flowchart of data acquisition and processing is shown below in Fig. 7. Since real-time underground conduit mapping is not required, all the IMU and encoder data is stored in SD card and analyzed and processed offline to construct an accurate 3D layout of the underground conduit with the method presented in the previous section.

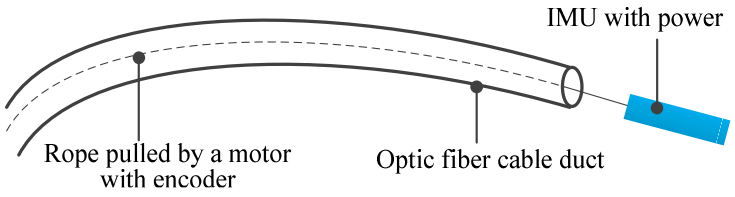


Figure 7. Schematics of Sensing system for 3D underground mapping

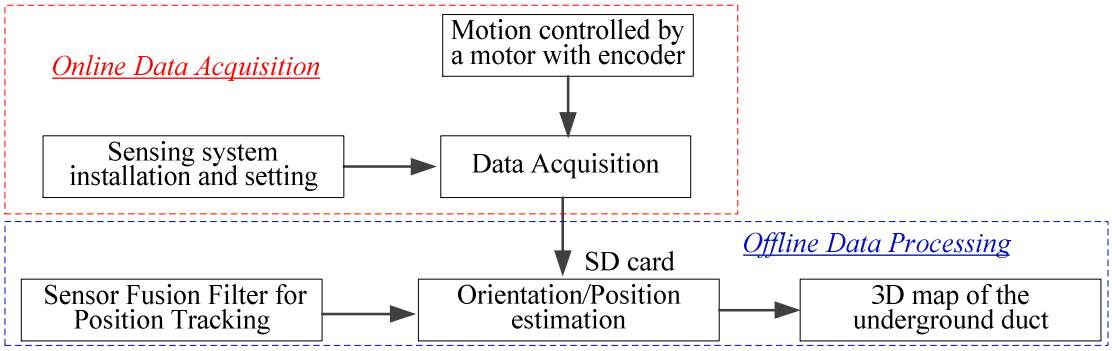


Figure 8. Flowchart of the data acquisition and processing

Even though the main algorithm for position tracking has been presented in the previous section, there are several additional improvements designed based on the application.

1. Motion detection (starting/ending points)

   After inserting the sensor into the duct, the sensor should be station for calibration, the station phase is then followed by the controlled motion through the duct. It is very important to be able to identify and crop the data for the calibration phase and the duct mapping phase, which is based on the detection and identification of the motion of the sensor. As illustrated in Fig. 9, by introducing a user-defined threshold h_3 illustrated by two red dash line, k_1 and k_2, which are the starting and ending points of the motion, can be automatically detected based on the acceleration measurement. The acceleration will be within a user-defined threshold h_3 if the sensor does not move.

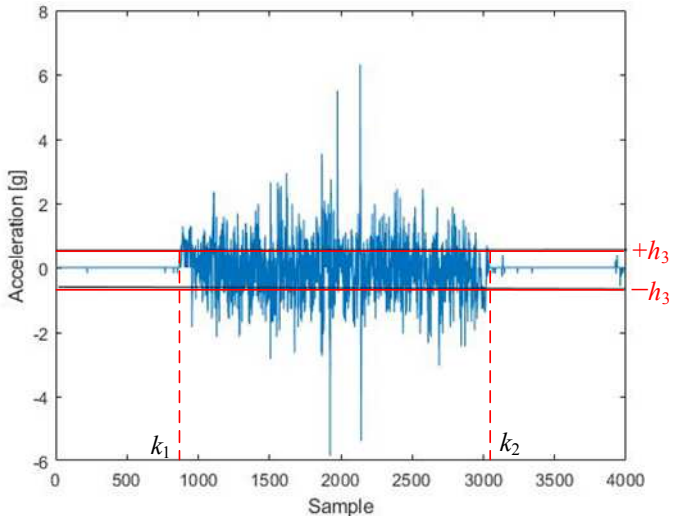


Figure 9. Illustration of motion detection

2. Gyroscope drifting correction

Gyroscopes' errors consists of two parts, slow time-varying bias and random error with zero mean as mathematically represented in Eq. (1). The constant offset within the error can be easily eliminated by calibration procedure explained in the previous section. Considering the working principle of the gyroscope for orientation estimation (integration manipulation), the random error with zero mean has little effect on the estimation results, which is not true for the time-varying bias. Thereby, an accelerometer, which detects the direction of the gravitational acceleration, is involved to correct parts of the drifting effect of the gyroscope with the proposed filter. The drifting effect on the heading angle (yaw), normally corrected by a magnetometer (not available in this application), is still unsolved.

