Ground Mobile Vehicle Velocity Control using Encoders and Optical Flow Sensor Fusion

Carlos Daniel Rodrigues Dionísio
carlos.dionisio@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

November 2016

Abstract

This work presents the wheels velocity estimation of a nonholonomic vehicle using a Kalman filter that uses measurements from the wheels encoders and an optical flow sensor. The estimated wheel velocities are then used in an inner loop to control the vehicle DC motors speed. This low level control is at the base of a better control of the vehicle velocity. With the technological advance, the mobile robots are increasingly used in many applications to help and protect the human life from taking unnecessary risks. For a mobile robot to move autonomously, it needs to know the environment around it and its location in the same environment. The location method often used is the dead-reckoning. This approach calculates the current position knowing the previous position, the movement direction, the speed and time difference. This type of localization is particularly useful in indoor environments where there is no GPS signal, sometimes being the only way to know a vehicle location. The encoders are one of the most used sensors in indoor localization measuring velocities, but its utilization brings problems in case of slippage, misleading the controller. Another sensor that measures velocities is the optical flow, which is not sensible to the slippage problem. Although it has its issues, its information, complemented with the one provided by the encoders, allows achieving a better estimation of the vehicle location. Therefore, using sensor fusion techniques such as Kalman filter, it is possible to combine the information provided by encoders and optical flow and get a better estimate of the vehicle speed and therefore increase the performance of the vehicles position control system. This work presents the estimation and speed control of DC motors using sensor fusion with information obtained from encoders and optical flow.

Keywords: Unmanned Ground Vehicle, Optical Flow, Sensor Fusion, Kalman Filter, Velocity Control, EGM30 motor, ADNS-3080, Encoders.

1. Introduction

There are many applications where the human presence may be inconvenient, dangerous or impossible, being nowadays increasingly substituted by unmanned vehicles. Unmanned vehicles are capable of moving in an unknown environment while executing specific tasks, such as exploration, surveillance, assistance, logistic services and environmental monitoring. [12] [6]

For these vehicles to move in unknown environments they are equipped with a set of sensors to observe the environment. With this information, they are able to autonomously make decisions about their behavior or to give this information to a remote human operator who will teleoperate the vehicle. Sensors accuracy is crucial for robots that operate in highly unpredictable environments.

In indoor applications, there is some limitation about the usable sensors. The Global Positioning System (GPS) is widely used in outdoor applications for vehicle positioning, but its signal is unavailable indoors or poor in dense urban areas. A common alternative used by ground vehicles is dead reckoning, an incremental solution that allows to know the current location knowing the previous location and the velocity over time.

An example of a dead-reckoning sensor is the encoder [14], that measures angular position or motion of a shaft, typically being aggregated to the wheels of the vehicle. Measuring the movement of the wheels, in case of slippage these sensors provide erroneous information relative to the wheels true displacement on the ground, and therefore an inaccurate vehicle velocity.

Another sensor that can measure velocities are the optical flow sensor. It obtains the vehicle longitudinal and lateral velocities by comparing the displacement of features in consequently acquired
images. Nowadays, these sensors are mainly used in unmanned aerial vehicles to solve the problem of GPS-denied environments [10]. Although not suffering from the slippage problem, being a vision-related sensor, optical flow is highly sensible to light conditions, as well as requiring that the acquired images constantly have identifiable features.

This work proposes a solution for the estimation of a vehicle wheels velocity based on a Kalman filter that uses measurements from incremental encoders and optical flow. In order to demonstrate the benefits of the wheel velocities estimation, a motor speed controller is designed. The performance of this low level controller will be a stepping stone to the performance of upper level controllers like vehicle velocity and position controllers.

In the remainder of this article, section 2 presents the considered ground vehicle and its respective components. The key features of the microprocessor, actuators and sensors will be presented. Section 3 introduces the mobile nonholonomic vehicle used, and the equations that relate the measurements of the encoders and of the optical flow sensor. Section 4 introduces the Kalman filter concept and its application to the vehicle wheels velocity estimation based on the measurements of encoders and optical flow. The design of the motors speed controller is presented in section 5, followed by section 6 with the experimental results obtained. Finally, section 7 closes with some concluding remarks and indication of future work.

