

UoA Undergraduate Mechatronics Research Journal

Volume 3 • December 2010

A Publication of Mechatronics Engineering Group,
Department of Mechanical Engineering,
The University of Auckland, New Zealand

ISSN 1171-9494

UoA Undergraduate Mechatronics Research Journal

EDITOR-IN-CHIEF

Kean C. Aw
k.aw@auckland.ac.nz

EDITORS

Enrico Haemmerle
e.haemmerle@auckland.ac.nz

Sheng Q. Xie
s.xie@auckland.ac.nz

UoA Undergraduate Mechatronics Research Journal is published annually by the Mechatronics Engineering Group of the Mechanical Engineering Department, The University of Auckland, 20, Symonds Street, Auckland, New Zealand.

Preface

Welcome to the Undergraduate Research Journal of the Mechatronics Engineering Group at the University of Auckland's Department of Mechanical Engineering. This third volume has a total of 7 articles based on some of the final year projects mechatronics engineering students have been working on during the year 2010. The journal offers undergraduates the chance to present a summary of their achievements and is published in December each year.

We hope you enjoy reading about the accomplishments of our students. This journal also provides an overview of the scope and capability of the Mechatronics Engineering at the University of Auckland. We like to thank those involved in supporting the publication of this journal.

Kean C. Aw
Editor-in-Chief

Table of Contents

A Simple Introduction to the Extended Kalman Filter with an Example Application <i>J. E. Donnelly</i>	1
Localisation and Path Tracking in a Steered-wheel Autonomous Vehicle <i>J. P. Jackson and J. E. Donnelly</i>	10
Low-cost Interactive Patient Rehabilitation Monitoring <i>T. Buranasetakul and N. Chang</i>	20
Low-cost Organic Thin Film Transistor Printer <i>T. Liu and Y. Sekikawa</i>	25
Sensor and Actuator Integration for Autonomous Guided Vehicle Vehicle <i>P. Hosking and J. Stroobant</i>	29
Temperature Sensing with SAW-based Sensor <i>T. J. Giffney and Y. H. Ng</i>	38
Development and Control of a Parallel Robot for Neurosurgery Applications <i>Josheel Pran Lal, Charlie Niu</i>	42

A Simple Introduction to the Extended Kalman Filter with an Example Application

J. E. Donnelly

The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract

This article serves as an introductory guide to the design and implementation of discrete-time Extended Kalman Filters. The Kalman Filter is an optimal recursive data processing algorithm for signal filtering and sensor fusion. The Extended Kalman Filter is merely an extension, which caters for nonlinear systems. It's underlying 'predictor-corrector' structure allows it to weight the effects of model-based predictions, and empirical data collected from sensors, to attain best estimates of state. An example application - localisation in Autonomous Ground Vehicles - is used for explanation throughout this article and MATLAB simulations show how the filter reduces the effects of noise in real and simulated sensor data, whilst amalgamating multiple sensor inputs. The intention is not to explain the probability theory behind Kalman's early derivations – more simply, it is to clarify the steps necessary for filter design and implementation, and to explain the advantages, intricacies and limitations of the filter for use in a wide range of applications.

Keywords: *Localisation, Sensor Fusion, Extended Kalman Filter, Autonomous Ground Vehicle*

1. Introduction

This article has been written with the intent to dispel any perceived difficulties surrounding the design and implementation of Extended Kalman Filters (EKF). All methodology featured in this article was extracted from final year undergraduate research into localisation and path-tracking in Autonomous Ground Vehicles (AGV) carried out by J. Donnelly and J. Jackson [2]. Explanations focus on the vehicle developed in said research. Exploratory discussion of further applications has been included where appropriate.

The Kalman Filter is an optimal recursive data processing algorithm [1] that was

developed in the late 1950s and published by R. E. Kalman in 1960 [3]. In traditional deterministic control theory, systems analysis is performed by making reasonable assumptions and carrying out empirical testing to develop mathematical models that explain the parameters of interest. In such systems, the effects of stochastic processes are often referred to as disturbances. Kalman sought to explain and account for these disturbances by incorporating probability theory within control theory [3] and hence the Kalman Filter was born. The Extended Kalman Filter is a simple extension of the basic form. It caters for nonlinear systems, as explained later.

Kalman Filters are used in AGVs to estimate the states (e.g. position, velocity, angular velocity, acceleration etc.) of the vehicle, particularly for localisation, from an array of sensors, while minimising the effects of noise in the sensors' outputs. For AGV applications, localisation is considered to be the process of finding a vehicle's current position in 2D. For aircraft, this process can be extended to 3D, or alternatively, aircraft height may be decoupled from 2D position as detailed by Mao et al. [4].

This article does not seek to completely explain the underlying probability theory in Kalman's early derivations, however further suggested reading for such explanations can be found in Maybeck's article [1].

2. Basic Kinematic Understanding

The AGV used in this research is shown in Figure 1 below. It is important to have a general understanding of the dynamics of the subject vehicle, i.e. to know whether it is differential drive or a steered-wheel robot etc., prior to developing a Kalman Filter for the system.

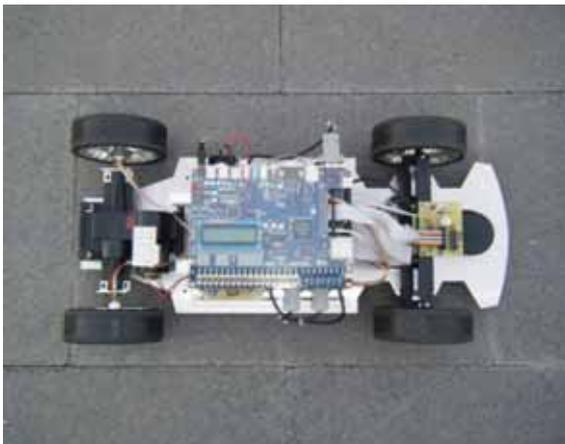


Figure 1. AGV Research Platform

Vehicle dynamics can be explained using planar, rigid body kinematics. A common kinematic model of steered-wheel vehicles

is depicted in Figure 2. Assuming no wheel-slip, the vehicle has three degrees of freedom. The vehicle is prevented from moving sideways by friction between the wheels and the road [5].

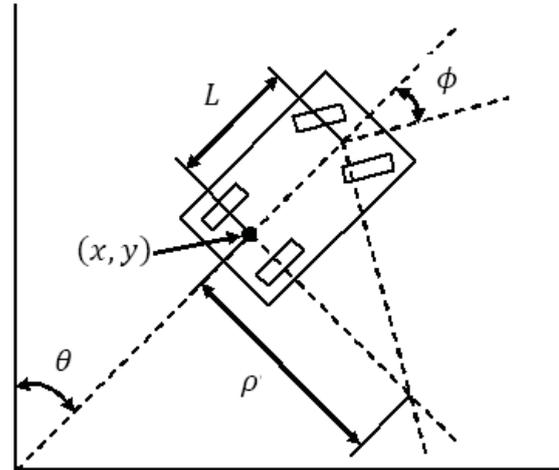


Figure 2. Kinematic model

Here, only three kinematic equations, Eqs. (1), (2) and (3), are needed to completely describe the motion of the vehicle in the 2D Cartesian coordinate frame:

$$\dot{x}(t) = v(t) \sin \theta(t) \quad (1)$$

$$\dot{y}(t) = v(t) \cos \theta(t) \quad (2)$$

$$\dot{\theta}(t) = \frac{v(t)}{L} \tan \phi \quad (3)$$

where $\{x(t), y(t)\}$ is the vehicle position in metres from a defined origin a Cartesian coordinate frame, $v(t)$ is vehicle speed in metres per second, $\theta(t)$ is the bearing with respect to true north in radians $[-\pi, \pi]$, L is the vehicle wheel base in metres and ϕ is the steering angle in radians.

This simple model is presented from a deterministic viewpoint and because speed, $v(t)$, and steering angle, ϕ , are considered as inputs, control efforts demand extremely accurate mathematical models of the drive motor (to find output speed with respect to motor control input), and of the steering

servo (to find steering angle with respect to servo control input). While this simple model may be used in a Kalman Filter, a more complex, but highly adaptable inertial-odometric model has been developed to create an unforced EKF, as will be explained in the next section.

3. Process and Observation Models

All Kalman Filters consist of two distinct models: a process model and an observation (measurement) model. The process model aims to explain vehicle dynamics. The observation model combines empirical measurements from AGV sensors in a process now commonly referred to as sensor fusion. It then uses existing statistical knowledge of the noise affecting the process, and the observations, to generate state estimates. Ultimately this algorithm seeks to minimise the variance of the state error, $\underline{e} = \hat{\underline{x}} - \underline{x}$, where $\hat{\underline{x}}$ is the state estimate and \underline{x} is the actual state.

The subject AGV, shown in Figure 1, is fitted with a GPS receiver for absolute position measurement, optical encoders (on rear axles) to measure speed, an analogue compass to measure the magnetic bearing and a yaw-rate gyroscope to measure angular velocity. As speed and angular velocity can be directly measured, a set of 7 state equations are now used to describe vehicle dynamics. They combine to make up the process model, which can be expressed by the continuous-time equations:

$$\dot{x}(t) = v(t) \sin \theta(t) + w_x(t) \quad (4)$$

$$\dot{y}(t) = v(t) \cos \theta(t) + w_y(t) \quad (5)$$

$$\dot{v}(t) = w_v(t) \quad (6)$$

$$\dot{\theta}(t) = \omega_\theta(t) \quad (7)$$

$$\dot{\omega}(t) = w_\omega(t) \quad (8)$$

$$\dot{b}_{encoder}(t) = w_{b_{encoder}}(t) \quad (9)$$

$$\dot{b}_{gyro}(t) = w_{b_{gyro}}(t) \quad (10)$$

where $\{x(t), y(t)\}$ is the AGV position in metres from a defined origin in an East-North-Up Cartesian coordinate frame (GPS coordinates are naturally in the WGS84 geodetic coordinate frame and must be converted into the local East-North-Up Cartesian coordinate frame), $v(t)$ is AGV speed in metres per second, $\theta(t)$ is the bearing with respect to true north in radians $[-\pi, \pi]$, $\omega(t)$ is angular velocity in radians per second, $b_{encoder}$ is a bias term introduced to account for wheel-slip and b_{gyro} is a bias term introduced to account for intrinsic and gravity biases in the gyroscope output. At this stage, AGV position is inferred from previous a-posteriori estimates of speed and bearing, while bearing is itself inferred from the previous a-posteriori estimate of angular velocity. $w_x(t)$, $w_y(t)$, $w_v(t)$, $w_\theta(t)$, $w_\omega(t)$, $w_{b_{encoder}}(t)$ and $w_{b_{gyro}}(t)$ are process noise terms, which represent disturbances to the plant [1]. They are assumed to be zero-mean, Gaussian (normally distributed), white noise terms, with known covariances, i.e. $p(w) \sim N(0, Q)$. AGV process noise is typically derived from acute terrain deviations and inaccurate or incomplete process models.

Notably, linear and angular accelerations, the rate of change of encoder bias and the rate of change of gyroscope bias are assumed to be solely affected by plant disturbances; hence the direct equivalence to their respective noise terms. The implications of these assumptions are discussed later.

Now, it is important to recognise that the Kalman Filter is an optimal state estimator for linear systems only. In this particular application the process model contains nonlinear terms (Eqs. (4) and (5)) so a modified version of the basic Kalman Filter, the EKF, must be used. The basic recursive

structure of the Kalman Filter remains, however in this extended form the filter does not generate optimal state estimates because the system has to be linearised about the most recent state estimate with a first order Taylor expansion. Subsequent state transition is considered as a perturbation about the linearization (equilibrium) point. While not optimal, the state estimates generated by an EKF are still much more useful than unfiltered sensor data (provided initial conditions are accurate).

This particular EKF is known as an unforced EKF, because there are no inputs to the system. Furthermore, the process model does not rely on specific platform characteristics to achieve state estimation. For instance, the use of a gyroscope means no prior knowledge of wheelbase is required in the measurement of angular velocity. Consequently, the EKF that has been proposed here is platform independent and only requires that any target platform includes equivalent measurement devices (sensors) to those mentioned earlier.

The discrete-time representation of the process model is known as the state transition equation and is shown in Eq. (11). The results produced from this equation are linear approximations of the nonlinear system outputs, found using the Euler method [4]:

$$\underline{x}_{k+1} = \underbrace{\begin{bmatrix} x_k + v_k \Delta t_k \sin \theta_k + w_x \\ y_k + v_k \Delta t_k \cos \theta_k + w_y \\ v_k + w_v \\ \theta_k + \Delta t_k \omega_k + w_\theta \\ \omega_k + w_\omega \\ b_{encoder_k} + w_{b_{encoder}} \\ b_{gyro_k} + w_{b_{gyro}} \end{bmatrix}}_{f(x_k, w_k)} \quad (11)$$

The discrete-time state-vector is:

$$\underline{x}_k = [x_k \quad y_k \quad v_k \quad \theta_k \quad \omega_k \quad b_{encoder_k} \quad b_{gyro_k}]^T$$

In this application, all states have zero initial conditions apart from the bearing, $\theta(0) = \theta_0$, which indicates the AGVs orientation with respect to true north prior to any movement. In discrete-time form, the -aforementioned assumptions regarding acceleration terms can be justified by the sampling frequency of the control system (10Hz, in this case) - given the low speeds expected from the target AGV (<1.5 m/s), acceleration over a 0.1s time-step will have a negligible effect on localisation and control efforts. Similarly, the rate of change of bias terms is considered negligible over the short sampling period and may be approximated by noise terms.

Empirical measurements are contained within the observation model, which integrates data from each of the sensors. The discrete-time representation of the observation model follows:

$$z_k = \underbrace{\begin{bmatrix} x_k + n_x \\ y_k + n_y \\ v_k + b_{encoder_k} n_v \\ \theta_k + n_\theta \\ \omega_k + b_{gyro_k} + n_\omega \end{bmatrix}}_{h(x_k, n_k)} \quad (12)$$

where $\{x_k, y_k\}$ are the most recent position measurements from the GPS (updated at 1Hz and converted into the East-North-Up coordinate frame, as mentioned earlier), $v_k + b_{encoder_k}$ is the average vehicle speed in metres per second and encoder bias measured over each time-step using the optical encoders, θ_k is the bearing from the analog compass in radians $[-\pi, \pi]$ adjusted for magnetic declination (adjusted from magnetic north to true north) and $\omega_k + b_{gyro_k}$ is the angular velocity in radians per second with bias measured from the gyroscope. n_x, n_y, n_v, n_θ and n_ω are measurement noise terms, which represent observation uncertainty. Like process noise,

they too are assumed to be zero-mean, Gaussian, white noise terms with known covariances, $p(n) \sim N(0, R)$. Measurement noise is typically derived from measurement device inaccuracies [1], which may be inherent, caused by mechanical vibrations or electromagnetic interference. Notably, the GPS measurements only update at 1Hz, whereas all other sensors update at 10Hz. The filter will account for this intrinsically – a significant advantage of Kalman Filtering over other filtering techniques.

4. Core Algorithm

EKF implementation is broken into two distinct stages: prediction and correction [6]. It is a recursive process, repeating with every time-step, k . EKFs are well suited for embedded applications because they only use information from up to one time-step in the past [1]. Consequently, computational requirements are significantly less than systems that store and use all previous state information for estimation.