Contrary to commonly seen applications that process live orientation and position information, offline processing of data can be done instead. All the data is accessible

for the orientation and position estimation. In this research project, the orientation and/or position of the starting and ending point may be available. A new refinement method is developed and implemented by following the procedures as below.

Assume the orientation at the starting and ending points are known and represented with quaternions $\mathbf{q}_{true,start}^{ns}$ and $\mathbf{q}_{true,end}^{ns}$, respectively.

1) Sensor drifting estimation and compensation

Since the output of the gyroscope at the starting point in stationary is zero after the calibration procedure, the measurement of the gyroscope at the ending in stationary is the sensor drift $\boldsymbol{\omega}_d$.

Assume the sensor output linearly drifts with time elapsing, the compensated measurements $\boldsymbol{\omega}_{com,t}$ at time t can be obtained with Eq. (21)

$$\boldsymbol{\omega}_{com,t} = \boldsymbol{\omega}_{meas,t} - \frac{t}{T}\boldsymbol{\omega}_d \tag{21}$$

Where T is the time for data collecting and $\boldsymbol{\omega}_{meas,t}$ is the measurement at time t.

2) Orientation refinement

Since all the data is available before processing, it is possible to do the orientation estimation in the opposite direction (from the ending point to starting point). Thereby, two sequences of orientation estimation can be calculated. Because of the estimation principle (numerical integration), the accuracy of the estimation will decline over time. Inspired by the complementary filter, the orientation can be estimated with Eq. (22), which involves two weight coefficients related to t and T.

$$\mathbf{q}_{est,t}^{ns} = \left(1 - \frac{t}{T}\right)\left(\mathbf{q}_{est,t}^{ns}\right)_{start} + \frac{t}{T}\left(\mathbf{q}_{est,t}^{ns}\right)_{end} \tag{22}$$

Where $\left(\mathbf{q}_{est,t}^{ns}\right)_{start}$ and $\left(\mathbf{q}_{est,t}^{ns}\right)_{end}$ are the orientation estimation at time *t* starting from the starting and the ending points, respectively.

The orientation at the starting and ending points represented with quaternion $\mathbf{q}_{true,start}^{ns}$ and $\mathbf{q}_{true,end}^{ns}$ respectively are utilized as the initial orientation (guess) for two estimations mentioned above.

3) Position correction

The information of the locations of the ending point can also benefit the position estimation if it is available. Given a constant orientation error, the position error increases linearly with the movement of the sensor. Thereby, the position estimation can be corrected using Eq. (23).

$$\overline{\mathbf{r}}_{est,t} = \mathbf{r}_{est,t} + \frac{d_t}{D}\left[\mathbf{r}_{true,end} - \mathbf{r}_{est,end}\right] \tag{23}$$

Where *D* is total displacement of the sensor for the whole data collecting process, d_t is the displacement of the sensor at time *t*, $\mathbf{r}_{true,end}$ is the location of the ending point, $\overline{\mathbf{r}}_{est,t}$ is the corrected position estimation at time *t*.

4. Experimental Implementation and Discussion

To verify and demonstrate the design of the algorithm and sensing system for 3D underground duct mapping presented in Chapter 2 and 3, several experiments have been

conducted and analyzed. The experimental setup, procedures, results and analysis will be provided in the following two sections.

4.1 Experimental setup description

Based on the design introduced in Chapter 3, a sensing system for 3D underground mapping has been implemented as illustrated in Fig. 10, which consists of a sensing unit, a duct rodder (08078545) and a fiber blowing machine (GS350). The sensing unit contains a sensing housing made of steel, an IMU (SEN-14001) provided by SparkFun and a 400 mAh Lithium-ion battery (13851). The sensing device is then mounted to the duct rodder, which has high mechanical strength and durability, using a small swivel that allows the sensing device to rotate about its own moving axis to prevent the high torsion generated in the duct rodder. The fiber blowing machine can push the sensing unit in the duct by pushing the duct rodder, which is placed between two gear belts driven by pneumatic motors. The PLC box in the blowing machine does not only control the speed of the pneumatic motors, but also measures and records the speed and distance of the rodder using an integrated encoder (Z7934-ND) with 1Hz sampling rate. Both the duct rodder and fiber blowing machine are manufactured by Condux. A 2” diameter conduit was utilized to perform the experiment as shown in Fig. 10.