2. Basic Components

To implement a control process sensors, actuators, and microcontrollers are used to collect, transmit and process the information, respectively.

2.1. Microcontroller

The microcontroller board utilized in this project is an Arduino Pro 3.3 V/8 MHz[1]. It was designed and manufactured by SparkFun Electronics and based on the ATmega328 (Figure 1a). It has 14 digital input/output pins, where six can be used as PWM outputs, six analog inputs, a battery power jack, a power switch, a reset button, and holes for mounting a power jack, an ICSP header, and pin headers. A six pin header was connected to a breakout board (figure 1b) to provide USB power and communication to the board.

The main reason for selecting this microcontroller was the 3.3 V circuit operating voltage necessary to connect to the optical flow sensor used in this work.

2.2. Actuation

The actuators are responsible for converting the control action signal computed by the microprocessor into a physical action applied to the system. The actuation used in this control loop it is com-

posed of two components: one MD25 board, making the interface between the microprocessor and the motors, and two EMG30 motors are responsible for the motion of the vehicle.

MD25

The MD25 board (Figure 2) was designed to work with EMG30 gear motors, and be able to drive two motors [3]. This electronic component is a Dual H-Bridge Motor Drive that enables a voltage to be applied across a load in either direction. It has two modes of communication Inter-Integrated Circuit (I2C) or Serial communication, and it is designed to work with a 12 V battery.

The communication used was I2C. It is possible to use two sets of values, 0 to 256 (mode 0) or -128 to 127 (mode 1). Mode 1 was used in this work, where the symbolization of the speed registers is exact velocity in the range of -128 (full reverse), 0 (stop) and 127 (full forward).

Analyzing the outputs of MD25 it is possible to conclude that the relation between the digital values and voltage can be considered linear. However, between the digital values -1 and 1 there is a dead zone nonlinearity. The equation that represents the
MD25 board is shown bellow:

\[
U'(V_{dig}) = \begin{cases} 
-10.9, & \text{if } V_{dig} \leq -128 \\
0.0878 \cdot V_{dig} - 0.2119, & \text{if } -128 < V_{dig} < -1 \\
0, & \text{if } -1 \leq V_{dig} \leq 1 \\
0.0845 \cdot V_{dig} + 0.3703, & \text{if } 1 < V_{dig} < 127 \\
10.9, & \text{if } V_{dig} \geq 127 
\end{cases}
\]

where \( U \) is the voltage produced by MD25 and \( V_{dig} \) a value between -128 and 127.

**EGM30 motors**

The EGM30 (Figure 3) assembles a 12 V DC motor with an encoder with 30:1 reduction gearbox [2]. It is excellent for small or medium robotic applications, providing cost-effective drive and feedback for the user. It also has a conventional noise suppression capacitor across the motor windings. The EGM30 specifications are shown in Table 1.

![Figure 3: EMG30 Motor.](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated Voltage</td>
<td>12 V</td>
</tr>
<tr>
<td>Rated Torque</td>
<td>1.5 Kg/cm</td>
</tr>
<tr>
<td>Rated Speed</td>
<td>170 rpm</td>
</tr>
<tr>
<td>Rated Current</td>
<td>530 mA</td>
</tr>
<tr>
<td>No load Speed</td>
<td>216 rpm</td>
</tr>
<tr>
<td>No load Current</td>
<td>150 mA</td>
</tr>
<tr>
<td>Stall Current</td>
<td>2.5 A</td>
</tr>
<tr>
<td>Rated Output</td>
<td>4.22 W</td>
</tr>
<tr>
<td>Encoder Resolution</td>
<td>360 counts/ shaft turn</td>
</tr>
</tbody>
</table>

Table 1: EMG30 Motor specifications.

A simplified schematic of the DC motor electrical and mechanical contributions is represented in Figure 4.

The DC motor may be modelled as a second-order system as:

\[
\dot{\theta}_{rotor} = \frac{K}{(J_s + B)(L_e s + R_e) + K^2} \]

Figure 4: Electric equivalent circuit of the armature and the free-body diagram of the rotor.

where \( \dot{\theta}_{rotor} \) is the angular velocity or rotor, \( K \) the motor constant, \( J \) the moment of inertia of the rotor, \( B \) the Motor viscous friction constant, \( L_e \) the electric inductance and \( R_e \) the electric resistance.

