Date of current version June 22, 2020.

Digital Object Identifier PLACEHOLDER

EEET 4075 Mechatronic System Design 2 - Project Report

MICHAEL J. DUKE[1]

[1]University of South Australia, Mawson Lakes, SA 5095 Australia (e-mail: dukmj002@mymail.unisa.edu.au)

This work was supported in part by the University of South Australia

ABSTRACT These instructions give you guidelines for preparing papers for IEEE Access. Use this document as a template if you are using LaTeX. Otherwise, use this document as an instruction set. The electronic file of your paper will be formatted further at IEEE. Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title; short formulas that identify the elements are fine (e.g., "Nd–Fe–B"). Do not write “(Invited)” in the title. Full names of authors are preferred in the author field, but are not required. Put a space between authors’ initials. The abstract must be a concise yet comprehensive reflection of what is in your article. In particular, the abstract must be self-contained, without abbreviations, footnotes, or references. It should be a microcosm of the full article. The abstract must be between 150–250 words. Be sure that you adhere to these limits; otherwise, you will need to edit your abstract accordingly. The abstract must be written as one paragraph, and should not contain displayed mathematical equations or tabular material. The abstract should include three or four different keywords or phrases, as this will help readers to find it. It is important to avoid over-repetition of such phrases as this can result in a page being rejected by search engines. Ensure that your abstract reads well and is grammatically correct.

INDEX TERMS Enter key words or phrases in alphabetical order, separated by commas. For a list of suggested keywords, send a blank e-mail to keywords@ieee.org or visit http://www.ieee.org/organizations/pubs/ani_prod/keywrd98.txt

I. INTRODUCTION

MULTIRATE Extended Kalman Filters and absolute versus relative (dead reckoning) localisation.

II. RELATED WORKS

[1] has infor on multirate, multisensor data fusion. Extracted info includes being able to cascade EKFs into each other indefinitely to allow for many sensors and rates to be included. [2] is more for lidar and imu based navigation, used for comparison more than anything. [3] specifies different ways to fuse delayed measurements together. [4] explains the specific method in detail from its creator. [5] has information regarding beacon based navigation and triangulation.

Several different ways to combine the estimations of different sensors, difficulties with multi-rate systems. Variants or modifications on EKF and UKF, compare and contrast, computational cost, relative accuracy, limitations. maybe compare to particle filters as well? Need for an initial estimate of position and heading. Beacon based navigation.

Using Alexander’s Method for fusion of absolute and dead reckoning system due to update rate of LiDaR being much slower than that of odometry and IMU.

III. METHODOLOGY

A. SYSTEM MODEL

As with most things in control theory, the first step is to determine the state transition model of the system. In the case of the TurtleBot, a lot of the lower level control over the robot is controlled directly by the servomotors themselves. The only variables that are able to be controlled are linear velocity in the x-direction of the bot’s local frame (v), and angular velocity around the z-axis of the bot’s local frame (ω). As such, the model in 1 was developed, where T is the sample period. This model uses a few assumptions and holonomic constraints to keep the model simple.

- The robot cannot move sideways or vertically.
- The wheels will never slip.
- The effects of inertia are negligible.

$$\boldsymbol{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + Tv\cos(\theta_{k-1}) \\ y_{k-1} + Tv\sin(\theta_{k-1}) \\ \theta_{k-1} + T\omega \end{bmatrix} \tag{1}$$

With the non-linear system model created, the Jacobian in 2 was calculated to be used for gain and error calculation by the EKF.

$$\boldsymbol{F}_k = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x_{k-1}}} = \begin{bmatrix} 1 & 0 & -Tv\sin(\theta_{k-1}) \\ 0 & 1 & Tv\cos(\theta_{k-1}) \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

B. DEAD RECKONING

With the state transition model taken care of, the observation model can be determined. In the case of this EKF, there is two observation models which will be referred as the primary observation model, which includes all of the faster dead reckoning measurements, and the secondary observation model, which includes the slower, absolute measurements.