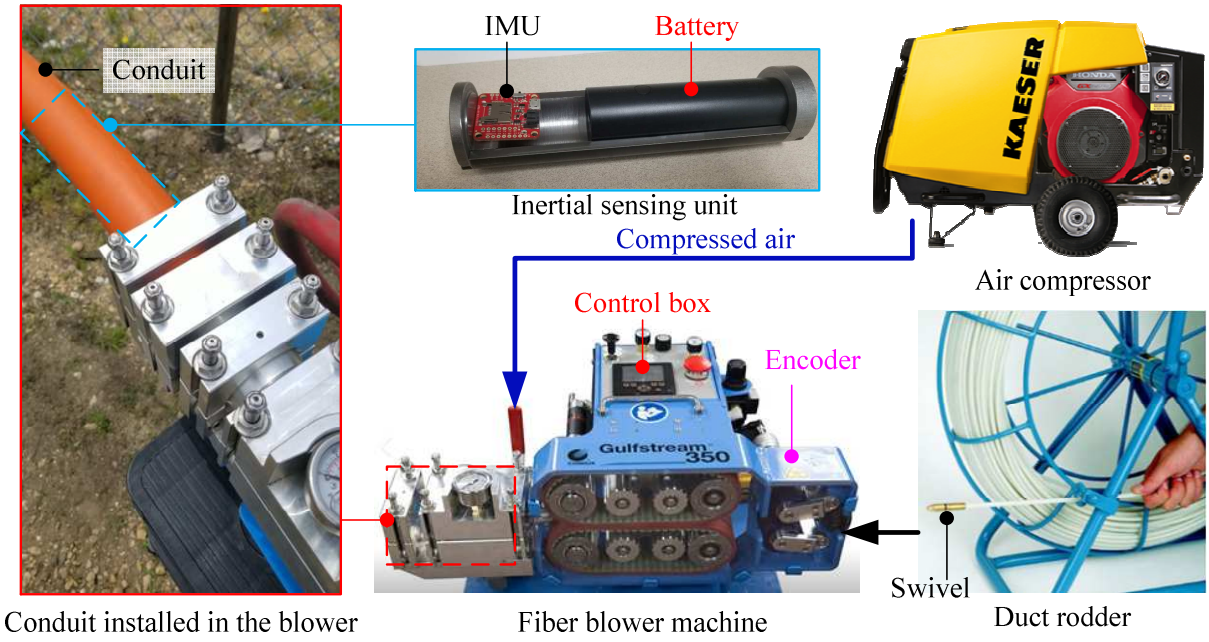


Figure 10. Experimental setup

In order to keep the high accuracy of the estimation, the IMU collects the data with the sampling rate of 100 Hz, which is not consistent with the sampling rate of the encoder (1 Hz). Since the same sampling rate is required for position tracking algorithm, additional data are estimated between each pair of the encoder's measurements with linear interpolation.

4.2 Experimental Procedures

The details of the experimental procedures are as follows:

- Blow two small sized foam pieces through the conduit to ensure the cleanness of the conduit.
- Power on the sensing unit and keep it in stationary for 2-3 minutes for calibration. A water level was used to make sure that the *z* axis of the IMU is exactly vertical during calibration.

- Power on the PLC box to record the reading of the encoder. Insert the sensing unit into the conduit and keep it in stationary for another 2-3 minutes to identify the starting point. Measure the yaw angle of the conduit/sensing unit with the cell phone magnetometer.
- Power on the fiber blowing machine (pneumatic motors) and slowly move the sensing unit through the conduit by feeding the duct rodder.
- Keep the sensing unit in stationary for another 2-3 minutes to identify the ending point when it moves to the end of the conduit. Measure the yaw angle of the conduit/sensing unit again.
- Store the IMU and encoder data in the computer and reset the IMU and fiber blowing machine for another run.

The measurements of the IMU and encoder are stored in an SD card internal flash driver respectively. These two sequences of data can be aligned by identifying the motion or starting/ending points. The motion of the sensing unit should be smooth, but there is no need to maintain absolute constant moving speed.

4.3 Experimental results

Several experiments were conducted with three different duct layouts with the dimensions and shapes shown in Fig. 11. The length of each section of the layouts (lines and curves) was measured with a walking wheel tape. Assume z component of all the layouts is zero except one section of layout3 represented with a red line shown in Fig. 11. The curve in *x*-*z* plane can be modeled as a Gaussian function with $\mu = 0, \sigma^2 = 1$. This

section was implemented by placing the conduit on a 0.8m, 0.5m table height and width, respectively, as shown in Fig. 13(a).