However, in many cases the model of motor is approximated by a first-order system, obtained neglecting the fast electrical dynamics:

\[
\frac{\dot{\theta}_{rotor}}{U} = \frac{K_m}{T_m s + 1}
\]

where \( K_m \) and \( T_m \) are the gain and time constant of motor, in this case equal to 1.92 and 0.1, respectively.

The EMG30 model parameters were obtained from [13] and are resumed in Table 2. These parameters were estimated directly from EMG30 output, and although gearbox reduction was not considered in the formulation, it is implicit in the parameters values.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>0.509 Nm/Amp</td>
</tr>
<tr>
<td>L_e</td>
<td>3.4e-3 H</td>
</tr>
<tr>
<td>R_e</td>
<td>7.101 Ω</td>
</tr>
<tr>
<td>B</td>
<td>0.000931 Nms</td>
</tr>
<tr>
<td>J</td>
<td>0.00377 kg * m²/s²</td>
</tr>
</tbody>
</table>

Table 2: EMG30 parameters

2.3. Sensors

This section describes the sensors used in this work, namely encoders and optical flow.

**Encoders**

Encoders are one of the most used sensors in industrial automation and mechatronics to measure motions. This electro-mechanical device converts linear or angular motions of a shaft or axle to an analog or digital code [11]. There are two models of encoders: absolute and incremental. The absolute encoder provide an numerical value for each angular position even over several revolutions, making angle transducers. On the other hand, incremental encoders generate a precisely defined number...
of pulses per revolution, which is usually processed into information such as speed, distance, and position. The encoders applied in this work are optical incremental encoders.

The digital value \( P(t) \) given by an incremental encoder at time \( t \), corresponds to the sum of pulse values read since the beginning of the movement. To know what is the distance traveled between two measurements it is necessary to subtract the previous value read to the current value. The linear velocity of a wheel is obtained from:

\[
V_{\text{wheel}} = \frac{\Delta P_{\text{wheel}}}{R_{\text{ENC}} \Delta t}
\]

where \( P_{\text{wheel}} \) corresponds to the perimeter of wheel, \( R_{\text{ENC}} \) the resolution of sensor (in this case 360), \( \Delta P \) the difference between two consecutive measurements and \( \Delta t \) to time difference between the measurements.

In order to simulate the EMG30 encoder we considered that the noise of sensors was white noise with power 0.01. Comparing the simulated encoder measurements in Matlab-Simulink with the real measurements shown in Figure 5, one verifies that the results are identical.

![Figure 5: Encoders: real vs. simulated measurements](image)

In normal conditions the ADNS-3080 gives three values, \( \Delta X \), \( \Delta Y \) and \( \text{Squal} \). The first two values provide displacements information of the flat surface under the sensor, and the third value gives information about the surface quality.

The values \( \Delta X \) and \( \Delta Y \) returned by the sensor give the average movement of surface features identified by the sensor since the last reading. To convert these measures into distance values it is necessary to know the altitude of the sensor, \( h \). If two sensors are moving the same distance but at different heights, for the sensor that moves at a lower altitude the surface features will appear to go further, and this will result in a higher sensor value (see Figure 7).

![Figure 7: Altitude effect on OF sensor value](image)

Optical Flow

Another kind of sensor that can be used to measure velocity is the optical flow (OF) sensor. This sensor uses a sequence of images and through it obtains information about the motion in X and Y direction.

Two optical flow sensors were tested and compared: The ADNS-3080 [4] (Figure 2a) manufactured by Avago Technologies and the PX4FLOW [5] developed by PX4 autopilot (Figure 2b).

The ADNS-3080 is a conventional OF mouse sensor that contains an internal low-resolution camera and a digital signal processor (DSP) programmed to estimate the relative displacement from micro-shadows of images acquired.