All stages of the EKF implementation follow:

- i. The first predictive stage is a-priori state estimation, which is achieved using the a-posteriori (\hat{x}_{k-1}) state estimate from the previous time-step. Notice that noise (w_k) is not considered at this stage because it's exact value is unknown [6]:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, 0) \quad (13)$$

- ii. The next stage is a-priori error covariance estimation. Error covariance provides a measure of the accuracy of the state estimate :

$$P_k^- = F_k P_{k-1} F_k^T + W_k Q W_k^T \quad (14)$$

This stage requires F_k and W_k , which are Jacobian matrices (the matrix form of the first order Taylor approximation) of the process model with respect to state, and process noise. Jacobians are recalculated in every time-step to linearise the system about the current state. F_k and W_k are calculated as follows:

$$F_k = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \omega} & \frac{\partial f_1}{\partial b_{encoder}} & \frac{\partial f_1}{\partial b_{gyro}} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \omega} & \frac{\partial f_2}{\partial b_{encoder}} & \frac{\partial f_2}{\partial b_{gyro}} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial \theta} & \frac{\partial f_3}{\partial \omega} & \frac{\partial f_3}{\partial b_{encoder}} & \frac{\partial f_3}{\partial b_{gyro}} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial v} & \frac{\partial f_4}{\partial \theta} & \frac{\partial f_4}{\partial \omega} & \frac{\partial f_4}{\partial b_{encoder}} & \frac{\partial f_4}{\partial b_{gyro}} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial v} & \frac{\partial f_5}{\partial \theta} & \frac{\partial f_5}{\partial \omega} & \frac{\partial f_5}{\partial b_{encoder}} & \frac{\partial f_5}{\partial b_{gyro}} \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial v} & \frac{\partial f_6}{\partial \theta} & \frac{\partial f_6}{\partial \omega} & \frac{\partial f_6}{\partial b_{encoder}} & \frac{\partial f_6}{\partial b_{gyro}} \\ \frac{\partial f_7}{\partial x} & \frac{\partial f_7}{\partial y} & \frac{\partial f_7}{\partial v} & \frac{\partial f_7}{\partial \theta} & \frac{\partial f_7}{\partial \omega} & \frac{\partial f_7}{\partial b_{encoder}} & \frac{\partial f_7}{\partial b_{gyro}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t_k \sin \theta_k & \Delta t_k v_k \cos \theta_k & 0 & 0 & 0 \\ 0 & 1 & \Delta t_k \cos \theta_k & -\Delta t_k v_k \sin \theta_k & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t_k & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

$$W_k = \begin{bmatrix} \frac{\partial f_1}{\partial w_x} & \frac{\partial f_1}{\partial w_y} & \frac{\partial f_1}{\partial w_v} & \frac{\partial f_1}{\partial w_\theta} & \frac{\partial f_1}{\partial w_\omega} & \frac{\partial f_1}{\partial w_{b_{encoder}}} & \frac{\partial f_1}{\partial w_{b_{gyro}}} \\ \frac{\partial f_2}{\partial w_x} & \frac{\partial f_2}{\partial w_y} & \frac{\partial f_2}{\partial w_v} & \frac{\partial f_2}{\partial w_\theta} & \frac{\partial f_2}{\partial w_\omega} & \frac{\partial f_2}{\partial w_{b_{encoder}}} & \frac{\partial f_2}{\partial w_{b_{gyro}}} \\ \frac{\partial f_3}{\partial w_x} & \frac{\partial f_3}{\partial w_y} & \frac{\partial f_3}{\partial w_v} & \frac{\partial f_3}{\partial w_\theta} & \frac{\partial f_3}{\partial w_\omega} & \frac{\partial f_3}{\partial w_{b_{encoder}}} & \frac{\partial f_3}{\partial w_{b_{gyro}}} \\ \frac{\partial f_4}{\partial w_x} & \frac{\partial f_4}{\partial w_y} & \frac{\partial f_4}{\partial w_v} & \frac{\partial f_4}{\partial w_\theta} & \frac{\partial f_4}{\partial w_\omega} & \frac{\partial f_4}{\partial w_{b_{encoder}}} & \frac{\partial f_4}{\partial w_{b_{gyro}}} \\ \frac{\partial f_5}{\partial w_x} & \frac{\partial f_5}{\partial w_y} & \frac{\partial f_5}{\partial w_v} & \frac{\partial f_5}{\partial w_\theta} & \frac{\partial f_5}{\partial w_\omega} & \frac{\partial f_5}{\partial w_{b_{encoder}}} & \frac{\partial f_5}{\partial w_{b_{gyro}}} \\ \frac{\partial f_6}{\partial w_x} & \frac{\partial f_6}{\partial w_y} & \frac{\partial f_6}{\partial w_v} & \frac{\partial f_6}{\partial w_\theta} & \frac{\partial f_6}{\partial w_\omega} & \frac{\partial f_6}{\partial w_{b_{encoder}}} & \frac{\partial f_6}{\partial w_{b_{gyro}}} \\ \frac{\partial f_7}{\partial w_x} & \frac{\partial f_7}{\partial w_y} & \frac{\partial f_7}{\partial w_v} & \frac{\partial f_7}{\partial w_\theta} & \frac{\partial f_7}{\partial w_\omega} & \frac{\partial f_7}{\partial w_{b_{encoder}}} & \frac{\partial f_7}{\partial w_{b_{gyro}}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

The effects of process noise are considered through Q , which is the process noise covariance matrix. It provides statistical knowledge of process noise to the EKF. Later, this knowledge is used in conjunction with the measurement noise covariance to weight the impact of model based predictions and empirical measurements on state estimation. For the AGV, values in the Q matrix are fixed. In fact, process noise can be very difficult to characterise, so matrix entries usually require tuning with simulation and empirical testing. The Q matrix is often diagonal. The first element on the diagonal relates to the noise affecting equation f_1 . Other diagonal elements relate to the noise affecting f_2, f_3, f_4, f_5 and f_6 , respectively. Notably, this matrix must be positive semi-definite with all eigenvalues ≥ 0 .

$$Q = \begin{bmatrix} 0.00001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.00001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The first corrector stage of the EKF process is empirical measurement. This stage uses the observation model developed earlier. Again, notice that noise (n_k) is not considered here, because, as in the process model, its exact value is unknown:

$$z_k = h(x_k, 0) \quad (17)$$

The next stage is known as innovation. Innovation is a measure of the difference between empirical state measurements and predicted states, which are derived using the H_k Jacobian matrix and the a-priori state estimates \hat{x}_k^- as follows:

$$\tilde{y}_k = z_k - H_k \hat{x}_k^- \quad (18)$$

H_k is the Jacobian matrix of the observation model with respect to state:

$$H_k = \begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial y} & \frac{\partial h_1}{\partial v} & \frac{\partial h_1}{\partial \theta} & \frac{\partial h_1}{\partial \omega} & \frac{\partial h_1}{\partial b_{encoder}} & \frac{\partial h_1}{\partial b_{gyro}} \\ \frac{\partial h_2}{\partial x} & \frac{\partial h_2}{\partial y} & \frac{\partial h_2}{\partial v} & \frac{\partial h_2}{\partial \theta} & \frac{\partial h_2}{\partial \omega} & \frac{\partial h_2}{\partial b_{encoder}} & \frac{\partial h_2}{\partial b_{gyro}} \\ \frac{\partial h_3}{\partial x} & \frac{\partial h_3}{\partial y} & \frac{\partial h_3}{\partial v} & \frac{\partial h_3}{\partial \theta} & \frac{\partial h_3}{\partial \omega} & \frac{\partial h_3}{\partial b_{encoder}} & \frac{\partial h_3}{\partial b_{gyro}} \\ \frac{\partial h_4}{\partial x} & \frac{\partial h_4}{\partial y} & \frac{\partial h_4}{\partial v} & \frac{\partial h_4}{\partial \theta} & \frac{\partial h_4}{\partial \omega} & \frac{\partial h_4}{\partial b_{encoder}} & \frac{\partial h_4}{\partial b_{gyro}} \\ \frac{\partial h_5}{\partial x} & \frac{\partial h_5}{\partial y} & \frac{\partial h_5}{\partial v} & \frac{\partial h_5}{\partial \theta} & \frac{\partial h_5}{\partial \omega} & \frac{\partial h_5}{\partial b_{encoder}} & \frac{\partial h_5}{\partial b_{gyro}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (19)$$

The next stage is to calculate the innovation covariance:

$$S_k = H P_k^- H^T + N_k R N_k^T \quad (20)$$

This stage requires H_k , as derived above, and N_k , which is the Jacobian matrix of the observation model with respect to measurement noise:

$$N_k = \begin{bmatrix} \frac{\partial h_1}{\partial n_x} & \frac{\partial h_1}{\partial n_y} & \frac{\partial h_1}{\partial n_v} & \frac{\partial h_1}{\partial n_\theta} & \frac{\partial h_1}{\partial n_\omega} \\ \frac{\partial h_2}{\partial n_x} & \frac{\partial h_2}{\partial n_y} & \frac{\partial h_2}{\partial n_v} & \frac{\partial h_2}{\partial n_\theta} & \frac{\partial h_2}{\partial n_\omega} \\ \frac{\partial h_3}{\partial n_x} & \frac{\partial h_3}{\partial n_y} & \frac{\partial h_3}{\partial n_v} & \frac{\partial h_3}{\partial n_\theta} & \frac{\partial h_3}{\partial n_\omega} \\ \frac{\partial h_4}{\partial n_x} & \frac{\partial h_4}{\partial n_y} & \frac{\partial h_4}{\partial n_v} & \frac{\partial h_4}{\partial n_\theta} & \frac{\partial h_4}{\partial n_\omega} \\ \frac{\partial h_5}{\partial n_x} & \frac{\partial h_5}{\partial n_y} & \frac{\partial h_5}{\partial n_v} & \frac{\partial h_5}{\partial n_\theta} & \frac{\partial h_5}{\partial n_\omega} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

The effects of measurement noise are considered through R , which is the measurement noise covariance matrix. It provides statistical knowledge of the measurement noise to the EKF. Like the Q matrix, the values in this matrix are fixed.

Here, they are set to represent sensor variances, calculated during sensor characterisation. Again, this matrix is often diagonal. The first element on the diagonal represents noise affecting the sensor related to equation h_1 . Other diagonal elements represent noise affecting the sensors relating to h_2 , h_3 , h_4 , h_5 and h_6 , respectively. Notably, this matrix must be positive-definite with eigenvalues > 0 . Manipulating the Q and R matrices affects filter performance. They must be tuned concurrently, to affect response speed and filtering ability for each specific application. Larger values in the R matrix compared to the Q matrix, filter more measurement noise, but slow the state reconstruction (speed of the filter), and vice versa [7].

$$R = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0.00001 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.005 \end{bmatrix}$$

The next stage adjusts the Kalman gain, which provides the weighting effect described earlier. More explicitly, it is used to update the a-priori estimate based on the value of empirical measurements and the statistical significance of the noise affecting them, with respect to the significance of the noise affecting the process as follows:

$$K_k = P_k^- H S_k^{-1} \quad (22)$$

Next, an a-posteriori state estimate is obtained using the Kalman gain:

$$\hat{x}_k = \hat{x}_k^- + K_k \tilde{y}_k \quad (23)$$

Finally, the error covariance is updated with an a-posteriori estimate:

$$P_k = (I - K_k H) P_k^- \quad (24)$$

The a-posteriori state estimates are the useful filter outputs. Error covariance may

be used to track filter performance. All subsequent AGV control efforts are executed using the a-posteriori state estimates.

5. Results and Discussion

The EKF was tested and tuned using MATLAB. The graphical output from MATLAB facilitated visualisation of the filter's effect on real sensor data collected from the AGV during the characterisation phase, as well as its effect on simulated sensor data for different operating conditions. For the AGV example application, the EKF was later translated into C code so that it could be programmed into the AGV's onboard microcontroller as a part of the overall control system.

Figure 3 below shows the effect of the filter on positional data (in the local East-North-Up Cartesian coordinate frame) for a circular path. Raw sensor data for this graph was simulated using MATLAB. All simulated sensor data is wholly based on characterized sensor performance of the subject vehicle, with additional random noise.

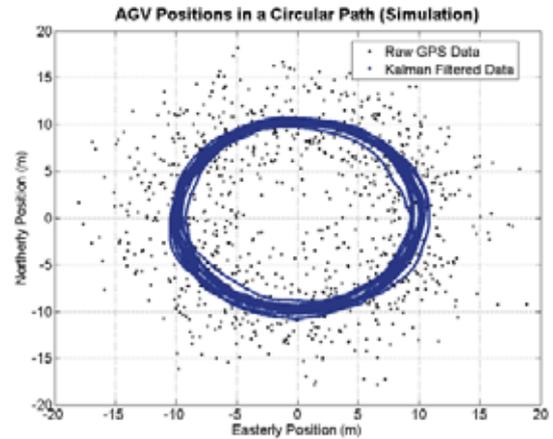


Figure 3. AGV Positions in a Circular Path

Figure 4 below shows the effects of the filter on 1000 samples of raw sensor data

collected (at 10Hz) during a characterisation test of the stationary AGV. Here, the filter can be seen to reduce deviations from up to 4m in either direction to less than 1 metre. GPS drift is evident in the raw data. Sensor noise is very rarely absolutely random. Therefore it is important to evaluate the effectiveness of the filter on real data as well as on simulated data.

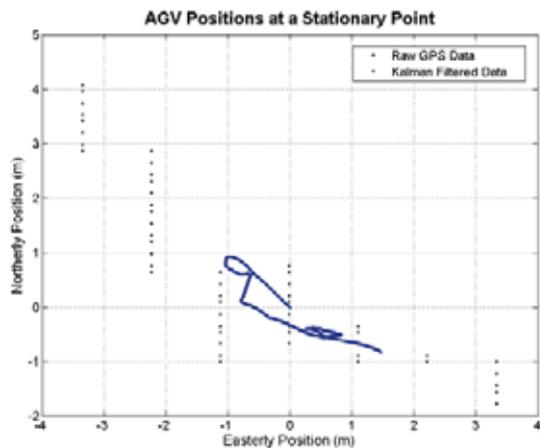


Figure 4. AGV Positions at a Stationary Point

Figure 5 below shows the effects of the filter on the gyroscope readings collected from the stationary AGV. Here, the filtering effect is noticeable because the filter was configured to place a strong emphasis on the process model, thus eliminating the effects of noise in the raw sensor data. Similar graphs of the other sensors illustrate comparable filtering performance.

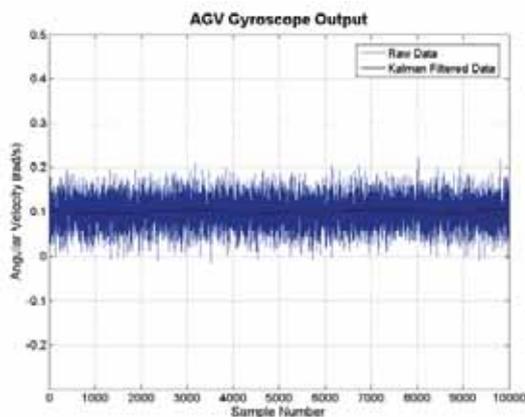


Figure 5. AGV Gyroscope Output at a Stationary Point

The proposed filter is a specific example of a discrete-time EKF. This formulation can be easily manipulated for other applications. EKFs are commonly seen in aircraft stabilisation systems, and they are especially useful in situations where low-cost and noise-prone sensor combinations are used. Obviously, sensor redundancy can be easily incorporated into a system using an EKF, allowing for the event of sensor failure in certain applications. As has been mentioned, localisation does not have to be constrained to 2D, as Kalman Filters can be used in 1D, 2D and 3D applications.