As mentioned in the previous section, the sensing device always moves along the local *x* axis of the sensing coordinate system (*xyz*). For each trial, the sensing unit originally sits at the origin of the navigation coordinate system (*XYZ*) with the initial orientation shown in Fig. 11. Fig. 12(a), (b) and (c) compare the experimental results and actual layouts in *x*-*y* plane whereas Fig. 13 (b) compares the results of part of Layout3 in *y*-*z* plane. The mean and standard deviation (STD) of the estimation errors are shown in Table I.

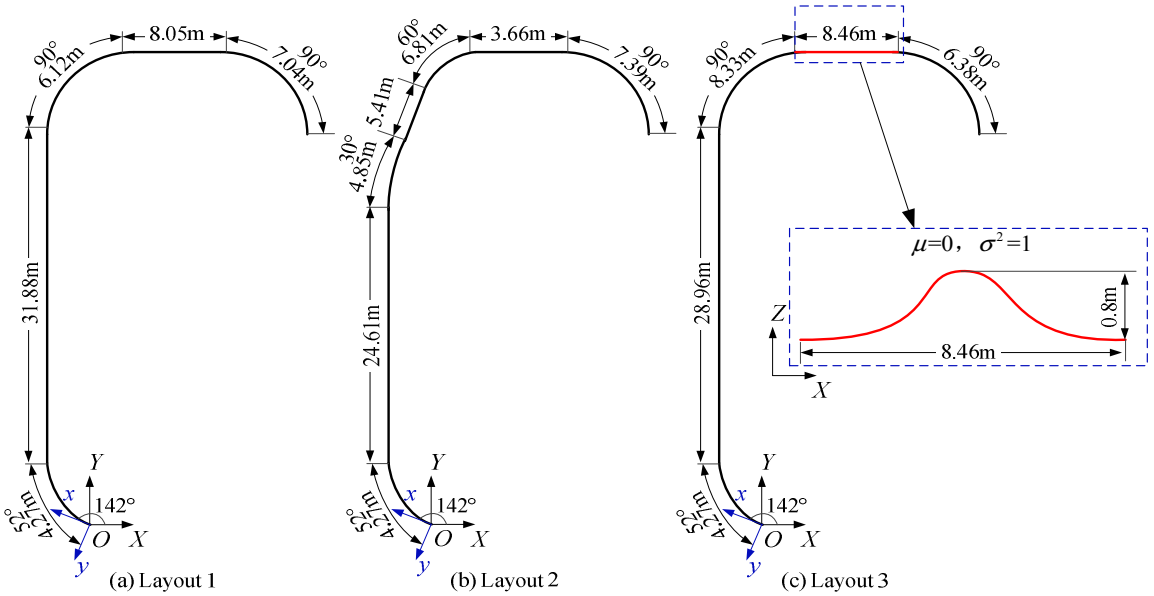
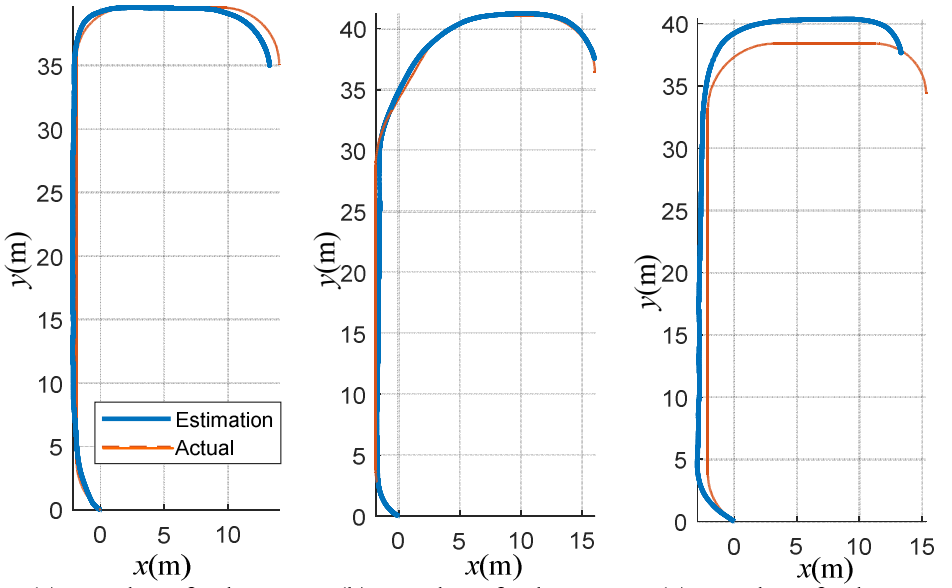
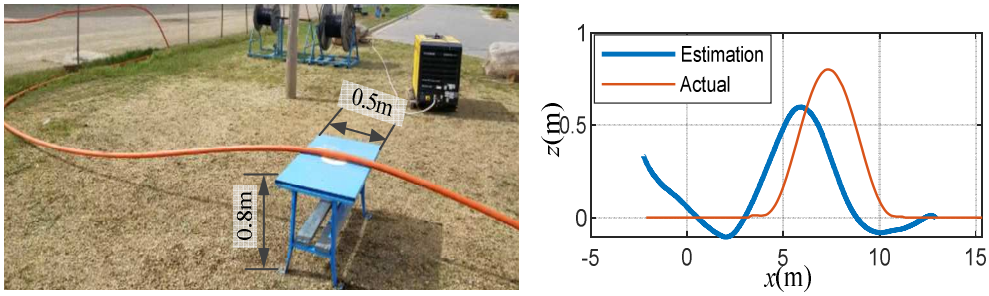