The key features that should be considered when choosing an optical flow sensor are frame rate, image size and resolution. The frame rate is the number of pictures that an OF sensor can take per second. For a given image sensor, one with a higher frame rate indicates its ability to detect small movements and consequently to identify higher speeds.

| Frame Rate | 2000 - 6469 FPS |
| Image Size | 30 x 30 Pixels |
| Resolution | 400 or 1600 CPI |
| Communication | SPI |

Table 3: Mainly characteristics of ADNS-3080
of the sensor and a Scalar that is related to the value given by the sensor for a single pixel moved:

\[
Dist_{X/Y} = \frac{\Delta X/Y}{R_{OF} \times Scalar} \times h \times 2 \times \tan \left( \frac{FOV}{2} \right)
\]

(5)

where \( R_{OF} \) is the resolution of sensor (in this case 30).

The OF sensor also provides information about the surface quality. This characteristic is given by the Squal value, and it is a measure of 1/4 of the number of strong features visible by the sensor in the current frame. Its maximum value is 169.

Some known issues of this sensor are that it only works in well-lit environments and rotating the sensor will lead to erroneous readings.

The PX4FLOW is an optical flow smart camera, and for many application, it is much better than the OF sensors based on an optical mouse sensor. This sensor has a 752x480 pixels resolution and calculates optical flow at 250 Hz over an area of 64x64 pixels. Unlike the mouse based sensors, this OF sensor does not need good light conditions to give successful results. Another advantage is that it can be reprogrammed to do any other computer vision task using the 752x480 pixel camera.

This sensor has aggregated to the camera a gyroscope and sonar. With these two sensors it is possible to know in real time the height, the roll and pitch angles of the vehicle and compensate its effects in distance calculation.

Comparing ADNS-3080 and PX4FLOW specifications it is possible to conclude that PX4FLOW is better than ADNS-3080, mainly in aerial applications because of changes in height, roll and pitch. However PX4FLOW is much more expensive than ADSN-3080, costing about 150 € against the 15 € of the ADSN-3080. For this reason and because ground applications do not have significant changes in height, roll or pitch, the sensor chosen was the ADNS-3080.

### ADNS-3080 Application

To convert the values provided by the OF sensor into velocities some parameters as FOV and Scalar should be determined. The FOV, in case of optical instruments or sensors, is the solid angle through which a detector is sensitive to electromagnetic radiation.

Knowing the sensor’s height and using a picture with known features dimensions, using trigonometric relations it is possible to calculate the FOV from:

\[
FOV = 2 \times \beta = 2 \times \tanh \left( \frac{5.4}{21.8} \right) = 0.4855 \ [\text{rad}]
\]

Initially, the ADNS-3080 values were always zero, and since this sensor was developed to well-lit outdoor environments, the problem was associated to poor light condition. To improve the image acquired by the ADNS-3080 a lighting system developed it was composed of two red LEDs with high-brightness and a structure that aligns the center of the light emitted by the LEDs and the center of the sensor’s camera, see Figure 9.

Comparing the Squal value given by the sensor with and without lighting system, the mean increases to 32 for the former, as seen in Figure 10. With this lightning system it was possible to start getting data from the ADNS-3080.

On the other hand, the Scalar is the parameter that relates the number of pixels that moved from one picture to the next one and the value given by the sensor. This value is obtained comparing the velocities provided by the optical flow sensor with the velocities obtained from another system that can be considered as truth.

The Laboratory of Control, Automation and Industrial Informatics of Instituto Superior Técnico has a motion capture system that allows calculating the velocity of any object using markers. Considering this system measurements as truth with the Figure 8: Field of view determination.
ones provided by the OF system it is possible to obtain the Scalar parameter value.

Although the Scalar increases with the light system and the sensor starts giving better information, the floor where the motion capture system is located does not have the perfect conditions to acquire OF data because it is polished and with few features, see Figure [11].

![Figure 11: Surface covered by motion capture system.](image)

In order to confirm that the surface indeed influenced the quality of the optical flow sensor measurements, the sensor was tested in other surfaces found in indoor and outdoor environments. First we get the true velocity of the vehicle with motion capture system, and next we test, in different floors, with the vehicle at the same velocity, and compare the results with the values obtained with the motion capture (MoCap) system. The results are shown in Table [4].