EKFs have a strong dependence on the accuracy of initial conditions. Because a linearisation scheme is used, any large discrepancies at the outset will make the filter deviate from the expected value for many iterations (time-steps).

6. Conclusions

The EKF is an extremely powerful tool for state estimation, concurrently accepting the input signals from multiple sensors in a process known as sensor fusion, whilst filtering signals to give best estimates of state. This article has shown a specific application of an EKF to AGV localisation, however the filter has much more widespread application potential. All Kalman Filters require careful tuning. EKFs, in particular, require thorough consideration of initial conditions, due to the effects of nonlinearities and subsequent linearisation schemes.

Acknowledgements

Many thanks go to K.C. Aw and K. Stol at The University of Auckland for their guidance and assistance in understanding the concepts of Kalman Filtering and for

pointing out several intricacies, which are covered in this paper.

References

1. Maybeck, P.S., *Stochastic models, estimation, and control*. 1979, Air Force Institute of Technology, Wright-Patterson Air Force Base: Ohio.
2. Jackson, J.P., and Donnelly, J. *Final Year Project Report*, 2010.
3. Kalman, R.E., *A New Approach to Linear Filtering and Prediction Problems*. Transactions of the ASME - Journal of Basic Engineering, 1960. **82**: p. 35-45.
4. Mao, G., S.P. Drake, and B.D.O. Anderson, *Design of an Extended Kalman Filter for UAV Localization*. 2007, School of Electrical and Information Engineering - the University of Sydney, Research School of Information Sciences and Engineering - Australian National University, Defence Science and Technology Organization (DSTO), National ICT Australia Limited: Sydney, Canberra.
5. LaValle, S.M. *Planning Algorithms - A Simple Car*. 2006 24 April 2010 [cited 2010 10 September]; Available from: <http://planning.cs.uiuc.edu/node658.html>.
6. Green, W.E., *Simultaneous Localization and Mapping - Lecture 02: Extended Kalman Filter*. 2006, Drexel University: Philadelphia, PA.
7. Stol, K., *MECHENG 423: Digital and Computer Control Lecture Material*. 2010, The University of Auckland: Auckland. p. 6-6.

Localisation and Path Tracking in a Steered-wheel Autonomous Vehicle

J.P Jackson, J.E. Donnelly
The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract

Autonomous Ground Vehicles are becoming increasingly common in industry today. Their prevalence is partly due to the vast range of applications that have a need for an accurate and robust method for automatically guiding a vehicle along a predetermined path. Their rise to prominence in a diverse range of applications has created a need for accurate, environmentally insensitive path tracking techniques. They could potentially revolutionise the transportation industry. The primary objective of this project was to improve vehicle navigation by developing a more reliable path tracking algorithm and enhancing pre-existing sensor filtering techniques in order to change the balance of vehicle-sensor dependency thus reducing noise-induced path deviations. A complete mechanical rebuild provided a dependable foundation for the system. The construction of a computationally efficient and effective central processing unit was achieved by utilisation of asynchronous exception handling and logical and well-structured software. Finally, the implementation of an extended Kalman filter and incorporation of GPS and sensor fusion helped to assist in the localization and path tracking of the AGV system.

Keywords: *AGV, GPS, Localisation, Path tracking*

1. Introduction

Autonomous Ground Vehicles (AGVs) are vehicles which are designed to automatically navigate along a determined path between pre-programmed waypoints without any external human intervention or assistance. During the last couple years the development of such vehicles has progressed significantly. The concept of autonomous vehicles is becoming increasingly common, leading to the adoption of such systems in a diverse range of applications. The scope of their prospective use is seemingly unbounded, from unmanned military vehicles [1] to automatic lawn mowers [2]. They have even

found a place in industrial factory environments like robotic forklifts [3]. This project is focused primarily on the incorporation of GPS and sensor fusion to assist in the localization and path tracking of the AGV system. Since 2005, students from the Mechatronics engineering department at the University of Auckland has been designing and refining a GPS guided path tracking autonomous vehicle (Figure 1). This project is a continuation of their work, more specifically this project extends from the most recent improvement to the AGV, completed by C.C. Wei and L.H.A. Lee in 2009 [4, 5].



Figure 1. A picture of the AVG

The introduction of two optical encoders on the rear axles to measure linear velocity initiated the transformation of the AGV system from a straight inertial navigation system to an inertial-odometric navigational system. Including these encoders there are four on-board sensor units actively involved in the AGV system. A GPS receiver is used to set the waypoints of the desired path and to detect significant deviations from the desired path. An electronic compass is used to obtain the cars bearing. Finally, an Inertial Measurement Unit (IMU) containing a gyroscope and accelerometers is used to obtain the cars bearing while it is moving. The addition of the optical encoders rendered the accelerometers redundant. It would seem that the IMU would play a less important role than in previous years however it remains an integral part of the AGV system. This is because the gyro is vital in acquiring the cars angular velocity and bearing when mobile.

2. Sensors

The AGV system comprises of 4 sensor units, namely the Inertial Measurement Unit (IMU), compass, GPS and optical encoders. This makes up a total of seven individual sensors. The IMU contains a gyroscope and dual axis accelerometer package containing

x and y accelerometers. The introduction of optical encoders this year on each wheel allowed a direct measurement of velocity to be obtained which ultimately rendered the accelerometers redundant. They were therefore not used in the AGV system and were left out of the extended Kalman filter. A new higher quality Garmin GPS receiver was also introduced this year which allowed significantly more accurate measurements of the cars global position. The analogue compass used in previous years has also been retained. This was decided after testing proved its output to be reasonably accurate and reliable.

3. Optical Encoders

In order to acquire a more accurate representation of the cars velocity, two new optical encoders were added to either side of the rear axle, to measure the angular velocity of each of the back wheels. The devices chosen were the Agilent AEDB-9140-A13 modules [6]. See Figure 2.



Figure 2. An optical encoder

Two were necessary to allow for the fact that whilst turning the outer wheels of the car move faster than the inside wheels and so the average of the two encoder outputs is taken as the vehicle velocity. The optical encoders output three channels. Two of the channels are dedicated to the high resolution measurement with 500 CPR (counts per revolution) whilst the third channel provides one count per revolution. They operate on a

5V supply with high and low level thresholds of 2.4 V and 0.4 V respectively.

The transmitter arm consists of an LED, which serves to produce a constant light source. This light is emitted across the sensing gap (in which the code-wheel is located) to the receiver arm. If not blocked by the code-wheel this light is detected by an optodetector array consisting of photodiodes. The code-wheel uses a series of windows and bars in order to consistently interrupt the light signal as it rotates. The result is an alternating detector output of square waves.

The frequency of these waves is directly proportional to the rate at which the light signal is interrupted, and thus represents the rate at which the code-wheel rotates. With the appropriate signal conditioning, hardware logic and service routine this representation of the rotary motion of the shaft can be converted into a velocity of the car.

The first stage of signal processing is performed by quadrature decoders [7]. These require two signals with 90° phase difference between them, as is the case with Channel A and B from the optical encoders shown in the first two waves displayed in Figure 3.

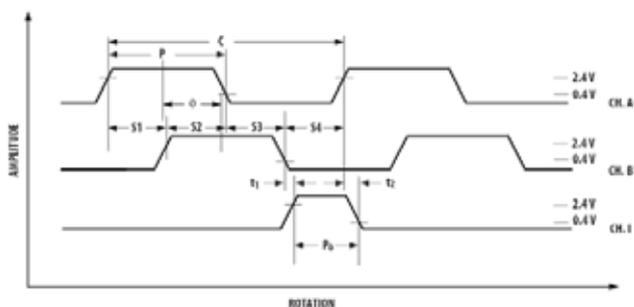


Figure 3. Outputs from the encoder.

The 3rd index channel, Channel I, is not used in the AGV system. By comparing the levels and edges of the two input signals, the

quadrature encoders can determine the counts travelled and also the direction. For the chosen optical encoders, during clockwise rotation of the code-wheel (from the power pin, V, to the ground pin, G) channel A will lead channel B as shown in Figure 8 and vice versa. In the AGV system the left encoder is mounted so that clockwise rotation of the code-wheel corresponds to forward motion whilst the right encoder is the opposite. Since the car will only be travelling forwards, the direction output of the quadrature decoder is somewhat redundant in the AGV system and it has thus been omitted, however if it need be it can easily be re implemented by a few additional logic blocks. The quadrature decoder implemented in the AGV system operates in 4x mode, meaning it counts all the transitions of the quadrature inputs. Therefore it count enables for every rising and falling edge of both channel A and channel B, thus incrementing the counter 4 times for every window in the code-wheel. The additional two D flip flops that can be seen on each input in the hardware block ensure that no metastability is introduced.

In order to eliminate any clock-induced errors from the quadrature decoders, an oversampling clock signal must be used, that is a clock signal that is at least three times faster than maximum possible quadrature signals. As will be proven below, the fastest realistic frequency of data output for the optical encoders is approximately 25kHz. Therefore the internal 50MHz crystal oscillator clock on the DE2 Board is more than reaches the specifications required for these hardware blocks.

The next stage of the signal processing involves the count enable output from the quadrature decoders being passed into a 12-bit unidirectional incremental counter. The directionality of these counters is imperative; it must only be allowed to count

up. This is because realistically the AGV system will only ever be moving forwards, and if the car experiences some vibration while stationary, the high sensitivity of the encoders means that a small backwards rotation may be sensed, which would cause a bidirectional counter to roll back to its highest possible value, causing the system to falsely believe it is travelling at a very high speed. The counter has an asynchronous reset which clears the counter every sampling period, 0.1s for the 10Hz timer interrupt implemented in the software. Before implementing the unidirectional counter, calculations were made to determine the maximum frequency of counts and hence the amount of bits required to ensure the counter never overflows. For these calculations the max speed was assumed to be $V_{max} \approx 5 \text{ ms}^{-1}$. The diameter of the wheel is $D = 128 \text{ mm}$ and the resolution of the code-wheel is given in terms of counts per revolution, $CPR = N = 500$.

Angular velocity:

$$\omega = \frac{V}{R} = \frac{5}{0.5 \times 0.128} = 78.125 \text{ rad. s}^{-1}$$

$$\therefore V_{rps} = \frac{78.125}{2\pi} = 12.44 \text{ rps}$$

Therefore the max frequency is given by:

$$f_{max} = V_{rps} \times N \times 4 = 12.44 \times 500 \times 4 = 24,868 \text{ Hz} \approx 25 \text{ kHz}$$

The system uses an UART interrupt time interval of 0.1s (10Hz), therefore max required counts of counter per time interval is:

$$n = \frac{24,867 \text{ Hz}}{10 \text{ Hz}} = 2486.7$$

Number of bits required:

$$2^b = 2486.7 \Rightarrow b = \frac{\log(2486.7)}{\log(2)} = 11.28$$

Therefore 12 bits are required. So a 12-bit incremental counter was implemented.

The final stage of the signal processing is the encoder subroutine found C program in the NIOS® II processor. This subroutine reads the count input from the incremental counter for both wheels every sampling period, $\Delta T = 0.1\text{s}$.

Let C_L = count from left wheel and C_R = count from right wheel.

$$d_L = \frac{C_L}{N} \times 2\pi r = \frac{C_L}{N} \times \pi D, \quad d_R = \frac{C_R}{N} \times \pi D$$

Velocities in rps, with

$$\Delta t = \text{fixed time interval} = 0.1\text{s}:$$

$$V_L = \frac{1}{N} \frac{C_L}{4 \times \Delta t}, \quad V_R = \frac{1}{N} \frac{C_R}{4 \times \Delta t}$$

The four in the denominator represents the fact that the quadrature encoders operate in the 4x mode, meaning they count enable every rising and falling edge of both channel A and channel B.

Acceleration in revolutions per second squared (rps^2) using a backwards difference approximation:

$$a_L = \frac{1}{N} \frac{C_{L,n} - C_{L,n-1}}{(4 \times \Delta t)^2}, \quad a_R = \frac{1}{N} \frac{C_{R,n} - C_{R,n-1}}{(4 \times \Delta t)^2}$$

We get the final distance, velocity and acceleration values by averaging over the two wheels:

$$d = \frac{d_L + d_R}{2}, \quad V_{rpm} = \frac{V_L + V_R}{2}, \quad a_{rpm} = \frac{a_L + a_R}{2}$$

We can then find the velocity and acceleration values in the more conventional SI units, i.e. ms^{-1} and ms^{-2} respectively:

$$V_{ms^{-1}} = V_{rps} \times \pi D, \quad a_{ms^{-2}} = a_{rpm} \times \pi D$$

In order to test the stability of the encoders, testing was performed at motor signals of :

PWM =
 {10% 20% 30% 40% 50%}

These were undertaken whilst “freewheeling”, meaning that the wheels were never in contact with the ground. This allowed greater wheel speeds to be reached and thus the encoders could be tested more rigorously. Any PWM motor signals over 0.5 saturate the battery, which is why the max motor signal was set at this value. The graph produced from this testing is shown in Figure 4.

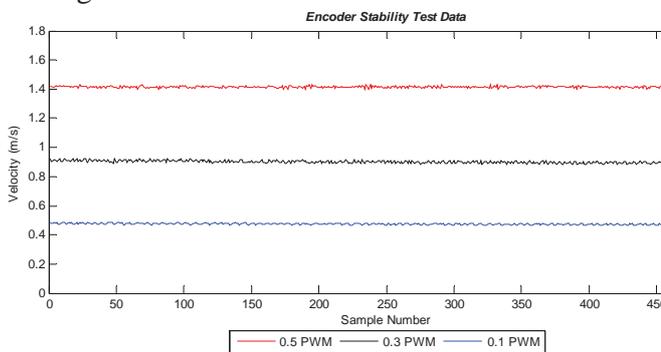


Figure 4. PWM versus AVG velocity

This shows that with increasing speed, the noise and fluctuation of the encoders increase, but never any more than $\pm 0.1 \text{ m/s}$, an acceptable deviation

4. Global Positioning System (GPS)

In previous years the Navman 1240 GPS receiver was used. This is a fairly old model that was designed specifically for nautical applications and as such was not very accurate or suitable for the AGV system. It was therefore necessary to upgrade to a newer GPS receiver, the Garmin GPS 18X LVC [8]. This is a high sensitivity, low power receiver that produces pulse-per-second logic level outputs in NMEA 0183format (industry standard). It was designed for use in automotive, fleet vehicle and electronics applications where a small, highly accurate GPS is needed and is

therefore ideal for the car. It runs on 5V supply drawing 90mA of current. Compare this with the 12V supply of the previous receiver. With a non-differential fix, the GPS Standard Positioning System (SPS), it induces errors less than 15m, and has an update rate of 1Hz, spitting out GPS data every second. Its rugged, waterproof design allows the receiver to withstand continuous exposure to the most prevailing weather conditions and its high sensitivity means it can even be used under light cover. See Figure 5.



Figure 5. The position of the GPS on the AVG

In order to test the accuracy and investigate the drift of the GPS receiver, over 15 minutes of GPS data was logged and transformed into the East North Up coordinate frame. The variations are shown in the histogram graphs in shown in Figure 6.