Figure 11. Schematics and dimensions of three layouts



(a) *x*-*y* plane for layout 1 (b) *x*-*y* plane for layout 2 (c) *x*-*y* plane for layout 3

Figure 12. Comparison of the estimation results and actual layouts in *x*-*y* plane



(a) part of layout 3 (b) *x*-*z* plane for layout 3

Figure 13. Part of layout 3 and corresponding results (*x*-*z* plane)

Table 1. Estimation errors of three layouts

	Layout 1	**Layout2**	**Layout3**
Mean (m)	0.3917	−0.1230	−0.1522
STD (m)	0.5975	0.2109	1.0870

4.4 Discussion and Analysis

Some observations can be made from the experimental results illustrated in Figs. 12, 13 and Table I.

- The 3D mapping of the conduit can be reconstructed with the sensing system developed based on the new designed 3D tracking system.
- The tracking errors increase with the time elapse or displacement increase.
- As illustrated in Fig. 12, the estimation results in *x*-*y* plane of layout1 and 2 are better than the results of layout3, which includes additional *z* axis (3D tracking).
- The vertical motion of the sensing unit can be reconstructed as shown in Fig. 13(b). The large error close to $x = 0$ and peak shift are caused by the poor estimation of the previous section.
- The orientation and value of *z* component at the ending location are successfully pulled to 90° and 0 with Eqs. (22) and (23) respectively.

The accuracy of the 3D position tracking mainly depends on the accuracy of the orientation estimation. Even a small deviation of the angle will result in a tremendous error with the increase of the travel distance. The experimental errors at certain sections of the layouts are relatively large especially for layouts 3. The errors may be caused by the following factors: 1) uncompensated sensor drift, 2) unsuccessful calibration and 3) measurement error.

5. Conclusion and Future Work

The use of MEMS sensors have notably increased over the past decades due to their high demand in various localization related applications. A new algorithm has been developed and experimented in this research. The algorithm was developed bas on the use of an IMU sensor that gathered information about acceleration and orientation at 100 Hz. The encoder, that gathered information about distance at 1 Hz, is also a critical input to the algorithm. Three different duct layouts were utilized to test the algorithm. The first two duct layouts were on the xy-plane and the third duct layout was on the xyz-plane. The analysis showed higher accuracy on the xy-plane versus the xyz-plane. Uncompensated drift of the IMU, inaccurate distance data, and imprecise duct layout measurements are all possible causes of the resulting error. The standard deviations of the estimated trajectories from the actual duct layouts were 0.5975, 0.2109, and 1.0870 in layouts 1, 2, and 3, respectively. The error is expected to increase over longer ducts. A possible solution to this problem is to frequently bring the sensing device to a complete stop for re-calibrating the sensing system. A potential improvement is to also use an alternative to the GS350 machine to blow the sensing device through the conduit, since it was only used in this application to utilize its encoder. Finding an alternative to the GS350 machine would save a big amount of money to whoever is interested in performing a similar experiment. Instantaneous transmission of data from the sensing device to a computer is also a potential improvement in this research project. Live analysis of data would allow the developer to monitor the errors as they arise and eliminate them. This would definitely decrease the sensor accumulative errors. Live analysis of the data could be obtained only if the sensor

is connected to the computer during the complete run. This could be achieved by custom manufacturing a duct rodder with a wire in it that transmits the data instantaneously. Future cooperation between Condux, the sponsor of the research, and potential researchers at the department of Mechanical and Civil Engineering at Minnesota State University, Mankato (MNSU) would positively affect the outcome of this research project.