The primary observation model in this implementation includes information taken from the *joint_states* topic and the *imu* topic. The *joint_states* topic has information regarding the position, velocity, and torque of each wheel. For the observation model, only the wheel positions (s_l and s_r) are used. The *imu* topic has information regarding the linear acceleration, angular velocity, and magnetic field orientation of the bot. For the observation model, only the angular velocity around the z axis was used, while acceleration was considered initially, due to the noise, and need for double integration for it to be used, it was not used. for s_l and s_r to be usable, a small amount of processing is used to convert the information to the change in distance traveled (δs_k) and the change of yaw ($\delta\theta_k$), this is reflected in 3 and 4 and visualized in figure 1.

$$\delta s_k = \frac{(s_{r,k} - s_{r,k-1} + s_{l,k} - s_{l,k-1})\, r}{2} \tag{3}$$

$$\delta \theta_k = \frac{(s_{r,k} - s_{r,k-1} + s_{l,k} - s_{l,k-1})\, r}{b} \tag{4}$$

Where b is the wheelbase and r is the radius of the wheels, both in meters.

$$b = 0.287 \tag{5}$$

$$r = 0.033 \tag{6}$$

The angular velocity measurement from the *imu* topic (ω_k) can be used directly. Similar to the state transition model, both the non-linear model and the Jacobian of that model is needed for the EKF, which are presented in 7 and 8 respectively.

$$\boldsymbol{h}_k = \begin{bmatrix} \delta s_k \\ \delta\theta_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2} \\ \theta_k - \theta_{k-1} \\ \frac{\theta_k - \theta_{k-1}}{T} \end{bmatrix} \tag{7}$$

$$\boldsymbol{H}_k^{\top} = \begin{bmatrix} \frac{x_k - x_{k-1}}{\sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}} & 0 & 0 \\ \frac{y_k - y_{k-1}}{\sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}} & 0 & 0 \\ 0 & 1 & \frac{1}{T} \end{bmatrix} \tag{8}$$

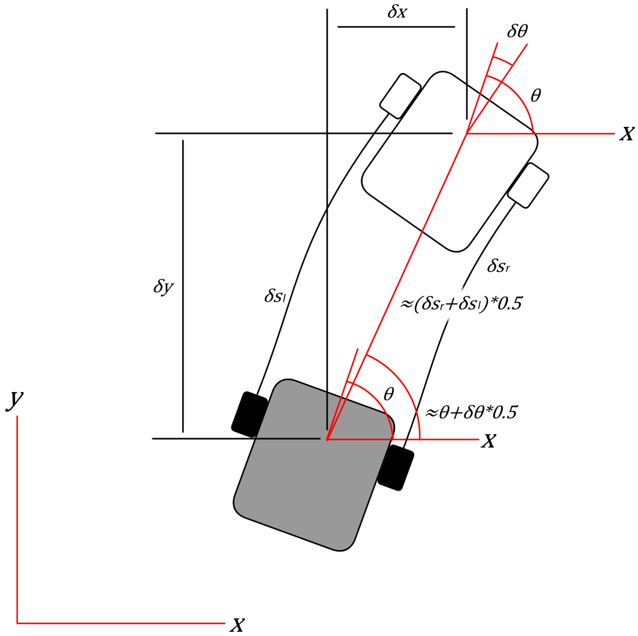


FIGURE 1: Geometric estimation of $\delta\theta$, δx, and δy based off δs_r and δs_l.

C. ABSOLUTE

The secondary measurement was implemented as a beacon based navigation system with a method similar to [5]. The main difference being that the number of beacons being measured is limited to only two. This is done to reduce the computation time at the expense of accuracy. The method used also does not allow for a definite single estimation position, but a pair of them. With the appropriate assumptions and control measures in place though, one of the positions can always be ruled out.

To estimate the orientation of the turtlebot, the angle to the closest beacon is used as a correction.

To estimate the position of the turtlebot, the intersection of two circles is calculated knowing their center coordinates and radii.

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 &= r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 &= r_2^2 \end{aligned} \tag{9}$$

Rearranging, and with the assumption that $x_1 = x_2$ in both cases we get

$$y = -\frac{r_1^2 - r_2^2 - y_1^2 + y_2^2}{2(y_1 - y_2)} \tag{10}$$

$$x = \pm\frac{\sqrt{\beta_1 \beta_2}}{2(y_1 - y_2)} \tag{11}$$

$$\beta_1 = (r_1 + r_2 + y_1 - y_1)(r_1 + r_2 - y_1 + y_1) \tag{12}$$

$$\beta_2 = (r_1 - r_2 + y_1 - y_1)(-r_1 + r_2 + y_1 - y_1) \tag{13}$$

As we know the fence around which the bot cannot escape, one of the possible positions can be eliminated, leaving

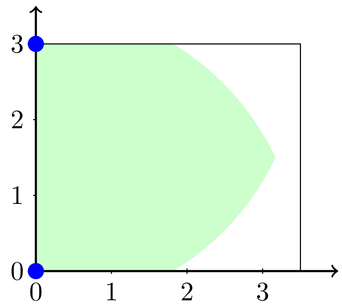


FIGURE 2: Left beacons showing measurable locations in green.