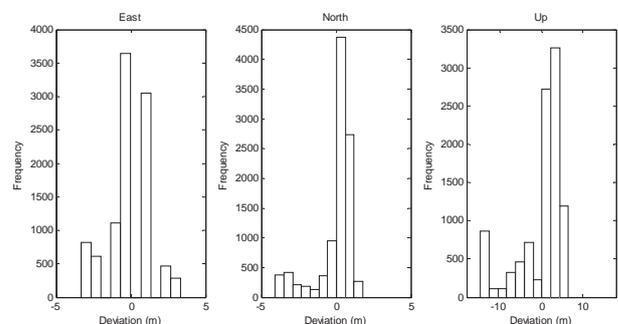


Figure 6. Variation in the GPS

These show that for all directions the deviations are approximately normally distributed. Over the first 5 minutes the GPS experiences the greatest drift. It then plateaus with minor deviation from the mean value. There are numerous explanations for this inherent drift found in GPS receivers. One of the more significant is the atmospheric and ionospheric delays [9]. As the waves travel through different mediums in the earth's atmosphere they slow down or speed up causing errors in the measurements of the distance from the receiver to the satellites. Another common source of the errors is caused by Doppler Effect. This occurs when the relative speeds of the GPS and satellite cause the radio signals to experience the Doppler shift phenomenon which causes the signal to bunch up or spread out, thus introducing error into the processing of the signal due to timing discrepancies.

5. Power Flow

The power flow in the AGV system is best described by the flow diagram shown below. Figure 7 shows the main power distribution and the voltage outputs to all components of the system.

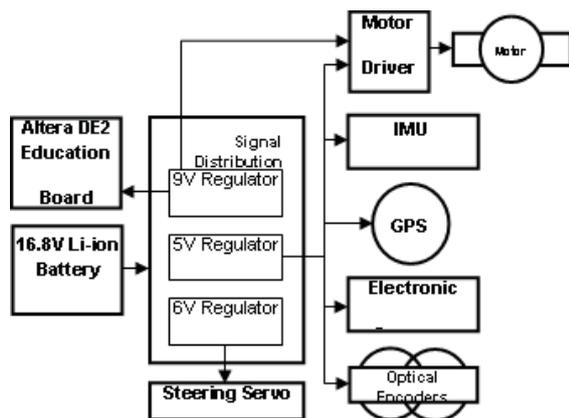


Figure 7. Power distribution in the AGV

The power distribution board used in previous years was inefficient and had

developed some leakage problems over the years due to the poor quality of the PCB. A number of quick soldering fixes had been undertaken to keep it functional. This was obviously not ideal and in addition to this, almost all the power dissipated from the board was transferred as heat through one voltage regulator whose power rating was exceeded. This resulted in an extremely hot heat sink. It was therefore necessary to produce a new more suitable power distribution PCB that could meet the voltage regulation and supply requirements in a more efficient and effective manner. A switching regulator was thus presented as a desirable alternative to the multiple linear voltage regulators used previously. The new power distribution PCB is shown in Figure 8.



Figure 8. The new power distribution board

Switching regulators spend most of their time in saturation or cut-off modes, both of which dissipate very little power, and as such have a much higher efficiency than linear regulators which operate within their active regions [10]. These favourable power characteristics also mean that a switching regulator can drive an output power much greater than its own maximum power rating. Switching regulators also produce significant noise, due to the large inductive components. A large capacitor is used as a low pass filter to help to reduce noise levels. Strategic placement of components and careful consideration of the proximity noise sensitive circuits, the signal circuitry, to the

power distribution board was made in order to preserve signal integrity.

A National Semiconductor switching regulator, the LM2576HV Series Simple Switcher® [11], was used to provide the regulation requirements of the AGV system. It is a simple, high efficiency step down (buck) voltage regulator stored in a single DIP package. It is implemented in the AGV system in the adjustable voltage layout as shown in Figure 9.

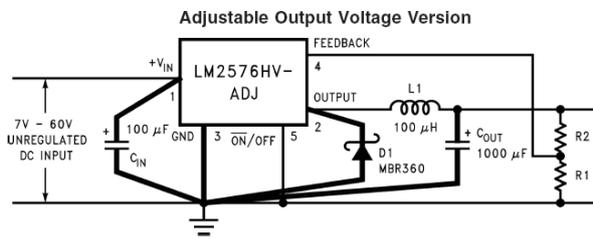


Figure 9. Voltage regulator

The output voltage is given by:

$$V_{out} = V_{ref} \left(1 + \frac{R_2}{R_1} \right) \quad \therefore \quad R_2 = R_1 \left(\frac{V_{out}}{V_{ref}} - 1 \right)$$

6. Path Tracking

The “Follow the Carrot” path tracking algorithm uses the concept of a Look Ahead Point (LAP), or the “carrot” to guide the car to each waypoint. The LAP is situated a fixed distance along the path, the straight line connecting the two active waypoints, i.e. the waypoint being navigated from and the waypoint been navigated to. This constant distance is termed the Look Ahead Distance (LAD). The sequence of events in the path tracking subroutine is as follows:

1. Read in the filtered position and bearing data from the extended Kalman filter
2. Calculate the position of the Look Ahead Point (derivation below)

3. Execute the “Follow the Carrot” algorithm which produces the required turning radius $R_{turning}$ (description below)
4. Use this turning radius value to calculate the required steer angle and thus output the corresponding PWM signal to the servo (description below).

Look Ahead Point

For each iteration of the follow the carrot algorithm, a new value of the look ahead point must be obtained. The full derivation of this point can be found in J.P. Jackson’s or J.E. Donnelly’s project reports [12, 13]. A shortened version will be presented here. The vector equation of the path between the current waypoint (\underline{A}_n) and the previous waypoint (\underline{A}_{n-1}) is given by:

$$\underline{P} = \underline{A}_{n-1} + u(\underline{A}_n - \underline{A}_{n-1})$$

An orthogonal projection of $\overrightarrow{\underline{A}_{n-1}\underline{V}}$ on $\overrightarrow{\underline{A}_{n-1}\underline{A}_n}$ gives the distance along WP path or vector component of AGV position along path. The total length L from the previous waypoint to the new Look Ahead Point (LAP) is an addition of D and LAD , the fixed look ahead distance, defined by the user:

$$L = D + LAD$$

Non-dimensionalisation using the path length gives the proportional distance along the path:

Follow the carrot algorithm

The following derivation is based on the algorithm developed by S.K Manley and J.Z. Wong in 2005 [14, 15]. The first step is to calculate the heading error between the vehicles line of motion and the line between the vehicles reference point (middle of rear axle) and the LAP. This is shown in the

diagram in the right column. This is given by:

$${}^V\theta_L = \theta_{V-L} - \theta_V$$

where,

$$\theta_{V-L} = \arctan\left(\frac{X_L - X_V}{Y_L - Y_V}\right) \quad -\pi < \theta_{V-L} < \pi$$

This works with the convention that a right hand/clockwise turn is defined as positive with a left hand/counter-clockwise turn defined as negative.

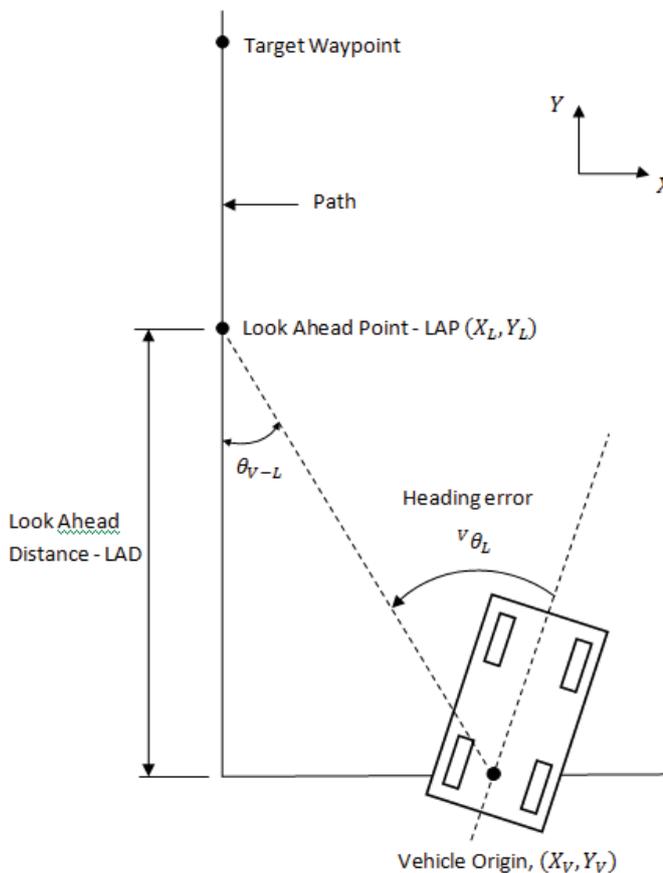


Figure 10. Follow the carot algorithm

The algorithm outputs a required turning radius. This is inversely proportional to the heading error as an increase in the heading requires a sharper turn and hence a smaller turning radius. The turning radius is given by:

$$R_{turning} = \frac{1}{k \times {}^V\theta_L}$$

$$R_{min} \leq R_{turning}$$

where k is the proportional path-tracking gain and R_{min} is the minimum turning radius allowed, limited by the hardware. Note ${}^V\theta_L$ must not be passed into this equation when it is equal to zero in order to avoid math errors. This corresponds to the car being exactly on path between waypoints in the correct orientation. Therefore if ${}^V\theta_L = 0$, $R_{turning}$ must be set to a very large number representing a turning radius equal to infinity, which is a straight line.

Using the models derived in previous sections the steer angle and corresponding servo PWM signal and given by:

$$\theta_{steer} = \arctan\left(\frac{Wheelbase}{R_{turning}}\right)$$

$$PWM_{servo} = \frac{2.232 - \theta_{steer}}{32.8815}$$

The motor PWM signal is dependent on the proportional distance along path, u . When $u \leq 0.1$ or $u \geq 0.9$, i.e. within 10% of a waypoint, the PWM output to the motor is 30%. For all other values of u , the PWM output to the motor is 50%.

7. Extended Kalman Filter

The AGV system uses an Extended Kalman filter to provide the localisation required for path tracking. A Kalman filter is simply an optimal recursive data processing algorithm [13] that provides a non-deterministic (stochastic) model of the system and uses optimal state estimates in order to control the system as desired. The filter incorporates sensor fusion between the outputs the GPS, optical encoders, analog compass and the gyroscope as it represents the dynamic model of the AGV with 6 state variables.

These can be expressed by the continuous-time equations:

$$\begin{aligned}\dot{x}(t) &= v(t) \sin \theta(t) + w_x(t) \\ \dot{y}(t) &= v(t) \cos \theta(t) + w_y(t) \\ \dot{v}(t) &= w_v(t) \\ \dot{\theta}(t) &= \omega_\theta(t) \\ \dot{\omega}(t) &= w_\omega(t) \\ \dot{b}_{gyro}(t) &= w_{b_{gyro}}(t)\end{aligned}$$

The Kalman filter uses the system and sensor dynamics and the statistical description of the sensor noises (the variances) in order to predict the current state of the system, the appropriate components of this state can then be inputted into the path tracking algorithm. A detailed explanation of the construction, application and implementation of an Extended Kalman filter implemented in the AGV system can be found in a project report by J.E. Donnelly's [13].

Acknowledgements

The authors would like to thank Dr Kean Aw, for the opportunity to work on this project and for his continued guidance, assistance and technical expertise that was essential to the successful completion of this project. Thanks also goes to Logan Stuart, the Mechatronics lab technician, whose guidance and assistance on a daily basis helped to keep the project on track and helped give invaluable knowledge into the practical aspects of the project.

Finally, the authors would also like to thank Karl Stol, whose knowledge and insight into the implementation of a Kalman filter was much appreciated.

References

1. *Military Robots / Unmanned Ground Vehicles (UGV).*" Retrieved June 26,

- 2010, from <http://www.globalsecurity.org/military/systems/ground/ugv.htm>
2. *Robomow®.* Retrieved June 26, 2010, from <http://www.robomow.com/>
3. *Automated / Automatic Guided Vehicles Systems (AGVs).*" Retrieved June 28, 2010, from <http://jbtc-agv.com/>
4. Wei, C.C. (2009). Improvement to Sensor Accuracy and Path Tracking in Autonomous Ground Vehicle. Department of Mechanical Engineering. Part IV project report, 2009-MT23.
5. Lee, L.H.A. (2009). Improvement to Sensor Accuracy and Path Tracking in Autonomous Ground Vehicle. Department of Mechanical Engineering. Part IV project report, 2009-MT23.
6. AEDB-9140 Series Three Channel Optical Incremental Encoder Datasheet (2010). Avago Technologies.
7. *Quadrature Decoder.* Retrieved September 13, 2010, from <http://www.fpga4fun.com/QuadratureDecoder.html>
8. GPS 18x Technical Specifications (2008). Garmin International Inc., Olathe, KS.
9. Hurn, J. (1989). *GPS – A Guide to the Next Utility.* Trimble Navigation Ltd, Sunnyvale, CA.
10. Paynter, R. T. (2006). *Introductory Electronic Devices and Circuits.* New Jersey, Pearson Prentice Hall.
11. LM2576/LM2576HV Series SIMPLE SWITCHER® 3A Step-Down Voltage Regulator Datasheet (2004). National Semiconductor.
12. Jackson, J.P. (2010). Localisation and Path Tracking in a Steered-Wheel Autonomous Ground Vehicle. Part IV project report, 2010-MT10
13. Donnelly, J.E. (2010). Localisation and Path Tracking in a Steered-Wheel Autonomous Ground Vehicle. Part IV project report, 2010-MT10

14. Manley, S. K. (2007). GPS-Guided Autonomous Ground Vehicle with Improved Localisation and Path Tracking. Part IV Project Report, 2007-MT-24.
15. Wong, J. Z. S. (2007). GPS Guided Autonomous Ground Vehicle with Improved Localisation and Path-tracking, Part IV Project Report, 2007-MT-44.

Low-cost Interactive Patient Rehabilitation Monitoring

T. Buranasetakul, N. Chang

The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract

There is a growing market for interactive patient monitoring with regards to physical rehabilitation. Current rehabilitation monitoring devices do allow patient monitoring albeit with either large cumbersome or unspecialized devices that are limb specialised. In contrast, our system operates wirelessly and is flexible for the monitoring of any limb, combined with the comprehensive history of individual patients and associated profiles, this allows the supervising physiotherapist to monitor and make adjustments to the exercise regimes based on the patient recovery potential. The system is not only beneficial to the therapists but also the patients undergoing the recovery program, as the system is integrated with a virtual simulation on the computer in real-time to motivate patients to exercise longer and reduce recovery time.

Keywords: *Wireless; Medical; Therapy; Health Monitoring; Low-cost*

1. Introduction

There are many medical conditions present that require individuals to undergo general and/or specific rehabilitation regime(s) as prescribed by physiotherapists (PTs) that could involve exercises of targeted extremities for conditions that the individuals exhibit such as stroke, joint-replacement, post-cardiac operation and others. PTs use specific rehabilitation techniques where they must be well documented with respect to specific measurement parameters to gauge an individual's performance toward full/partial recovery or a targeted goal. The rehabilitation program normally requires an individual's attendance periodically over a significant period of time, depending on the recovery speed. These tasks could be perceived by the individual undergoing the treatment as a tedious and repetitive act. This negative mentality can be detrimental

for the individual as the recovery is delayed due to mental rejection to the exercises and/or deliberately not attend sessions regularly - this possible effect can be exacerbated if the individual is also suffering from depression.