6. References

[1] Fiber Optic Connector Identifier. [Online]. Available: http://www.thefoa.org/tech/ref/OSP/install.html. [Accessed: 14-Jan-2019].

"Japan", IMV Corporation. [Online]. Available:

[2] Https://www.imv.co.jp/e/products/vibrograph/pickup/e-comp/. [Accessed: 10-Jun-2019].

[3] A. Albarbar, Samir Mekid, Andrew Starr, Pietriszkiewicz Robert, "Suitability of MEMS Accelerometers for Condition Monitoring: An experimental study". 2018

[4] "Capacitive Accelerometer," Capacitive Accelerometer - an overview | ScienceDirect Topics. [Online]. Available: https://www.sciencedirect.com/topics/engineering/capacitive-accelerometer. [Accessed: 08-Jul-2019].

[5] B. Lent, "Simple Steps to Selecting the Right Accelerometer," FierceElectronics, 01-Mar-2009. [Online]. Available: https://www.sensorsmag.com/components/simple-steps-to-selecting-right-accelerometer. [Accessed: 08-Jul-2019].

[6] Masteradmin, "Know About Gyroscopes and Their Types," Aeron Systems, 27-Aug-2018. [Online]. Available: https://www.aeronsystems.com/gyroscopes-and-their-types/. [Accessed: 08-Jul-2019].

[7] "The difference between ring laser gyro and fiber optic gyro," Welcome to SenNav. [Online]. Available: http://www.sennavs.com/the-difference-between-ring-laser-gyro-and-fiber-optic-gyro/. [Accessed: 08-Jul-2019].

[8] E. Cheilakou, P. Theodorakeas, M. Koui, S. Moustakidis, C. Zeris, "Determination of Reinforcement and Tendon Ducts Positions on Pre-Stressed Concrete Bridges by Means of Ground Penetrating Radar (GPR)" 2013

[9] Tran, Long. "Data Fusion with 9 Degrees of Freedom Inertial Measurement Unit to Determine Object's Orientation. 2017

[10] Sebastian O.H. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays" 2010

[11] Sebastian O.H. Madgwick, Andrew J.L. Harrison, Ravi Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm” ETH Zurich Science City, Switzerland. 2011

[12] U-Xuan Tan, Kalyana C. Veluvolu, Win Tun Latt, Cheng Yap Shee, Cameron N. Riviere, and Wei Tech Ang, , “Estimating Displacement of Periodic Motion With Inertial Sensors” IEEE Sensors Journal, VOL. 8, No.8. August 2018

[13] Win Tun Latt, Kalyana Chakravarthy Veluvolu, Wei Tech Ang. “Drift-Free Position Estimation of Periodic or Quasi-Periodic Motion Using Inertial Sensors.”Hamlyn Centre for Robotic Surgery, and Department of Computing, Imperial College Longon. May 2011

[14] Charlotte Treffers Luc Van Wietmarschen. “Position and Orientation Determination of a Probe with Use of the IMU MPU9250 and a ATmega328 Microcontroller.” TU Delft. June 2016

[15] Nebot E., Durrant-Whyte H., “Initial Calibration and Alignment of Low-Cost Inertial Navigation Units for Land Vehicle Applications”, Journal of Robotics Systems, Vol. 16, No. 2, pp. 81-92. February 1999

[16] Benoit Huyghe, Jan Vanfleteren, Jan Doutreloigne., “3D Orientation Tracking Based on Unscented Kalman Filtering of Accelerometer and Magnetometer Data.”10.1109/SAS.2009.4801796, IEEE. 2009

[17] Manon Kok, Jeroen D. Hol and Thomas B. Schon, “Using Inertial Sensors for Position and Orientation Estimation”, Foundations and Trends in Signal Processing: Vol. 11: No. 1-2, pp 1-153. http://dx.doi.org/10.1561/2000000094. 2017

[18] Tim Bailey, Hugh Durrant-whyte. “Simultaneous Localization and Mapping (SLAM): Part II.” IEEE Robotics and Automation Magazine. 1070-9332-06. 2006

[19] Aleksandr Aravkin, Gianluigi, James Vincent Burke. “Optimization Viewpoint on Kalman Smoothing with Applications to Robust and Sparse Estimation. DOI: 10.1007/978-3-642-38398-4_8. arrXiv. 2014

[20] Rahul Jain, Mark Epelbaum. “Filtering Data” EE444 S’11. August 19th, 2011

[21] Toshak Singhal, Akshat Harit, D N Vishwakarma. “Kalman Filter Implementation on an Accelerometer Sensor Data for Three State Estimation of a Dynamic System.