<table>
<thead>
<tr>
<th>Floor</th>
<th>Mean of $V_x$</th>
<th>$\sigma_x$</th>
<th>Mean of $V_y$</th>
<th>$\sigma_y$</th>
<th>Mean of Scalar</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoCap system</td>
<td>-0.66</td>
<td>7.73</td>
<td>98.03</td>
<td>77.94</td>
<td>32</td>
</tr>
<tr>
<td>Floor 1</td>
<td>-0.62</td>
<td>34.85</td>
<td>159.62</td>
<td>34.85</td>
<td>38</td>
</tr>
<tr>
<td>Floor 2</td>
<td>-1.11</td>
<td>8.69</td>
<td>82.10</td>
<td>75.73</td>
<td>50</td>
</tr>
<tr>
<td>Floor 3</td>
<td>-0.03</td>
<td>3.02</td>
<td>167.39</td>
<td>22.82</td>
<td>55</td>
</tr>
<tr>
<td>Floor 4</td>
<td>-0.23</td>
<td>9.16</td>
<td>165.75</td>
<td>29.15</td>
<td>78</td>
</tr>
<tr>
<td>Floor 5</td>
<td>-1.92</td>
<td>14.32</td>
<td>75.53</td>
<td>81.32</td>
<td>19</td>
</tr>
<tr>
<td>Floor 6</td>
<td>-0.40</td>
<td>5.72</td>
<td>156.95</td>
<td>22.46</td>
<td>80</td>
</tr>
<tr>
<td>Floor 7</td>
<td>-0.55</td>
<td>8.59</td>
<td>67.77</td>
<td>77.77</td>
<td>27</td>
</tr>
<tr>
<td>Floor 8</td>
<td>0.23</td>
<td>3.05</td>
<td>0.02</td>
<td>4.63</td>
<td>33</td>
</tr>
</tbody>
</table>

Table 4: Values of ADNS-3080 measurements at different floors.

Analyzing the results, it is possible to conclude that the best floors are Floor 1, Floor 3, Floor 4 and Floor 6. Although Floor 1 presents good values in velocities, the Scalar value is very low compared with the others. One possible reason is because floors 3, 4 and 6 are outdoor environments with good light and Floor 1 is an indoor environment in the same location of the motion capture system.

To simulate OF sensor was considered the noise as white noise with noise power of 0.1. The power noise like encoders, was found comparing true and simulated measurements (see Figure [13]).

3. Nonholonomic Mobile Vehicle

Nonholonomic systems are characterized by having a finite dimension where some restriction is imposed in one or more states of the system. An example of this systems is the differential robot because are allowed to reach any point in the state space but at a certain time or state there are constraints imposed on the motion (nonholonomic constraints). This section presents the mobile vehicle platform used and the relations between wheels and vehicle velocities.

3.1. Platform

The differential wheeled robot used in this work is shown in Figure [14]. This type of mobile robot is characterized by having the movement based on two separately driven wheels placed on each side of the
robot body. To change the vehicle direction it is necessary to modify the relative angular velocity of its wheels and hence it does not require an additional steering motion. Normally wheels or casters are added to balance the robot. This vehicle has one caster.

Figure 14: Differential wheel mobile robot platform

3.2. Odometry
Odometry is the process of using data from a motion sensor to estimate the change in position over time. It is utilized by some wheeled robot to estimate their position from a starting location. This method has the problem of achieving position estimation by integration of velocity measurements making it sensitive to errors.

The objective of this work is to develop a low-level velocity control system. In the following, the sensors used and the respective measurements are described.

Encoders
The encoders give the velocity of each wheel, but from these measurements it is possible to obtain the velocity of the vehicle. Considering the point $O$ as the center of geometry (see Figure 15), and $\omega_R$ and $\omega_L$ the angular velocities given by right and left encoders, respectively, it is possible to relate these values with the linear and angular velocities of point $O$.

\begin{equation}
V = (\omega_r + \omega_l) \frac{r}{2} \tag{6}
\end{equation}

where distances $b$ and $r$ were approximately measured as 115 mm and 50 mm, respectively.

Optical Flow
Equivalently to what was done for the encoders measurements, it is possible to obtain the velocity of each wheel from the vehicle velocities measurements from the optical flow sensor.

Figure 16: Optical Flow odometry nonholonomic vehicle.