2. Hardware

To sense the user's motions we have opted for solely internal sensing mechanism as this increase the flexibility of the overall system during its use, as opposed to an external sensory system, such as an optical based system which would rely on structured environments i.e. lighting, objects and also restrict users to be visible to the camera's field of view.

The hardware system used is an existing product in the market named the WiTilt v3 that has been developed by Sparkfun Electronics[1] which is a 4-axis IMU

(Inertial Measurement Unit) costing only USD200 which communicates wirelessly to any compatible Bluetooth device, in this instance, a PC. It contains 2 MEMS sensor chips, a 3-axis accelerometer and a single axis gyroscope. Figure 1 below depicts the device:

As opposed to some other research which

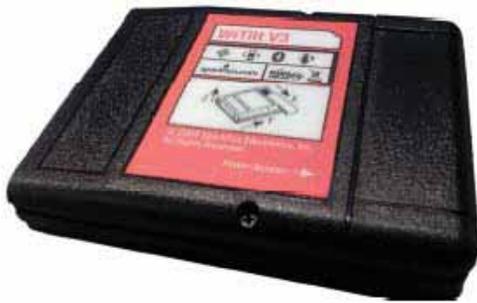


Figure 1. WiTilt v3.0 from Sparkfun

had made use of the Wii[®] console by Nintendo used in conjunction with the existing non-targeted games in the genre of Wii Sports for patient exercise. This pathway disallow any specific performance analysis as the main purpose of the device is a gaming console, additionally the Wii Remote is strictly a hand-held device.[2]

This product was chosen for use because of its readiness for application with suitable sensory ranges for detecting human motion (up to 6Gs and 150 degs/s) and thus we have allocated more time to develop and refine the software aspects of the system.

With only 4 DoF, the device is not a full cluster IMU and cannot perform a complete orientation sensor without an external reference therefore the device must be mounted according to our specified orientations in order for the orientation detection algorithms to perform correctly.

3. Software

The software system was developed in and for the Microsoft Windows environment,

where the .NET framework was utilised in C# and the virtual environment was developed with the XNA development platform which contained the essential elements in order for us to simply create interactive environments.

The overall architecture of the software can be divided into 4 major parts, the graphical user interface (GUI), the communication, the game and the database system. Some of these parts run on separate threads to optimise computation efficiency, such as in the case of the game environment and the



Figure 2. Sample image of the program

GUI. Effort was put in to ensure efficient use of resources in terms of the CPU cycles and the RAM to maintain the robustness of the software across diverse hardware environments.

The GUI of the program is based on C# windows forms for increased user familiarity with the existing Windows based programs. As seen in Figure 2, the program looks and feels similar to typical PC programs.

4. Hardware Interface

The WiTilt device has an onboard firmware along with an interface which communicates either wired or wirelessly and sends out the 4 DOF sensor readings, along with the availability of options to change settings and the calibration of the sensors.

The Bluetooth communication between the device and the PC connects via Bluetooth

only and henceforth communicates serially. The program utilises the .NET framework's serial port to further simplify the communication handling from the device to the program's internal memory.

5. Game Design

The game was developed with the XNA development platform due to its tight integration with the Microsoft Visual Studio and mainly due to its native C# language.

The use of the development platform has been very beneficial to the development of this project as the platform provides a hardware abstraction layer where there was no need to handle low-level hardware interfaces, only the high-level XNA libraries that are available.

The platform is capable of creating 2D and 3D virtual environments with associative sound effects, useful in our case to increase user immersion experience. Presently though, the virtual environments that have been developed have only been in 2D. Preliminary findings have found that moving from 2D to 3D, the development would not be alarmingly different and thus easily implemented.

The objective of the game design is to immerse the user in the virtual environment and become distracted from the exercise at hand. This reduces de-motivation and thus prolongs therapeutic exercise sessions which in turn hasten recovery time of the patients [3]. To increase the immersion of the user in

the environment, 2 concepts have been trialled.

Firstly is the virtual environment displayed to the user, what we have concluded is that



Figure 3. Footsie exercise game

the environment should resemble real-life activities as seen in the following Figure 3.

For example, a game that has been developed to exercise the lower limb has the footsie theme to complement the exercise movement that is being targeted as a swing of the leg as the exercise.

6. Data Analysis

The data from the WiTilt device can be transmitted at different frequencies up to 135Hz under the "g" mode (the device computes the raw sensor outputs and transmits the values in terms of g's of acceleration).

Given the raw data sent from the device, the program filters them to minimise noise and random fluctuations present. Figure 4 below shows the result of using a low-pass filter to enhance the resulting data for improved processing.

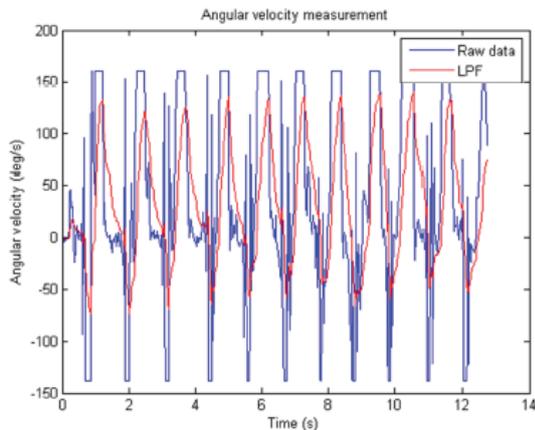


Figure 4. Low-pass filtered (LPF) gyroscope data

The LPF used here is of an exponential smoothing variation designated by the given difference equation:

$$y_k = y_{k+1} + \alpha(x_k - y_{k-1})$$

where y is the filtered data and x the raw data and α is the smoothing factor, the filter begins work with only 2 time steps and is low in computation overhead[4].

The use of a Kalman filter has been attempted but yet to be applied as a robust attitude estimator for detecting the pitch and the roll of the device through the combination of the accelerometer and gyroscope data. Currently only the accelerometer data is used to approximate the attitude[5][6].

7. Database System

The implemented database system uses SQL to handle the communication with the associated database file. SQL was chosen due to its direct integration with the Visual Studio and its simplicity which also offers security of the database through a password system.

The database stores the user credentials plus any other relevant information to the therapist such as symptoms etc.

Currently the program stores the filtered sensor data for re-graphing for later analysis. This current implementation has major disadvantages such as heavy disc I/Os as the data is written to file as they are received and hence a significantly large file size per recorded session. In order to improve/rectify these issues, suggestions have been made to perform analyses first and store only the relevant, significant data such as cycle counts and peaks and troughs.

Given the recorded patient performance data, they can be used to plot the trends over time for the therapist to review the recovery progress of patients. Specific analytical data such as pulse widths and peaks/troughs have yet to be implemented as work in progress.

8. User Survey

Preliminary trial of the system performed internally wished to obtain feedback from the users regarding issues such as the user interface of the program, the intuitiveness of the games and their robustness.

Unfortunately due to time constraints, we were not able to trial the system with real patients and physiotherapists and thus had to resort to internal testing with uninvolved colleagues that poses no objective bias towards the project.

The survey was also divided into 2 parts; quantitative and qualitative feedbacks. Both parts were queried for each of the categories in question and only some will be addressed here.

The surveyees were asked to score from the scale of 1 to 5, where 5 is best of each of the categories. The median results are as

follows; 5 for GUI, 4 for game intuitiveness and 3 for game robustness. Qualitative comments from the surveyees proved to be useful feedbacks as they had addressed critical issues identified within the system, both hardware and software.

An example of the feedback regarding the game was that the games looked “too simple”. We acknowledged that fact and have begun the development of improved visuals to better immerse the users.

Through the use of the system and the related games, the users were more inclined to perform the repetitive movements for a longer period of time as opposed to without the system.

9. Conclusion

We have integrated the sensory hardware of a 4-axis IMU with our developed C# program to display real-time simulated interactive environment for patient interaction and embedded history system of their performance for analysis on a PC. This is in an attempt to increase therapy efficiency and also improve the motivation level of patients and through preliminary findings, the results are promising.

Acknowledgements

We appreciate the people that have assisted in the undertaking of this project especially our project supervisor, Dr. KC Aw for sound advices and Stuart Logan for lab assistance.

References

1. Sparkfun Electronics. Retrieved June 27, 2010 from: www.sparkfun.com.
2. Halton, J. (2008) Virtual rehabilitation with video games: A new frontier for occupational therapy. *Occupational Therapy Now*. 9.6.
3. Burke J.W., M. D. J., Charles D.K., Morrow P.J. (2009) Serious Games for Upper Limb Rehabilitation Following Stroke. *Proc. Games and Virtual Worlds for Serious Applications*. 103-110.
4. Duke, U. Averaging and Exponential Smoothing Models. 2005 Retrieved July, 2010 from: duke.edu/~rnau/411avg.htm
5. Seong-hoon, W., Melek, W. and Golnaraghi, F. (2008) Position and orientation estimation using Kalman filtering and particle diltering with one IMU and one position sensor, in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*.
6. Vinande, E., Axelrad, P. and Akos, D. (2010) Mounting-Angle Estimation for Personal Navigation Devices. *Vehicular Technology, IEEE Transactions on*. 59(3), 1129-1138.

Low-Cost Organic Thin Film Transistor Printer

Taochang (Darien) Liu, Yoshimi Sekikawa

The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract

OTFT technology shows promise as being the next step forward for display technologies with its ability to be applied to flexible substrates and capacity to cover larger areas compared to current silicon based thin film transistor technology at lower manufacturing and material costs. This project concerns the integration of a low-cost print-head into a 2-axis printer frame to realise a low-cost organic thin film transistor (OTFT) printer. The objective of the project is to integrate the project work completed in previous years into a useable OTFT printing device that interfaces with the PC using a universal serial bus (USB) connection.

Keywords: *Organic Thin Film Transistor, OTFT printer*

1. Introduction

The use of semiconducting organic material in the production of electronics has several advantages: it has the ability to be fabricated on biodegradable substrates, thus dramatically reducing hazardous waste material; electronics using organic materials can be made using less energy and raw material and with 100 times less CO₂ emissions than with silicon based equivalents [1]. Organic materials have little effect on the environment.

It was therefore not surprising to find that research and development into OTFTs have been on the increase. Thin Film Transistors (TFT) are widely used in commercial Liquid Crystal Display (LCD) screens for its ability to control each pixel with its own individual transistor and for its high switching speeds. One of the most significant features of OTFTs is that they are flexible, so there is research in flexible active matrix displays.

OTFT can also be used in flexible large-area electronics, wearable electronics and sensors.

There are a number of known methods of fabricating OTFTs, the main two being, vapour deposition, and direct solution application [2]. Compared to these two methods, direct printing of organic semiconductors has several advantages: there is very little to no waste product; the time it takes is relatively short; it is simple to define the exact area of printing; and the equipment can cost much less than the other complex methods. A low cost OTFT printer is therefore desired. One has been under the development at the University of Auckland. In previous years, two parts of the printer were developed separately: the base printer was chosen and a positioning system using stepper motors was implemented; and a prototype print head with electronic circuitry was developed.

2. Hardware

Base Printer

An HP 1120c printer was chosen as a base for the OTFT printer. The original chassis, cartridge holder (or carriage) and slide rail was retained for this project.

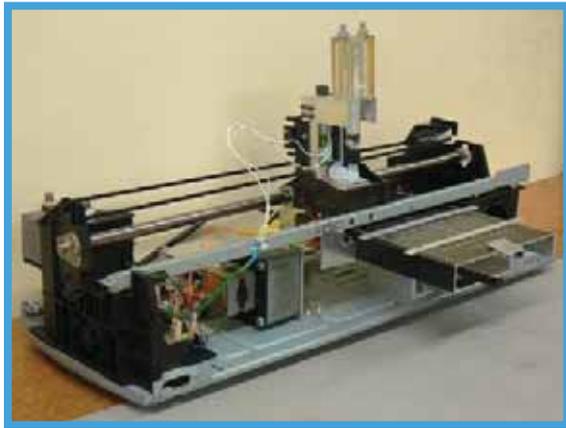


Figure 1. 2010 OTFT Printer Design

Positioning system

The positioning system involves two stepper motors with reduction gearboxes, a transmission belt, and a rack and pinion mechanism. X-axis movement of the print head is achieved by a stepper motor and the belt drive, and the Y-axis movement of the platform is done by a stepper motor and the rack and pinion mechanism. The motors have a step size of 1.8° , accuracy of 5%, coil charge time constant of 0.4ms, and max current draw of 0.16A each. The gearboxes' step down ratios are 125:1 and are directly connected to the stepper motors. Simple limit switches were implemented in order for the positioning system to have a home position.

Print head

The method of injection is similar to that of a fuel injector in an automotive engine. The pressurised fluid sits behind a nozzle, which is blocked by a pin. The pin is inside an electrical solenoid. When it actuates, the pin momentarily unblocks the nozzle, allowing a

small amount of fluid to escape due to the pressure difference, before blocking it again. In this case the pressure is created by a compressed spring, which creates a force on a syringe filled with fluid. The smallest deposit volume of the prototype was determined to be 20nL. Actuation times were between 4ms and 10ms.

The print head was designed so it would fit inside the original ink cartridge holder. It is rigidly held in place using screws on the side.

Electronics

There are various voltages required for components on the printer: the stepper motors, 12V (max 0.32A); motor optocoupler, 5V (max 0.1A); solenoid, 7.0 – 8.0V (max 1.5A); and solenoid optocoupler, 7.0 – 8.0V (max 0.1A).

Power Supply Unit (PSU)

The Meanwell 12V (max 4.2A) Power Supply unit was bought, at a cost of NZ\$40, a low price for a power supply that will power the printer's components, while being a safer and more reliable unit than the printer's original supply which was used in previous years. The unit was mounted upside down to prevent possible human contact or tampering with the live terminal blocks. The resulting area around the power supply is much neater, safer and professional looking.

Solenoid driver circuit

The solenoid can draw up to a maximum current of 1.5A at 7.5V. A step down from 12V was required. Previous year's linear voltage regulator overheated in a short amount of time, so it was replaced with a switch mode regulator, which was more efficient and had a higher current rating. The new regulator did not overheat even after long periods of use. The driver for the

solenoid consisted of an opto-coupler, and a darlington transistor.

The circuit was designed and fabricated onto a Printed Circuit Board (PCB), which is compact and easy to mount on the printer.

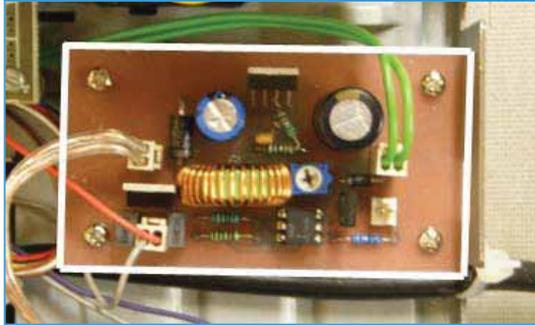


Figure 2. Solenoid driver circuit

Stepper motor driver circuit

The motor uses the 12V from the PSU. The driver circuit is similar to the solenoid one, with opto-couplers and darlington-pair transistors.

3. Software

Overview

To realise the printer control, separate software had to be developed for the microcontroller and the PC. The PC control software converts the user input printing schematic into movement instructions that are then sent to the printer. The embedded microcontroller software handles the real time hardware execution of the stepper motors and print-head according to instructions provided by the PC control software. The PC Control software interprets the desired printing pattern input in a non time critical process, generating a list of direction and distance values for both the x and y axes that are converted into executable instructions to be transmitted down the communication line when requested by the microcontroller.