International Journal of Research in Engineering and Technology (IJRET) Vol. 1, No. 6 ISSN 2277-4378. 2012

[22] Moses A. Koledoye, Daniele De Martini, Massimo Carvani, Tullio Facchinetti. "Design of a Mobile Robot for Air Duct Exploration" 2017

[23] Xiaoping Yun, Eric R. Bachmann, Hyatt Moore IV, and James Calusdian, "Self-contained Position Tracking of Human Movement Using Small Inertial/Magnetic Sensor Modules" IEEE International Conference on Robotics and Automation, Roma, Italy, 2007

[24] Yun X, Bachmann ER. Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking. IEEE Trans. Rob. 22:1216–1227. 2006

[25] Ang WT, Khosla PK, Riviere C. Kalman Filtering for Real-time Orientation Tracking of Handheld Microsurgical Instrument. Proceedings of IEEE International Conference on Robotics Systems; Sendai, Japan; pp. 2574–2580. 2014

[26] Eric L. Haseltine, James B. Rawlings. "A Critical Evaluation of Extended Kalman Filtering and Moving Horizon Estimation." Technical Report. Texas-Wisconsin Modeling and Control Consortium. March 2003

7. Appendix: Critial MATLAB Codes

1. Main function

```
clear
clc
%Data setting
%Trial2
data2Old= xlsread('trial2IMU.xlsx');
Data= xlsread('trial2PLC.csv');
startEncoder = 264;  % starting index of encoder reading
endEncoder = 390;   % ending index of encoder reading
k1 = 20054; % starting index of IMU
k2 = 31682; % ending index of IMU
startCalibration = 16601; % starting index of calibration data
endCalibration = 32100; % starting index of calibration data
angleStart = (270-56)/180*pi; %Yaw angle at the starting location
angleEnd = 90/180*pi;   %Yaw angle at the ending location
paraOri = 0.05; %  weight coefficient  paraOri = 0, orienation
estimated only from Gyro, with the increase of coefficent, more weight
on accelerometer
distanceData= Data(startEncoder:endEncoder,6);
addpath('../Quaternions');
%time index (unit: sec)
data2=data2Old(k1:k2,:);
dataTime=data2(:,1)/1000;

dataTime = dataTime-dataTime(1);
dataGyo=[];
aa = [80 80 80];
bb = [0.1 0.1 0.1];
driftGyro = zeros(3,1);
timeDrift = (data2Old(endCalibration,1) -
data2Old(startCalibration,1))/1000;
for mm = 1:3
    data = data2(:,(mm+4));
    for i=2:(length(data)-1)
     if abs(data(i)-data(i-1))>aa(mm) & abs(data(i)-data(i+1))>aa(mm)
        data(i)=(data(i-1)+data(i+1))/2;
     end
    end


meanBefStart=sum(data2Old(startCalibration:startCalibration+199,mm+4))/
200;

meanBefEnd=sum(data2Old(endCalibration:endCalibration+199,mm+4))/200;
    driftGyro(mm) = meanBefEnd-meanBefStart;
    data=data-meanBefStart;   % drift angular velocity
    %mechanical filter
    for i=1:length(data)
        if abs(data(i))<=bb(mm)
            data(i)=0;
        end
```

```
    end

    % correct the drift
    data = data - dataTime/timeDrift*driftGyro(mm);
    dataGyo=[dataGyo data];

end


dataAcc=[];
aa = [0.5 0.5 0.5];
bb = [0.02 0.02 0.02];

for kk = 1:3
    data = data2(:,(kk+1));
    %eliminate the arbitrary peaks
    for i=2:(length(data)-1)
        if abs(data(i)-data(i-1))>aa(kk) & abs(data(i)-
data(i+1))>aa(kk) & sign(data(i)-data(i-1))*sign(data(i)-data(i+1)) ==
1
            data(i)=(data(i-1)+data(i+1))/2;
        end
    end
    % calibration

meanBef=sum(data2Old(startCalibration:startCalibration+199,kk+1))/200;
    data=data-meanBef;
    %mechanical filter
    for i=1:length(data)
        if abs(data(i))<=bb(kk)
            data(i)=0;
        end
    end

    if kk==3
        data=data+meanBef;  %No need to calibrate z axis of the
accelerometer
    end
     dataAcc=[dataAcc data];
end

dataGyo = dataGyo * (pi/180);
quaternion = zeros(length(dataTime), 4);

%initial orientation, quaternion format
tempQuaternionStart = [cos(angleStart/2) -sin(angleStart/2)*[0 0 1] ];
tempQuaternionEnd = [cos(angleEnd/2) -sin(angleEnd/2)*[0 0 1] ];
quaternionTemp(1,:,1) = [tempQuaternionStart];
quaternionTemp(length(dataTime),:,2) = [tempQuaternionEnd];

%smooth the acceleratаion measurement
dataAccQuat = [smooth(dataAcc(:,1), 5) smooth(dataAcc(:,2), 5)
smooth(dataAcc(:,3), 5)];
```

```
%Start->End
tempQuaternion = quaternionTemp(1,:,1);
for t = 2:length(dataTime)
    tempQuaternion = estQuaternion(tempQuaternion, dataGyo(t,:),
dataAccQuat(t,:),dataTime(t)-dataTime(t-1), paraOri);
    quaternionTemp(t, :, 1) = tempQuaternion;
end

%End->Start
tempQuaternion = quaternionTemp(length(dataTime),:,2);
for t = length(dataTime)-1:-1:1
    tempQuaternion = estQuaternion(tempQuaternion, -dataGyo(t,:),
dataAccQuat(t,:),dataTime(t+1)-dataTime(t), paraOri);
    quaternionTemp(t, :, 2) = -tempQuaternion;
end

% Orientation refinement
for i = 1: length(dataTime)
    quaternion(i,:) = (1-
dataTime(i)/(dataTime(end)))*quaternionTemp(i,:,1)+(dataTime(i)/(dataTi
me(end)))*quaternionTemp(i,:,2);
    quaternion(i,:) = quaternion(i,:)/norm(quaternion(i,:));
end

timeEncoder = 0:(length(distanceData)-1);
distance = interp1(timeEncoder,distanceData,dataTime,'liner')*0.3048;

%multiplying dv by the time difference:
for i=2:length(dataTime)
deltad=(distance(i)-distance(i-1));
dv(:,i)= quaternRotate([1 0 0]*deltad, quaternion(i,:));
end
pos=zeros(3,length(dataTime));
for i=2:length(dataTime)
pos(:,i)=pos(:,i-1)+dv(:,i);
end


%Position correction
zEnd = 0.5;
for i =1:length(distance)
    pos(:,i) = pos(:,i)+distance(i)/distance(end)*([pos(1,end);
pos(2,end); zEnd]-pos(:,end));
end

figure
plot3(pos(1,:),pos(2,:),pos(3,:));
axis equal

% layout measurement
load('layout1.mat');
hold on
```

```
plot(x,y);
```