BLAH if using the left beacons, and BLAH if using the right beacons.

$$x = x_1 + \frac{\sqrt{\beta_1 \beta_2}}{2\left(y_1 - y_2\right)} \tag{14}$$

$$x = x_1 - \frac{\sqrt{\beta_1 \beta_2}}{2\left(y_1 - y_2\right)} \tag{15}$$

$$\boldsymbol{h}_k = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \tag{16}$$

$$\boldsymbol{H}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

D. EKF

a priori estimate

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}_{k-1}\left(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}\right) \tag{18}$$

$$\boldsymbol{P}_k^- = \boldsymbol{F}_{k-1} \boldsymbol{P}_{k-1}^+ \boldsymbol{F}_{k-1}^\top + \boldsymbol{Q}_{k-1} \tag{19}$$

Kalman gain update,

$$\boldsymbol{K}_{1,\,k} = \boldsymbol{P}_k^- \boldsymbol{H}_{1,\,k}^\top \left(\boldsymbol{H}_{1,\,k} \boldsymbol{P}1,\,k^- \boldsymbol{H}_{1,\,k}^\top + \boldsymbol{R}_{1,\,k}\right)^{-1} \tag{20}$$

a posteriori estimate,

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_{1,\,k}\left(\boldsymbol{y}_{1,\,k} - \boldsymbol{h}_{1,\,k}\right) \tag{21}$$

$$\boldsymbol{P}_{1,\,k}^+ = \left(\boldsymbol{I} - \boldsymbol{K}_{1,\,k} \boldsymbol{H}_{1,\,k}\right) \boldsymbol{P}_{1,\,k}^- \tag{22}$$

When the secondary measurement is received, the kalman gain, error, and state estimate are updated

$$\boldsymbol{K}_{2,\,k} = \boldsymbol{P}_{1,\,k}^- \boldsymbol{H}_{2,\,k}^\top \left(\boldsymbol{H}_{2,\,k} \boldsymbol{P}_{1,\,k}^- \boldsymbol{H}_{2,\,k}^\top + \boldsymbol{R}_{2,\,k}\right)^{-1} \tag{23}$$

$$\boldsymbol{P}_{2,\,k}^+ = \left(\boldsymbol{I} - \boldsymbol{K}_{2,\,k} \boldsymbol{H}_{2,\,k}\right) \boldsymbol{P}_{1,\,k}^- \tag{24}$$

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_{1,\,k}\left(\boldsymbol{y}_{1,\,k} - \boldsymbol{h}_{1,\,k}\right) + \boldsymbol{K}_{2,\,k}\left(\boldsymbol{y}_{2,\,k} - \boldsymbol{H}_{2,\,k}\hat{\boldsymbol{x}}_k^-\right) \tag{25}$$

If the secondary measurement is delayed by $s+N$ samples, where s is the sample with which the measurement refers to, then the recalculation adds one more term to account for the delay.

$$\begin{aligned} \hat{\boldsymbol{x}}_k^+ =& \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_{1,\,k}\left(\boldsymbol{y}_{1,\,k} - \boldsymbol{h}_{1,\,k}\right) \\ &+ \boldsymbol{W}\boldsymbol{K}_{2,\,s}\left(\boldsymbol{y}_{2,\,k} - \boldsymbol{H}_{2,\,s}\hat{\boldsymbol{x}}_s^-\right) \end{aligned} \tag{26}$$

where

$$\boldsymbol{W} = \prod_{i=1}^{i=N} \left(\boldsymbol{I} - \boldsymbol{K}_{s+i}\boldsymbol{H}_{1,\,s+1}\right)\boldsymbol{F}_{s+i-1} \tag{27}$$

IV. RESULTS AND DISCUSSION

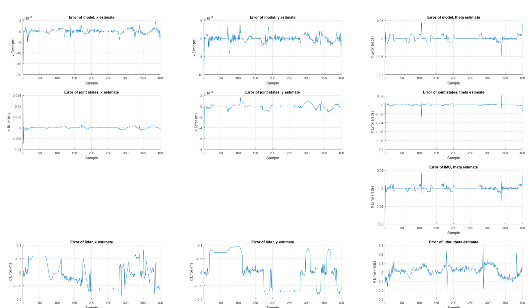


FIGURE 3: Measured errors of state transition model and observation models.