UART Communication

Communication between the PC and the microcontroller is carried serially out using the universal asynchronous receiver/transmitter (UART) pins of port D of the Atmega8 microcontroller. The voltage levels for the logic used by the microcontroller utilise the RS232 standard for UART communications with +3V to +25V used to signify a space (with a logic value 0) and -3V to -25V used to signify a mark (with a logic value 1). RS232 is different to the TTL logic levels used by modern PC connections. The MAX232 chip provides the interfaces circuitry to convert between these two logic levels.

The UART is configured for sending 8 bit data with 1 stop bit and no parity bits at a baud rate of 2400. A high baud rate is not required for our application as the majority of the software execution time is spent actuating the stepper motors meaning that a lower baud rate is sufficient. The 8 bit binary data strings can be described using the 'unsigned char' data structure, allowing ASCII representations of characters and numbers to be sent and received.

Serial Transmission of Integers

The maximum numerical value that an 8 bit binary number can represent is only 255, which severely limits the usefulness of transmitting numbers through UART. As larger numbers require sending arbitrary amounts of additional ASCII digits down the transmission line, an algorithm was developed to disassemble larger numbers into their ASCII digit representations in order to transmit and receive them serially. The algorithm is able to transmit and receive numbers of the 'long unsigned integer' data structure, which is able to store a 32 bit binary number representing values up to 4,294,967,295. Numbers that require multi-digit representation can be decomposed into individual ASCII digits which can then be transmitted through UART with the additional declaration ASCII numbers that lead the number transmission to describe to the

Sensor and Actuator Integration for Autonomous Guided Vehicle

Peter Hosking, Josh Stroobant

The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract

This project was a continuation of previous work on an Autonomous Guided Vehicle (AGV) capable of real-time obstacle avoidance. In 2009 an AGV was developed using an DE-2 FPGA board and a Hokuyo UBG-04LX-F01 Laser Range Finder (LRF). The UBG-04LX-F01 offered RS-232 communication which was well suited for the FPGA controller. In this project, UBG-04LX-F01 needed to be replaced with a Hokuyo URG-04LX-UG01 LRF. The URG-04LX-UG01 offered USB communication only and was therefore no longer compatible with the DE-2 FPGA board. The objective of this project was to restructure the AGV so that an AVR ATmega328P could be used as a controller in place of the FPGA. The ATmega328P is an improvement in terms of simplicity, cost, power consumption, and compatibility with sensory devices. In particular, AVR microcontrollers offer support for the method of communication used by the URG-04LX-UG01. As well as establishing the ATmega328P as a controller, a number of sensors were integrated into the AGV system. Among these sensors were an Inertial Measurement Unit (IMU), an electronic compass, and an ultrasonic range sensor.

Keywords: *Autonomous Guided Vehicle (AGV), Inertial Measurement Unit (IMU), Electronic Compass, Ultrasonic Sensor, Laser Range Finder (LRF), AVR microcontroller*

1. Introduction

An Autonomous Guided Vehicle (AGV) is a mobile Mechatronics system that can assess its environment via sensors, make decisions via a controller, and act on those decisions via actuators. Sensors used in AGVs include obstacle detection, angular position, translational position, and global position sensors. AGVs have been implemented in many industries ranging from the defense industry to health and logistics [1]. The advantages of using AGVs include reduction of operational costs, a safer working environment for people, and less error during processes [1].

This project is a continuation of previous work on the development of an Autonomous Guided Vehicle (AGV). Specific focus is given to real-time obstacle avoidance which is facilitated using carefully selected sensors that allow the AGV to perceive its surroundings.

In 2009 a four wheeled AGV system driven by two DC motors was developed using a DE-2 FPGA board as a controller. The system included a Hokuyo UBG-04LX-F01 Laser Range Finder (LRF) which was used as a vision sensor. A modified version of the VPH+ algorithm was implemented to demonstrate basic obstacle avoidance.

As of 2010, the Hokuyo UBG-04LX-F01 is no longer available. In its place, a lower end model called the Hokuyo URG-04LX-UG01 has been acquired. This model doesn't support the same communication methods as the UBG-04LX-F01 and is no longer compatible with the DE-2 FPGA board. As a result, the objective of this project was to restructure the AGV using an AVR microcontroller instead of the FPGA. The AVR microcontroller is an improvement in terms of simplicity, cost, power consumption, and compatibility with sensory devices. In particular, AVR microcontrollers offer support for the method of communication used by the URG-04LX-UG01.

Integration of the AVR microcontroller has been successful along with a number of additional sensory devices. Among these devices is an Inertial Measurement Unit which can be used to determine angular displacement. Also an electronic compass has been integrated which allows the AGV to determine its bearing relative to earth's magnetic field. Lastly, an ultrasonic range sensor has been integrated to enable object detection. Integration of the URG-04LX-UG01 itself has not yet been addressed due to time constraints. It is anticipated however, that this project will provide the groundwork for the URG-04LX-UG0 to be integrated in the near future.

2. AVR Microcontroller

In previous years, a DE-2 FPGA board was used as a controller for the AGV system. In this project, the board was found to be no longer suitable. This was because of large

configuration and compilation times, limited support, and back compatibility issues with older software versions. In particular the DE-2 FPGA board did not support USB communication with the Hokuyo URG-04LX-UG01 Laser Range Finder. The URG-04LX-UG01 was intended to be integrated as the primary vision sensor for the AGV system.

Due to the difficulties mentioned above, the DE-2 FPGA board was replaced this year with an ATmega328P AVR microcontroller. The ATmega328P is both low cost, low power, and offers a range of peripheral devices that are essential for the integration of sensors and actuators.

The ATmega328P ensures greater compatibility with future work on the AGV. It has 23 programmable input and output lines. AVR microcontrollers can be programmed using embedded C and future revisions of the AVR development studio and other software will not affect the back compatibility of older code. Peripheral devices on the ATmega328P include Timers, PWM channels and a 10-bit resolution ADC. These devices reduce the need for external circuitry to convert output from sensors. The PWM output channels can be used to control motor speed. Most importantly, AVR microcontrollers offer support for USB communication with the Hokuyo URG-04LX-UG01 Laser Range Finder.

3. Interfacing Circuitry

The interfacing circuitry in the AGV system consists of an STK500 and a pair of

identical routing boards designed to accommodate sensor and actuator modules.

The STK500 is a development board manufactured by Atmel specifically for interfacing with AVR microcontrollers. The STK500 handles the power supply to the ATmega328P and has easily accessible pins for access to the I/O ports.



Figure 1. STK500

All sensory devices and actuators have been designed as modules that connect with the STK500 via a standard routing board design. Each routing board consists of three 10-pin slots which connect with the sensor and actuator modules. Each of the 10-pin slots on a routing board is wired together and shares the same I/O lines from the ATmega328P. Modules are designed to utilise only the I/O lines that they require and leave the others disconnected. In this way, it doesn't matter which slot the sensor and actuator modules are plugged in to as long as they have been designed not to use the same I/O lines from the ATmega328P.



Figure 2. Routing Board PCB

The use of slotted module designs has three primary advantages. Firstly, circuit design is simplified as modules can be built and tested

one at time. Secondly, future sensory devices can be added to the system without modifying existing circuitry. Lastly, it reduces wire clutter and therefore mitigates 'noise' on sensor input lines.

4. Power Supply

The power for the AGV is supplied by custom built lithium ion battery. The battery is capable of supplying 16.8V at 2.5Ah. AGV system components require a voltage supply between 10-15V. For this reason a step down voltage regulator was constructed using an LM2576 switching IC.



Figure 3. Voltage Regulator

The LM2576 is capable of regulating voltages between 7-60V with a guaranteed maximum current output of 3A. The regulated output voltage can be adjusted simply by changing resistor values.

5. Inertial Measurement Unit

The Inertial Measurement Unit (IMU) is a single axis gyroscope which has been integrated into the system for the purpose of estimating angular displacement. The model number of the IMU which has been integrated into the system is the ADXRS300. Compared with other position sensors such as GPS and Electronic compasses, IMUs are less susceptible to environmental interferences and offer faster response times.



Figure 4. IMU Module

The ADXRS300 produces a voltage output between 0.25V and 4.75V that is proportional to its angular velocity, $\dot{\phi}$.

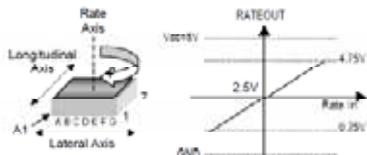


Figure 5. IMU Output Characteristics

Angular displacement ϕ at time t is determined simply by integrating angular velocity:

$$\phi = \int_0^t \dot{\phi} dt$$

A built in ADC on the ATmega328P is set to convert the output from the IMU continuously at a rate of 62.5 KHz. Integration is approximated in software by adding each conversion result to the previous one:

$$\phi \approx (ADC_{previous} - ADC_{equilibrium}) + (ADC_{current} - ADC_{equilibrium})$$

This calculation is performed within an Interrupt Service Routine which is triggered each time the ADC completes a conversion.

5. Devantech Electronic Compass

Signal Processing

The R117 Devantech Electronic Compass that replaces the Dinsmore R1525 uses two Philips KMZ51 magnetic field sensors,

Copyright © 2010 The University of Auckland, Mechanical Engineering.

which are sensitive enough to detect Earth's magnetic field. They are mounted at right angles to each other to compute the direction of the horizontal component of Earth's magnetic field [27]. The Devantech compass outputs a Pulse Width Modulation wave that changes between +5V and 0V TTL. The positive width of the pulse (Duty Cycle) represents the size of the angle. This positive pulse width varies from 1 millisecond (for 0 degrees) to 36.99ms (for 359.9 degrees). This means that the signal is 0.1ms per degree with a 1ms offset [2].

The reading of this pulse width is implemented using a timer interrupt in the ATmega 328P. When the timer overflows (reaches its maximum value, 256 in this case) the compass output is inputted. If the compass output is high, the raw angle is incremented. If the compass output is low, the raw angle is stored into another variable, and the raw angle is reset. The Devantech compass output runs at a maximum frequency of 15Hz (for 0 degrees). The timer interrupt runs at a frequency of 222 kHz, therefore the interrupt is running at a fast enough frequency to make full use of the 10 μ s accuracy of the compass.

Calibration

The Devantech compass module has been factory calibrated in Devantech's workshop for inclination, which is 67 degrees north [2]. This is not suitable for use in Auckland, which is at 37 degrees south, because of the angle of tilt of earth's magnetic field. Therefore the compass had to be recalibrated for use in Auckland. This calibration involves the following steps [2]:

1. Set the compass module flat, pointing north. Briefly press and release the switch (bringing the CAL pin low). The NC pin (pin 5) goes low; illuminating the LED indicating calibration is in progress.
2. Set the compass module flat, pointing east. Briefly press and release the switch.
3. Set the compass module flat, pointing south. Briefly press and release the switch.
4. Set the compass module flat, pointing west. Briefly press and release the switch. The NC pin (pin 5) goes high, turning off the LED and indicating calibration is complete.

The compass has been characterized and tested and results show that the parameters found are accurate in determining the bearing in degrees and can therefore be used in the AGV.

6. Ultrasonic Sensor

The UM30-13113 ultrasonic sensor uses a piezo element to generate ultrasonic waves. This same piezo element detects and measures the ultrasonic wave when reflected off an object, effectively acting as a transceiver [3]. The distance between the object and sensor is calculated by measuring the time taken for the ultrasonic wave to travel through the air. This is outputted via an analogue voltage from 0V to 10V. For the ATmega 328P to read this voltage into the ADC, the output voltage must be halved using a potential divider. Ten readings are then taken with the ADC running in single conversion mode, and these ten results are

Copyright © 2010 The University of Auckland, Mechanical Engineering.

averaged to reduce the impact of noise. The ultrasonic sensor runs off the power supply circuit at 10.5V.

7. DC Motors and Drive Circuitry

Signal Processing

The DC motor driver PCB containing two L298 dual full bridge driver Integrated Circuits (ICs) was tested using the microcontroller. The bottom level of the AGV showing the DC motors and the PCBs is illustrated in Figure 6 below. This figure also shows that the two front DC motors were removed. This was due to the fact that they had been de soldered during the 2009 project, and therefore there was no reason to leave them in the AGV.

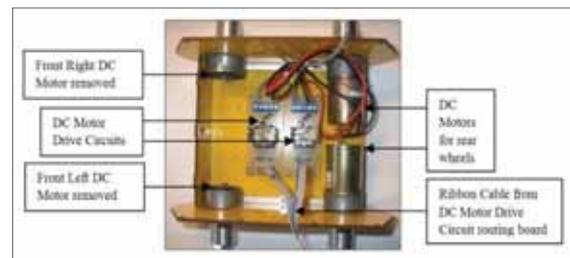


Figure 6. Bottom level of AGV

A simple routing PCB was designed to route the signals from Port B of the ATmega 328P to the DC motor driver circuit. This PCB plugs directly into one of the routing boards mentioned earlier for ease of use.

Pulse Width Modulation was employed to vary the duty cycle of these inputs (the ratio of high time to low time in a signal), and therefore control the rotational speed of the motors. This is required when the AGV needs to go at a set speed, or when it is

turning (and therefore one wheel of the AGV is set at a faster speed than the other). PWM was implemented by using one of the counters in the ATmega 328P operating in bidirectional mode (counting up to a maximum value, then counting down from that value). The current value of the counter (the carrier wave) is then compared to an output compare register (the modulating wave). If the counter value is higher than the set value, the output is one, and if lower, the output is zero. This is shown in Figure 7 below:

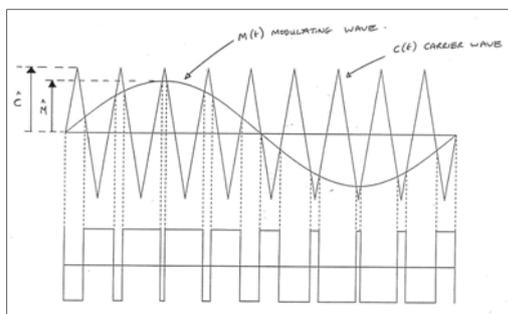


Figure 7. Pulse Width Modulation (Reproduced from [4])

Therefore, the rotational speed of the DC motors is directly proportional to the value in the output compare register for that motor, and this is how the speed of the DC motors is controlled.

Proportional Integral Controller

The power given to the motors is controlled by the PI controller. The proportional control is given by the angular difference between the current angle measured by the IMU, and the target angle that the AGV is turning towards. This difference is then multiplied by K_P , the proportional gain of the controller. The integral control is given by integrating the difference between the

current angle and target angle with respect to time. This is achieved in the digital system by adding the current angular difference to a cumulative total every time the function runs through. The integral is then multiplied by K_I , the integral gain of the controller. The equation for the controller is shown below:

$$\text{Motor Power} = (\text{Current Angle} - \text{Target Angle}) * K_P + \text{Integral} * K_I$$

This PI controller is run until the current angle is within a specific angular tolerance from the target angle, in which case the motors are turned off and the integral reset.

The PI controller was initially tuned using a zone-based tuning procedure. For a PI controller in particular, this involved the following steps [5]:

1. Set integral gain (K_I) to zero
2. Set proportional gain (K_P) low for system stability, i.e. so that for a 90 degree turn, the AGV will turn at a reasonable rate to 90 degrees without oscillating out of control
3. Apply a turning function (90 degrees in this case)
4. Increase K_P to a maximum value to make the AGV turn as quickly as possible without overshoot
5. Increase K_I for an overshoot of approximately 10% (approximately 9 degrees overshoot for a 90 degrees turn)

After this procedure was applied, the gains were fine-tuned manually until a desired rise time (rate of turn) and overshoot were obtained.