2. Orientation estimation function

```
function [Quaternion] = estQuaternion(quaternion, Gyroscope,
Accelerometer, SamplePeriod, Beta)
%Estimate quaternion with the measurements of accelerometer and
gyroscope
% inputs:     quaternion        previous orientation estimation
%             Gyroscope         measurements of angular velocity
%             Accelerometer     measurement of acceleration
%             SamplePeriod      time period
%             Beta constant     coefficient
% output:     Quaternion        Updated orientation estimation

          q = quaternion; % short name local variable for readability
            % Normalise accelerometer measurement
            if(norm(Accelerometer) == 0), return; end   % handle NaN
            Accelerometer = Accelerometer / norm(Accelerometer);    %
normalise magnitude

            % Gradient decent algorithm corrective step
            F = [2*(q(2)*q(4) - q(1)*q(3)) - Accelerometer(1)
                2*(q(1)*q(2) + q(3)*q(4)) - Accelerometer(2)
                2*(0.5 - q(2)^2 - q(3)^2) - Accelerometer(3)];
            J = [-2*q(3),   2*q(4),    -2*q(1), 2*q(2)
                2*q(2),     2*q(1),     2*q(4), 2*q(3)
                0,         -4*q(2),    -4*q(3), 0    ];
            step = (J'*F);
            step = step / norm(step);   % normalise step magnitude

            % Compute rate of change of quaternion
            qDot = 0.5 * quaternProd(q, [0 Gyroscope(1) Gyroscope(2)
Gyroscope(3)]) + Beta * step';

            % Integrate to yield quaternion
            q = q + qDot * SamplePeriod;
            Quaternion = q / norm(q); % normalise quaternion

end
```