Observing Figure 15 one obtains the following relations for the linear and angular velocities of point $O$:

\begin{equation}
V = -V_Y \tag{8}
\end{equation}
\begin{equation}
\dot{\theta} = l \star V_X \tag{9}
\end{equation}

where $V_Y$ and $V_X$ are the lateral and longitudinal velocities measured by the optical flow sensor. Using Eqs (6) and (7) it is possible to relate the velocities $V_X$ and $V_Y$ with the angular velocity of each wheel using the following equations:

\begin{equation}
\omega_r = \frac{r}{2 \star b \star l} \tag{10}
\end{equation}
\begin{equation}
\omega_l = -\frac{-r}{2 \star b \star l} \tag{11}
\end{equation}

The distance $l$, between the optical flow and the center of geometry is approximately 115 mm.

4. Kalman Filter Estimation
The Kalman filter is an algorithm that uses the dynamic model of the system and a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables. This estimation tends to be more precise than those based on a single measurement alone, by using Bayesian inference and estimating a joint probability distribution over the variables for each timeframe. [8]

The Kalman filter algorithm has two distinct set of equations, namely Time Update and Measurement Update.
Time Update (prediction)
\[ \hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \]  
(12)

\[ \hat{P}_k^- = A\hat{P}_{k-1}A^T + Q \]  
(13)

Measurement Update (correction)
\[ K_k = P_k^-C^T(CP_k^-C^T + R)^{-1} \]  
(14)

\[ \hat{x}_k = \hat{x}_k^- + K_k(z_k - C\hat{x}_k^-) \]  
(15)

\[ P_k = (I - K_kC)P_k^- \]  
(16)

where \( A \) is the state transition model which is applied to the previous state \( x_{k-1} \), \( B \) is the control input model which is being implemented to the control vector \( u_k \), \( w_k \) is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance \( Q \), \( C \) is the observation model which maps the actual state space into the observed space and \( v_k \) is the observation noise which is assumed to be zero mean Gaussian white noise with covariance \( R \).

Since the measures are discrete, the discrete first order model of the motors is used, obtained using Tustin approximation. The sample time was 0.025 seconds because it is the time the microcontroller needs to make the entire loop of control. The matrix \( R \) has as diagonal the variance of velocities given by encoders and optical flow in (mm/s)^2. Matrix \( Q \) is used as a tuning parameter to adjust the gain of the Kalman filter to smooth more or less the data. After some tests, the matrices were set as:

\[ Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \]  
(17)

\[ R = \begin{bmatrix} 422,863 & 0 & 0 & 0 \\ 0 & 333,9581 & 0 & 0 \\ 0 & 0 & 319,9581 & 0 \\ 0 & 0 & 0 & 319,9581 \end{bmatrix} \]  
(18)

5. Motors Velocity Control

Finally, to complete the motors velocity control loop, in this chapter we will be presented the controller implemented. In a simple way, a controller is hardware device or a software program that manages or directs the flow of information between two devices. What a controller does is calculate the difference between the reference value and process value, the error \( e(t) \), and applies a correction with the objective of determinate the input of the process, \( u(t) \). The motors velocity control has the objective of control the angular velocity of each motor (low-level control) using the information estimated by the Kalman Filter presented in section 4, Figure 5.1.

In order to design the low-level controller, performance requirements were specified. The controlled response should not have steady state error, small overshoot tolerated and the settling time should be less than 0.5 seconds.

Proportional-Integral-Derivative (PID) controller family was chosen for the low-level controller. Its digital form is given by:

\[ u(n) = u(n - 1) + q_0e(n) + q_1e(n - 1) + q_2e(n - 2) \]  
(19)

where \( e(n) \) is the difference between the reference and the estimated value at instants \( n \), and \( u(n - 1) \) is the previous control signal. The method used to implement the digital controller was Tustin method because this method ensures the stability of the approximation. The constants \( q_0, q_1 \) and \( q_2 \) are given by:

\[ q_0 = K_P\left(1 + \frac{T_d}{T_0} + \frac{T_0}{2T_i}\right) \]  
(20)

\[ q_1 = -K_P\left(1 + \frac{2T_d}{T_0} - \frac{T_0}{T_i}\right) \]  
(21)

\[ q_2 = K_P\frac{T_d}{T_0} \]  
(22)

where sampling time \( T_0 \) is the same used in the Kalman filter implementation, \( T_i, T_d, \) and \( K_P \) correspond to the integrative time, derivative time and proportional gain of the PID controller, respectively. To tune the PID parameters was studied the possibility of using one of Ziegler-Nichols tuning methods was considered, but considering the motors as first-order systems made it impractical.