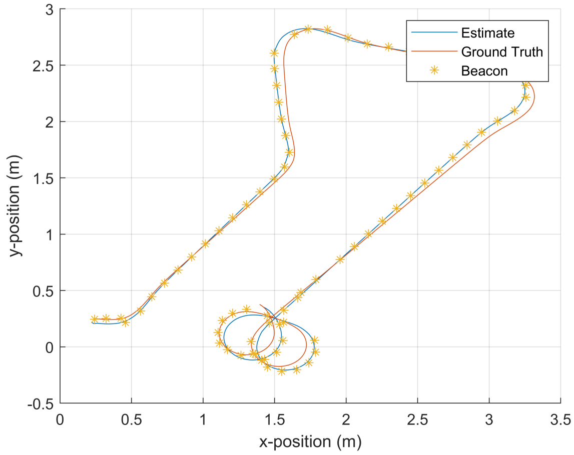


FIGURE 4: Position estimates versus ground truth (Simulation).

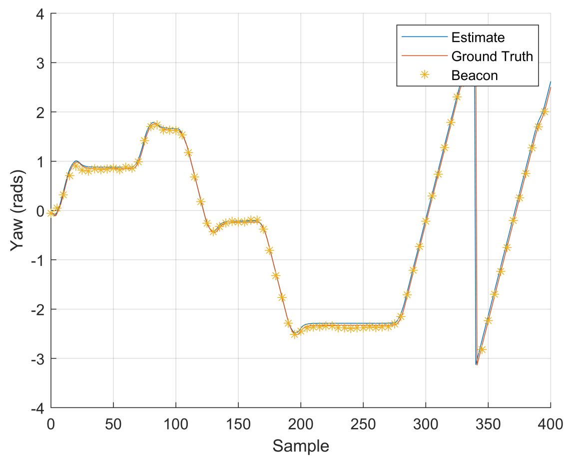


FIGURE 5: Yaw estimates versus ground truth (Simulation.

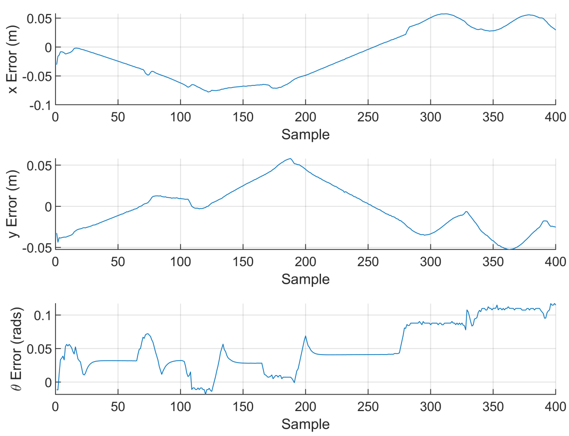


FIGURE 6: Error plot of figure 4 and **??**.

V. CONCLUSION

REFERENCES

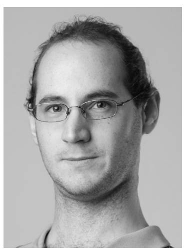
[1] S. Safari, F. Shabani, and D. Simon, "Multirate multisensor data fusion for linear systems using kalman filters and a neural network," Aerospace Science and Technology, vol. 39, pp. 465 – 471, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963814001126

[2] M. M. Atia, S. Liu, H. Nematallah, T. B. Karamat, and A. Noureldin, "Integrated indoor navigation system for ground vehicles with automatic 3-d alignment and position initialization," IEEE Transactions on Vehicular Technology, vol. 64, no. 4, pp. 1279–1292, 2015.

[3] A. Gopalakrishnan, N. Kaisare, and S. Narasimhan, "Incorporating delayed and infrequent measurements in extended kalman filter based nonlinear state estimation," Journal of Process Control, vol. 21, pp. 119–129, 01 2011.

[4] H. L. Alexander, "State estimation for distributed systems with sensing delay," in Data Structures and Target Classification, V. Libby, Ed., vol. 1470, International Society for Optics and Photonics. SPIE, 1991, pp. 103 – 111. [Online]. Available: https://doi.org/10.1117/12.44843

[5] M. Wu, H. Ma, M. Fu, and C. Yang, "Particle filter based simultaneous localization and mapping using landmarks with rplidar," in Intelligent Robotics and Applications, H. Liu, N. Kubota, X. Zhu, R. Dillmann, and D. Zhou, Eds. Cham: Springer International Publishing, 2015, pp. 592–603.



MICHAEL J. DUKE is was born in Adelaide, Australia in 1991. He is currently studying to receive the BEng(Hons) degree in mechatronic engineering for University of South Australia, Adelaide, Australia.

From 2009 to 2019, he held a full-time position as a Precision Agriculture Technician for Rocky River Ag Services in Crystal Brook, Australia. Since 2019, he has been working as a Mechatronic Design Engineer Undergraduate for Applidyne Australia Pty Ltd. in Adelaide, Australia. His current research interests include control systems, model based design and digital twins, and electric vehicle drive train and battery technology.