8. Software

The finalised software program consists of a main c file containing the main function as well as integrated functions. There are five other c files which contain function libraries. These are organised so there is one function library for each sensor and actuator.

9. Testing

Proportional Integral Controller

To test the accuracy of the Proportional Integral (PI) controller, a simple experiment was conducted. This was carried out in the following steps with the angles 45, 90, 135, and 180 degrees:

1. The “Turn_onSpot” function (found in the DC motor function library) was given a clockwise angle as an input
2. The ATmega 328P was programmed with this function
3. The AGV was placed on a large protractor coordinate system facing a bearing of zero degrees
4. The program was initiated by signaling in front of the ultrasonic sensor
5. Once the AGV had settled to within the angular tolerance of the PI controller (1 degree), the angle was measured using a protractor
6. The angular error was calculated as the angle measured minus the angle inputted to the function

These steps were repeated ten times per angle to give an appropriate sample size. The box and whisker plots in Figure 8 show the variance and accuracy of the angular errors resulting from the turns.

Copyright © 2010 The University of Auckland, Mechanical Engineering.

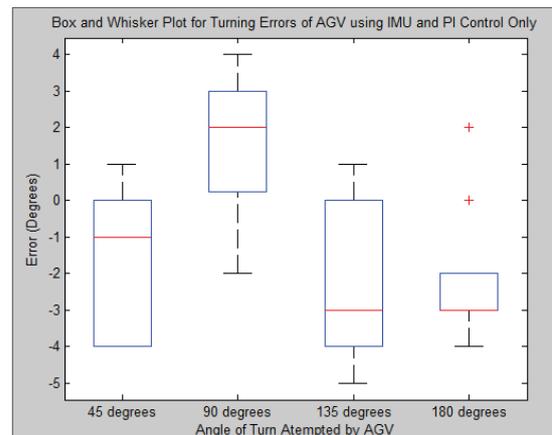


Figure 8. Box and whisker plots for turning errors of AGV using IMU and PI control only

These errors are all within plus or minus five degrees, with the majority of the results (inter quartile ranges) within plus or minus four degrees. Therefore the PI controller is accurate and repeatable enough to use in an obstacle avoidance function.

Turning Functions with Compass Checking

The turning function with compass checks before and after the turn was tested using similar methods as the PI controller. Figure 9 shows the box and whisker plot produced for a 45 degrees turn attempted by the AGV. Once again, the mean of the errors is close to zero error (+0.9 degrees); and the variance is acceptable with a standard deviation of 3.4 degrees.

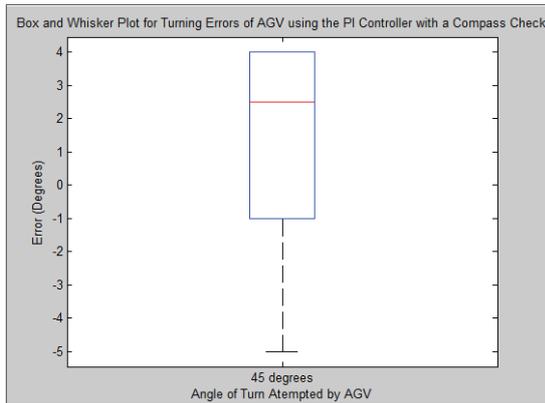


Figure 9. Box and whisker plot for turning errors of AGV with a compass check

The results show that the AGV is accurate at turning to a specific bearing inputted from the program. This means the main objective of this project has been met, which is to select and integrate a controller and peripheral sensory devices and demonstrate the AGVs ability to follow a set path.

10. Conclusions

- It has been decided to use the AVR ATmega 328P microcontroller, and this controller has been successfully integrated into the system.
- A power supply system has been designed with a light weight battery and switching mode step down voltage regulator.
- The LCD screen has been integrated.
- The ADXRS300EB Inertial Measurement Unit has been integrated into the system as a relative angular position sensor.
- Originally the Dinsmore electronic compass was to be integrated however, due to its non-linearity; this has been replaced by the Devantech electronic compass. This Devantech

compass is more linear, repeatable and accurate.

- The UM30-13113 ultrasonic sensor has been integrated into the system as a backup obstacle detection sensor.
- The URG-04LX-UG01 Scanning Laser Range Finder (LRF) has not been integrated so far, but its integration has been researched and it is found that the AT90USB1287-16AU AVR is capable of USB communication with the LRF.
- The DC drive motor actuators have been integrated along with a PI controller for angular turning control.
- Software function libraries have been designed for each of the sensors and actuators.
- These software function libraries have been implemented in the main function for testing.
- Testing results show the AGV can turn accurately using the IMU and PI controller with compass checks.
- Overall, the AGV has had a large reduction in complexity, cost, weight, and power consumption with the change in controller platform. There has been an increase in efficiency and potential for further sensor integration. Most importantly, this is a better platform in which to develop an obstacle avoidance algorithm. Therefore the main objectives of this project have been met.

Acknowledgements

We are indebted to our supervisor Dr. K.C. Aw and our lab technician Logan Stuart for their support and expertise.

References

1. Frog AGV Systems. Questions. (2008-last update). [Online]. <http://www.frog.nl/Service/Questions> . Retrieved on 2010, September 19.
2. CMPS03 Compass Module. [Online]. <http://www.robot-electronics.co.uk/html/cms3tech.htm> Retrieved on 2010, September 19.
3. *Ultrasonic Sensors*. [Online]. <https://www.mysick.com/saqqara/get.aspx?id=im0016670> Retrieved on 2010, September 19.
4. Generation of Pulse Width Modulation using Natural Sampling. [Online]. http://hermes.eee.nott.ac.uk/teaching/h5cpe2/triangular_PWM.gif . Retrieved on 2010, June 26.
5. Tuning a PI Controller. [Online]. <http://www.qxdesign.com/Visual%20ModelQ/Training/Tuning%20a%20PI%20controller.pdf>. Retrieved on 2010, September 19.

Temperature Sensing with SAW-based Sensor

T. J. Giffney, Y.H. Ng

The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract

Sensing low concentrations of gas using surface acoustic wave (SAW) devices requires excellent temperature control. The propagation velocity of surface acoustic waves is temperature dependent. Changes in temperature, and changes in gas concentration, will both affect the frequency response of a SAW gas sensor. The temperature sensitivity of a SAW ethanol sensor was investigated. The design used for testing was fabricated on a 128° rotated Y-Z LiNbO_3 substrate and consists of two interdigital transducers (IDTs), each containing 50 electrode pairs. The width and spacing of the electrodes is $15\mu\text{m}$, resulting in a central frequency at room temperature of approximately 66 MHz. Measurements confirmed that the temperature dependent frequency shift was linear between 35°C to 220°C , with a sensitivity of $-6.40\text{ kHz}/^\circ\text{C}$ and can be exploited as a temperature sensor.

Keywords: *Surface Acoustic Wave, Temperature Sensor, Interdigital Transducer*

1. Introduction

The authors have developed a SAW ethanol sensor based on ethanol adsorption onto zinc oxide nanorods. A schematic of the sensor is provided as Figure 1.

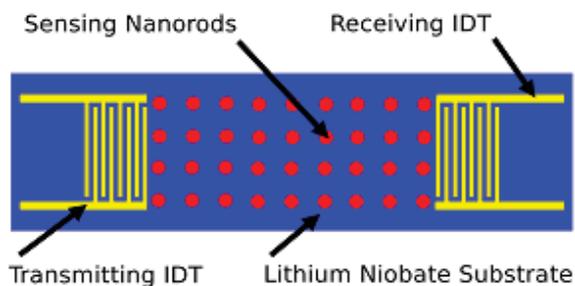


Figure 1. SAW ethanol sensor

Two identical interdigital transducers are fabricated on a LiNbO_3 substrate. The space separating the IDTs forms a surface acoustic wave delay line. Surface acoustic waves are generated by applying a sinusoidally varying voltage across the transmitting IDT. The waves propagate along the delay line and, on reaching the receiving IDT, are transduced back into voltage.

In order to sense ethanol a layer of zinc oxide nanorods is grown on the delay line. At high temperature ethanol is chemically adsorbed onto zinc oxide, with an accompanying change in mass and conductivity. The amount of ethanol adsorbed is itself temperature dependent.

The frequency response of the sensor depends on the speed at which the SAW propagates along the substrate surface. Any change to the balance of masses, stiffnesses, and distances near the surface of the LiNbO₃ crystal will alter the SAW propagation velocity. As such, changes in the operating temperature, and mass loading resulting from ethanol adsorption will both affect the frequency response.

In order for the device to be a useful gas sensor, it is necessary to distinguish between the effect of ethanol sensing, and the effect of temperature disturbances. However, this temperature sensitivity can be exploited for temperature sensing.

2. Design Parameters

The main parameters affecting the performance of a SAW device are:

- The substrate material
- The substrate orientation
- The SAW propagation direction
- The design of the IDTs
- The length of the delay line

Temperature sensitivity is predominantly influenced by the substrate material, the substrate orientation, and the centre frequency at room temperature. The use of a higher operating frequency will increase sensitivity to gas, but also increase temperature sensitivity.

The devices used in this investigation were fabricated on 128° rotated Y cut LiNbO₃, with the IDTs oriented so that the SAWs propagate along the x-axis of the crystal.

The IDTs consist of 50 electrode pairs each. The width of each electrode is 15µm, and the separation between the edges of adjacent electrodes is also 15µm. This IDT design

generates SAWs most efficiently when the SAW wavelength (λ) is 60µm.

For SAW waves in 128° Y-Cut LiNbO₃ with the propagation path orientated along the x-axis, the SAW propagation velocity (v) is 3980m/s at room temperature. The centre frequency is therefore estimated as:

$$f_n = \frac{v}{\lambda} = 66.3MHz \quad (1)$$

3. Fabrication

The performance of SAW devices is strongly influenced by the quality of fabrication [1]. To avoid contamination with small particles, fabrication was carried out in the cleanroom of the University of Auckland Microfabrication Facility.

Surface acoustic wave devices are commonly fabricated using either lift off or etching methods. LiNbO₃ has weak adhesion with photoresist, making production through lift off process more difficult. Therefore an etching method, where there is no direct adhesion between LiNbO₃ and photoresist required, was preferred.

Metal Evaporation

The first step in the fabrication process was the deposition of a uniform metal layer to form the conductive parts of the device. This was accomplished using a JEOS vacuum evaporator.

Gold was chosen as the electrode material due to its conductivity, non-reactive nature, and simple processing. However, as gold does not adhere well to LiNbO₃; a layer of chromium was deposited first to promote adhesion.

Vacuum evaporation was performed in batches. The LiNbO₃ wafer was first cleaned to improve the quality of metal adhesion.

After cleaning, the wafers were arranged on a glass plate in the bell jar. Vacuum evaporator setup was then completed by placing chromium and gold in the tungsten wire filaments.

Subsequently, the rotary pump and the diffusion pump were used to evacuate the bell jar to a pressure below 5×10^{-4} Pa. A 30nm thick layer of chromium and a 150nm thick layer of gold were then deposited on top of the LiNbO_3 wafer by vacuum evaporation. The thickness of the material layers was verified using a surface profilometer.

Photolithography

The second step in the fabrication of the temperature sensor was photolithographic patterning. This process creates a photoresist layer which is a positive image of the IDTs and contacts. This photoresist layer will act as a protective coating later in the etching process. As photoresists are UV-sensitive, patterning must be carried out in a room with UV-free lighting facilitated in the Microfabrication Facility cleanroom.

The sample was first covered with an even layer of nLof 2070 photoresist using spin coater, Laurell WS-400B-6NPP/Lite. After completing that, the sample was baked to improve the photoresist adhesion to the sample [2] and to reduce the solvent content of the photoresist, thus reducing the mask contamination during contact printing. The baked sample was then placed in contact with a negative sub-mask and exposed to UV light using the mask aligner. After that, the sample was baked again for the advance of the photoresist crosslinking mechanism and to reduce under-etching in the subsequent etching process [2]. The photoresist was then developed using developing solution, Az 326.

Etching Process

In etching process, the metal layers that are in the unwanted area are to be removed. The sample was etched first in gold etchant solution, followed by chromium etchant solution. After completing these steps, fabrication of the SAW device itself was now complete. Figure 2 shows a SAW device on lithium niobate. Additional process steps not relevant to temperature sensing were required to produce the zinc oxide nanorods used to sense ethanol.

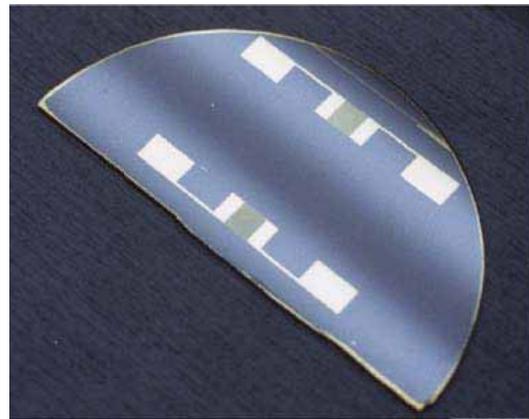


Figure 2. A completed SAW device on LiNbO_3

4. Experimental Procedure

In order to investigate the effect of temperature on the frequency response of the ethanol sensor, frequency response measurements were made over a range of temperatures between 35 °C and 270 °C.

In order to accurately measure the temperature of the sensor, a small thin film platinum resistance thermometer was attached directly to the substrate wafer, which was heated using a precision hot plate. To avoid connection problems resulting from the corrosion of conventional tinned copper wire at high temperature, all connections to the ethanol sensor were made using gold wire.

In order to measure the frequency response of the device, a function generator was used to apply a sine wave voltage of known frequency and amplitude to the terminals of the input IDT. The terminals of the input IDT were also connected to one channel of an oscilloscope. The terminals of the output IDT were connected to the second channel of the oscilloscope. Comparison of the amplitude and phase of the oscilloscope waveforms allows the calculation of the gain and phase response at that frequency. In order to rapidly make measurements at a large number of frequencies, computer software was developed to automate the measurement process.

5. Results and Discussions

The propagation velocity of the SAW was found to be highly influenced by temperature. This is investigated by recording the frequency response of the device, sample 4B, at different surface temperature. Figure 3 shows the frequency response near the resonance peak at temperatures from 35 to 110°C. The results indicate that as the temperature increases, the frequency response shifted to the left and the attenuation increased.