The next step was the controller manual tuning. Initially, the simulator was used to find an approximation of the PID parameters and subsequently the parameters were fine tuned in the vehicle control implementation.
Firstly, the integral and derivative gain were set at zero and the proportional gain was increased. However, although increasing the proportional gain decreases the steady-state error, it also increases the noise. Therefore, the steady-state error cannot be corrected by the proportional gain alone.

Secondly, the proportional controller term was fixed at 40 and the integrative term was increased. The integrative term together with the proportional remove the steady-state error, but on the other hand, the overshoot appear and the signal noise increase.

Finally, we decrease the proportional gain and test different times of integration. The controlled motor velocity results are shown in Figure 18.

![Figure 18: Real motor angular velocity closed-loop system with PI controller.](image)

Comparing the performance of each controller, the controller that better respect the defined specifications is the controller with the proportional gain equal to 4 and the time of integration equal to 0.05 seconds. The control error is associated with sensors measurement errors.

6. Results

In order to evaluate the performance of each sensor and of the Kalman filter estimation, the respective signal were compared with the measurements obtained by the motion capture system. In order to do so, digital references were given to the MD25 board, in open loop. The linear and angular velocities obtained by each of the systems were compared. The mean squared error (MSE) and standard deviation results, obtained relative to the motion capture system true measurements, are presented in Table 7.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>MSE</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoders</td>
<td>0.0370</td>
<td>1.9504</td>
</tr>
<tr>
<td>Optical Flow</td>
<td>0.0137</td>
<td>2.0948</td>
</tr>
<tr>
<td>Estimator</td>
<td>0.0130</td>
<td>2.0438</td>
</tr>
</tbody>
</table>

Table 5: MSE [(rad/s)^2] and standard deviation in open loop.

Observing the results it is possible to conclude that the estimator has the best result, with lower MSE, compared with the sensors.

In order to validate the performance of the low-level motor velocities control, linear and angular speed and convert into the reference angular velocities for each wheel before entering the control loop. As a result, the control loop tested is presented in Figure 20.

![Figure 20: Low-level motor velocity control with velocities conversion.](image)

To convert the linear and angular velocity of vehicle into angular velocity of each wheel, the following relations were used, obtained inverting eqs. (6)-(7):

\[ \omega_r = \left( V + \dot{\theta} \ast b \right) \ast \frac{1}{r} \]  

\[ \omega_l = \left( V - \dot{\theta} \ast b \right) \ast \frac{1}{r} \]  

In order to validate the controller performance, the vehicle was requested to perform a circle at different angular velocities. The obtained results for each motor velocity and angular velocity of vehicle are represented in Figure 21 and 22. The vehicle angular velocity has a MSE equal to 0.2679 (rad/s)^2 when compared with the motion capture system true values.

7. Conclusion and Future Work

After this study, it is concluded that the optical flow can be used to solve the wheels slippage problem, but at the same time, it presents some issues with the floor characteristics. It needs to have good light conditions the floor surface must have enough features to be detected by the sensor. This problem
Figure 21: Motors angular velocity with inner-loop control.

Figure 22: Vehicle angular velocity with inner-loop control.

can be justified by the low resolution of the optical flow camera.

The sensor fusion between encoders and optical flow presented results that can be useful to solve the problem of slippage and poor floor characteristics. It is possible to conclude that sensor fusion produces better results than encoders or optical flow alone.

As future work, some improvements can be made. The first one is to change the MD25 model used in order to include the battery model in the simulator, since the battery level greatly influences the motors performance. The seconds to improve the optical flow sensor measurements by modifying the lens of the sensor to one with under magnification and performing tests with the sensor at different heights. The third is to implement the high-level loop control.

References