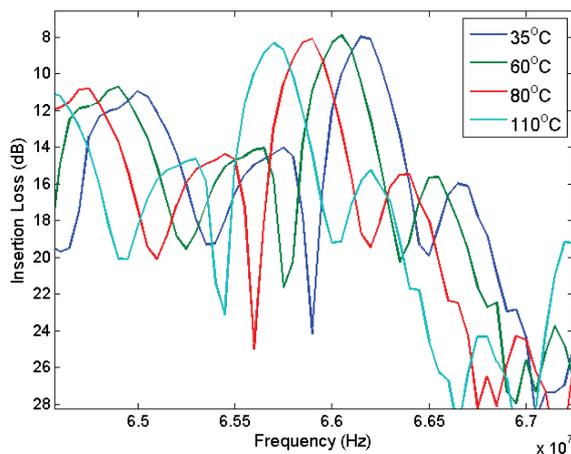


Figure 3. Frequency response at different temperatures

These changes in centre frequency correspond to a change in SAW velocity, that is;

$$\frac{\Delta V_R}{V_R} \propto \frac{\Delta f}{f} \quad (2)$$

The SAW velocity can be influenced by change in mass (m), temperature (T), conductivity (σ), permittivity (ϵ), pressure (p), etc [3]. Therefore the change in velocity can be stated as;

$$\Delta V_S = \frac{\partial V_S}{\partial m} \Delta m + \frac{\partial V_S}{\partial T} \Delta T + \frac{\partial V_S}{\partial \sigma} \Delta \sigma + \frac{\partial V_S}{\partial \epsilon} \Delta \epsilon + \frac{\partial V_S}{\partial p} \Delta p \dots \quad (3)$$

And the frequency shift due to temperature changes can be expressed using a polynomial expansion [4];

$$f_0(T) = f(T_0)[1 + a(T - T_0) + b(T - T_0)^2 + c(T - T_0)^3 + \dots] \quad (4)$$

Therefore, SAW temperature sensors would require a large value for a and low values for b, c coefficients for good linearity. Figure 4 shows the temperature sensitivity response of the SAW device. Assuming the 3rd or higher order terms are insignificant; a quadratic polynomial fit was constructed to estimate the parameters a and b . Parameter a was found to be $-6.40 \text{ kHz}/^\circ\text{C}$ while parameter b was estimated to be $5.79 \text{ Hz}/^\circ\text{C}^2$, which is much smaller than the value of a . This result proves the outstanding temperature sensing characteristic of SAW device on LiNbO_3 substrate for temperature range within 35 to 220°C.

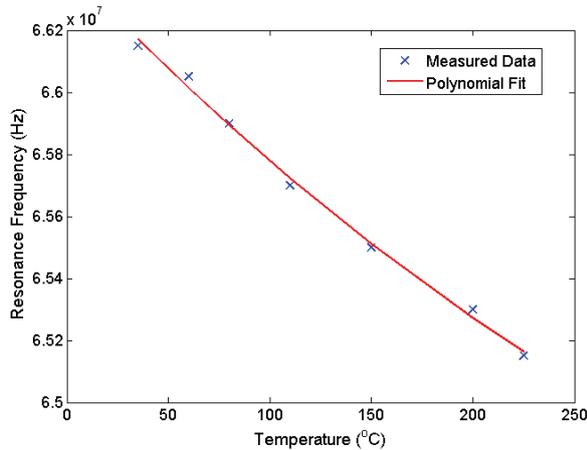


Figure 4. Temperature sensitivity plot

SAW temperature sensing behaviour based on LiNbO_3 was investigated in [3] and [5], the sensitivity was found to be -6.0 and -8.8 $\text{kHz}/^\circ\text{C}$ respectively.

For temperatures above 220°C , the frequency shift and insertion loss increases dramatically. This is explained by the increasing decomposition of LiNbO_3 at high temperature environment [6, 7].

6. Conclusion

- The fabrication process of the SAW device has been successfully developed.
- A temperature sensitivity of -6.40 $\text{kHz}/^\circ\text{C}$ was found in a temperature range of 35 to 220°C with outstanding linearity.
- The surface acoustic wave ethanol sensor is temperature sensitive and can be used as a temperature sensor.

Acknowledgements

The authors would like to thank Dr. Michael Leung for all the guidance and support provided throughout the project.

References

1. Morgan, D.P. (1973) Surface Acoustic Wave Devices and Applications. *Ultrasonics*, May 1973, 121-131.
2. AZ Photoresist Product Data Sheet (2009). MicroChemicals GmbH.
3. Ling, M., Li, H. (1993) SAW Temperature and Humidity Sensor with High Resolution. *Sensor and Actuators B*, **12**, 53-56.
4. Hauden, D., Jaillet, G., Coquerel, R. (1981) Temperature Sensor Using SAW Delay Line. *Ultrasonics Symposium*, **81**, 148-151
5. Ricco, A.J., Martin, S.J., Zipperian, T.E. (1985) Surface Acoustic Wave Gas Sensor Based On Film Conductivity Changes. *Sensors and Actuators*, **8**, 319-333
6. Steindl, R., Pohl, A., Reindl, L., Hornsteiner, J., Riha, E., Seifert, F. (1999) Passive Surface Acoustic Wave Sensors for Temperature and Other Measurands. *Tempmeko Proceeding*, 99, 424-429
7. Hauser, R., Reindl, L., Biniash, J. (2003) High-Temperature Stability of LiNbO_3 Based SAW Devices. *IEEE Ultrasonics Symposium*, 2003, 192-195

Development and Control of a Parallel Robot for Neurosurgery Applications

Josheel Pran Lal, Charlie Niu
The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract:

A significant contribution to the cost of neurosurgical interventions lies in the setup time required for traditional methods of manually orienting neurosurgical tool holders. The use of an automatic method by which a surgeon may orient their tools would yield significant reductions in patient and surgeon stress, patient preparation time as well as a reduction in surgery complication rate. This article investigates the application of a Stewart-Gough platform into the field of neurosurgery, where it is developed to be an accurate and user friendly robotic assistant for neurosurgeons. Using MATLAB & Simulink with Real Time Interface (RTI), a test bench was successfully developed to accurately orient a tool piece to 1mm accuracy and repeatability. An interactive force control strategy was also proposed, developed and tested to provide a user friendly interface by which a surgeon may co-manipulate the tool holder which is mounted on the end effector of the parallel robot.

Keywords: *Stewart-Gough platform, Neurosurgery, Inverse Kinematics, PID Control, Medical Robot*

1. Introduction

Robot-assisted surgery has garnered increasing levels of research interest over the last few decades, especially in the field of neurosurgery where precision, accuracy and repeatability greatly assist the operation process. The Stewart platform consists of two plates joined by six independently actuated legs, with the base plate being immobile whilst the top plate can move with six degrees of freedom. A parallel robot such as the Stewart platform has higher accuracy, repeatability and rigidity characteristics compared to a robot of a serial architecture, hence could potentially be a viable robotic assistant for neurosurgeons.

Traditionally in a neurosurgical intervention a stereotactic frame is affixed to a patient's head and a tool holder is then oriented manually about this frame. It has been

shown in the literature that tool positioning setup takes up to 80% of total intervention time. [1] A reduction in intervention time can be achieved by developing a robot to automatically orient a tool holder for a neurosurgeon. Utilizing a robotic assistant would thereby reduce the cost of surgery, surgeon or patient stress, decrease risk of surgery complication and ultimately increase throughput of patients.

Our project aims to apply an interactive force control scheme to the Stewart platform and to provide a test bench for a neurosurgical robotic assistant. With six degree of freedom point to point tracking and interpolation routines already proven through previous years work; our system design will integrate multiple design solutions into a test bench upon which advanced control strategies can be developed.

2. System Implementation

A MATLAB & Simulink with RTI system model is first compiled into a program on a dSpace DS1104 I/O board. We then use a Graphical User Interface (GUI) to control I/O to the DS1104 board with the program ControlDesk. Using six analogue channels from the dSpace DS1104 I/O board, six linear electric actuators were controlled via AMC 128M motor control modules. Two DC power supplies (dual tracking) were used to supply 24V to each motor driver, which in turn amplify their output voltage by a factor of 1.2 – meaning that the actuators receive a full 12V when they are supplied with 10V from the DS1104 at full scale. Each motor has an in-built quadrature incremental encoder, all six of which are parsed into a serial stream via the encoder interfacing circuit, which are then fed back to the DS1104 board. The force sensor signal goes through a FUTEK proprietary amplification module as well as an interfacing circuit before finally also being fed back to the DS1104 board. For further clarification this is depicted in Figure 1.

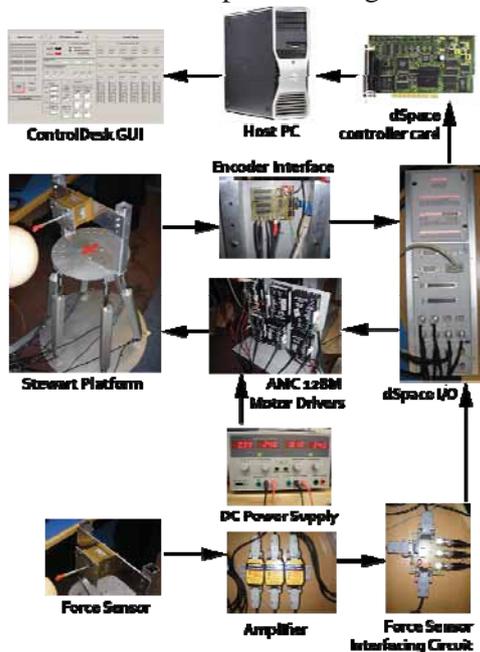


Figure 1. System implementation diagram

3. System Operation

The implemented system allows for four modes of pose control:

1. The Manual mode allows each actuator to be controlled independently.
2. The Point to Point mode takes a desired coordinate input and converts it into desired actuator extensions via inverse kinematics. These extensions are then passed as set points into a PID position controller affecting each actuator.
3. The Safe Trajectory Following mode takes care to avoid tool collision scenarios by safely interpolating around the patient's head. Initial and final coordinates are used as path start and end points and intermediate safe points are followed at a safe velocity.
4. An Interactive Force Control mode allows the surgeon to manipulate the tool holder position via a guiding force he or she imparts on a force sensor.

Inverse Kinematics for Position Control

For any position control, a kinematic solution of the structure is required. An inverse kinematic solution was used as it requires significantly less computation time compared to a forward kinematic solution [2]; as for a parallel mechanism there lies a unique solution for the leg lengths given a particular pose.

To better explain the inverse kinematics approach, consider the two coordinate systems, as shown in Figure 2. The world frame of reference W is set to the centre of the base plate, while the moving coordinate frame of the centre of the end effector is called P . The points where the leg structures are attached to the two plates are defined as α_i for the top plate, and b_i for the bottom plate, where $i = 1 \dots 6$.

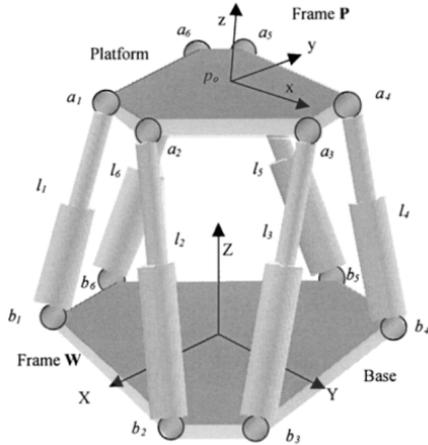


Figure 2. Depiction of world and end effector coordinate systems [2]

The moving frame P has a relative position and angular orientation vector with respect to the world coordinate frame. This is defined with the Cartesian space coordinate vector q in Equation 1.

$$q = [x \ y \ z \ \varphi \ \theta \ \psi] \quad (1)$$

The x - y - z coordinates of the end effector are transformed to the world coordinate system by a rotation matrix R_P (full definition located in [3]). Each leg attachment point then has a vector length from the world coordinate system W , given by

$$w_{a_i} = q_t + R_p a_i \quad (2)$$

The leg vector L_i is then obtained with the vector subtraction

$$L_i = w_{a_i} - b_i \quad (3)$$

where the leg length is the L2 norm of L_i . The inverse kinematics solution of leg lengths is thereby obtained for a given coordinate input. Subtracting the nominal leg length from L_i we find the required actuator extension. Once more, this is then used as the set point for the PID tracking controller for each actuator.

Interactive Force/Position Control Scheme

For safety and ethical reasons a robotic surgery assistant should not be allowed full control over a procedure. Whilst the robotic assistant may have superior positional accuracy compared to the surgeon; ultimate decision making during the procedure should remain with the surgeon. As such, there is the potential for a unique application of force sensing that allows the surgeon to have control over the surgical tool via application of a guidance force on the tool holder. The proposed and developed integrated control scheme for this is presented in Figure 3. In order to combine the decision making of capability of a surgeon and the accuracy of the Stewart platform, their control inputs are effectively doubly tiered – with the surgeon’s interactive or guiding control overriding the platform’s control effort should the need arise during the medical procedure. There is much worldwide interest in the successful application of this form of safe, user friendly control interface for surgeons.

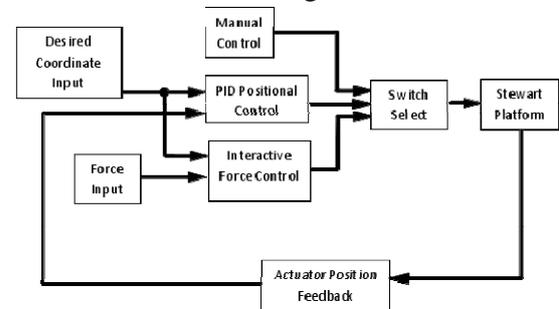


Figure 3. Depiction of IFPC scheme

Control System Integration

The positional controller used is capable of reaching a desired pose, and is able to follow a series of interpolated points. From within the positional control system it was desirable to be able to switch to an interactive force control system using an event handling methodology. This switching capability is also depicted in Figure 3. The system should be able to sense for

appropriate forces on the tool and respond accordingly.

For the case where the surgeon wishes to prevent any excessive forces from being generated by the tool – the force reading in the concentric axis of the tool is measured for a threshold value. This threshold value can be changed via the GUI. For instance, if the force sensor detects that the tool is applying too great a compressive force perpendicular to the skull, the sensor reading will trigger a detection sub-system that will cause the platform to move in the direction of the reaction force acting on the tool, thereby also on the force sensor.

For the case where the surgeon wishes to intervene and co-manipulate the tool position/orientation, once enabled, the position controller output will be overridden by the interactive force controller output. The interactive controller essentially sends position commands, emulating the position controller, depending on the direction of the force vector it experiences.

4. Experiments and Results

Tests were carried out to ascertain the performance characteristics of the system. The test bench was able to achieve an accuracy of 1mm and a repeatability of 1mm when in Point to Point mode, and in Safe Trajectory Following mode was able to follow to within 1mm of the calculated path. Figure 4 shows the typical actuator response when following such a safe trajectory.



Figure 4 Safe Trajectory Following actuator response

Testing of the interactive force control scheme yielded correct x and y axis response, however the motion response of the Z axis was not uniform due to non-linear combinations of motor backlash, friction and offsets.

5. Conclusions

- A neurosurgical robotic assistant test bench was designed, manufactured and developed to serve as a proof of concept for this application.
- Position control of the system was achieved to within 1mm accuracy and repeatability, for both Point to Point and Safe Trajectory Following modes.
- An Interactive Force/Position Control scheme was developed and allows for user friendly surgeon operation of the robotic assistant.

Acknowledgements

We would like to thank A/Prof Shane Xie for his continual support throughout the year as our supervisor. We would also like to express our gratitude to Logan Stuart, the Mechatronics Lab Technician, for his extraordinary expertise and assistance he lent us throughout the year.

References

1. Glauser, D.; Flury, P.; Villotte, N.; Burckhardt, C.W., "Conception of a robot dedicated to neurosurgical operations," *Advanced Robotics*, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on , vol., no., pp.899-904 vol.1, 19-22 Jun 1991.
2. Harib, K. and K. Srinivasan (2003) Kinematic and Dynamic Analysis of Stewart Platform-based Machine Tool Structures. *Robotica*. 21:p. 541-554.
3. S. Shah, K. Patel and S. Q. Xie (2008), "Simulation and control of a Stewart

platform for Long-bone fracture realignment surgery”, Final year project report, Mechanical Engineering Department, University of Auckland.