Faculty of Engineering Ain Shams University MCT431

Design of Autonomous Systems

Milestone 4

**TEAM 4**

## Presented by:

|  |  |
| --- | --- |
| **Name:** | **ID** |
| Kirolos Thabet Fouad | 19P6754 |
| Abdulrahman Mohammed Abdullah | 19P2862 |
| Omar Mahrous Eltoutongy | 19P1060 |
| Ahmed Amr Abdelbaki | 19P7696 |
| Yahia Ahmad Allam | 19P1714 |

**Presented to:**

Dr. Eng. Omar M. Shehata Eng. Ali Hussein

Contents

[1. FIRST REQUIREMENT 4](#_Toc135515919)

[1.1 Plots and Graphs of the odometry data taken before the addition of the processing noise. 4](#_Toc135515920)

[1.2 Comments on these results: 5](#_Toc135515924)

[1.3 Plots and Graphs of the odometry data taken after the addition of the processing noise. 7](#_Toc135515928)

[1.4 Comments on the results 8](#_Toc135515932)

[2. SECOND REQUIREMENT 10](#_Toc135515936)

[2.1 Brief Description of the Kalman filter 10](#_Toc135515937)

[2.2 Plots and Graphs showing the differences between the predicated states, noisy states and filter states 11](#_Toc135515938)

[2.3 Comments on the results: 14](#_Toc135515947)

List of Figures:

[Figure 1 Publishing of the data. 4](#_Toc135516034)

[Figure 2 Running of the python file. 4](#_Toc135516035)

[Figure 3 General Outlay of the data acquired from the sensors graphically. 5](#_Toc135516036)

[Figure 4 Theta data before adding noise. 5](#_Toc135516037)

[Figure 5 y-output data acquisition before adding noise 6](#_Toc135516038)

[Figure 6 x-output data acquisition before adding noise. 6](#_Toc135516039)

[Figure 7 Publishing the odometry data 7](#_Toc135516040)

[Figure 8 Running of the python file 7](#_Toc135516041)

[Figure 9 General overlay of the data collected from the sensors with noise included 8](#_Toc135516042)

[Figure 10 Theta-output data collected after noise addition 8](#_Toc135516043)

[Figure 11 y-output data collected after noise addition 9](#_Toc135516044)

[Figure 12 x-output data collected after noise addition 9](#_Toc135516045)

[Figure 13 Running of the python file showing the updated prediction error covariance 11](#_Toc135516046)

[Figure 14 Displaying of the kalman gain values 11](#_Toc135516047)

[Figure 15 Displaying of the kalman gain values 12](#_Toc135516048)

[Figure 16 Graphs of the noisy, filtered, and predicted theta 12](#_Toc135516049)

[Figure 17 Noisy, filtered, and predicted x data 13](#_Toc135516050)

[Figure 18 Noisy, filtered, and predicted x data 13](#_Toc135516051)

[Figure 19 Noisy, filtered, and predicted x data 14](#_Toc135516052)

[Figure 20 Noisy, filtered, and predicted x data 14](#_Toc135516053)

# FIRST REQUIREMENT

### Plots and Graphs of the odometry data taken before the addition of the processing noise.

### A screenshot of a computer Description automatically generated

Figure Publishing of the data.

### 

Figure Running of the python file.

### 

Figure General Outlay of the data acquired from the sensors graphically.

### Comments on these results:

What can be observed regarding the data acquisition of the three states that were monitored before adding the process noise from our noise, that the difference between the data collected from the sensors and the ground tooth is very close even nearing the zero mark. This indicates that the standard deviation of these data is very small as well as the covariance of the error resulting from the data collected.

### A picture containing screenshot, text, software, multimedia software Description automatically generated

Figure Theta data before adding noise.

### A screen shot of a graph Description automatically generated with medium confidence

Figure y-output data acquisition before adding noise

### A picture containing text, screenshot, software, multimedia software Description automatically generated

Figure x-output data acquisition before adding noise.

### Plots and Graphs of the odometry data taken after the addition of the processing noise.

### 

Figure Publishing the odometry data

### 

Figure Running of the python file

### 

Figure General overlay of the data collected from the sensors with noise included

### Comments on the results

After the addition of the process noise on the odometry data published on the odometry topic of the turtlebot3, it can be concluded that the standard deviation as well as the error covariance of the input variables being studied in our case increased drastically. The process noise coming from the sensor adversely affects error covariance of our data. In order to treat this recurring repercussion, a myriad of filters can be utilized such as the Bayesian filter or the Kalman filter. We will effectively use the Kalman filter in the second requirement to demonstrate the effectiveness of this filter.

### 

Figure Theta-output data collected after noise addition

### A screen shot of a graph Description automatically generated with medium confidence

Figure y-output data collected after noise addition

### 

Figure x-output data collected after noise addition

# SECOND REQUIREMENT

### Brief Description of the Kalman filter

The Kalman filter is a mathematical algorithm used in control systems and autonomous systems to estimate the state of a system based on noisy sensor measurements. It was first introduced by Rudolf Kalman in the 1960s and has since become a widely used tool in a variety of applications, including robotics, aerospace, and autonomous vehicles.

In autonomous systems, the Kalman filter is used to estimate the state of the vehicle, such as its position, velocity, and orientation, based on sensor measurements such as GPS, accelerometers, and gyroscopes. These measurements are often noisy and subject to errors, making it difficult to accurately determine the state of the system.

The Kalman filter works by combining the predicted state of the system based on its dynamics and the sensor measurements using Bayesian inference. The filter uses a model of the system's motion and measurement noise to create a probabilistic estimate of the system's state. As new sensor measurements are collected, the filter updates the estimate of the state based on the new information.

One of the key advantages of the Kalman filter is its ability to incorporate information from multiple sensors to produce a more accurate estimate of the system's state. It can also handle non-linear systems and non-Gaussian noise, which makes it a versatile tool for a wide range of applications.

In autonomous vehicles, the Kalman filter is used to estimate the vehicle's position, velocity, and orientation, which are crucial for navigation and control. The filter can also be used for obstacle detection and avoidance, as well as for predicting the behavior of other vehicles on the road.

Overall, the Kalman filter plays a crucial role in autonomous systems by providing accurate and reliable state estimation, which is essential for safe and effective operation.

### Plots and Graphs showing the differences between the predicated states, noisy states and filter states

### 

Figure Running of the python file showing the updated prediction error covariance

### 

Figure Displaying of the kalman gain values

### 

Figure Displaying of the kalman gain values

### 

Figure Graphs of the noisy, filtered, and predicted theta

### 

Figure Noisy, filtered, and predicted x data

### 

Figure Noisy, filtered, and predicted x data

### 

Figure Noisy, filtered, and predicted x data

### 

Figure Noisy, filtered, and predicted x data

### Comments on the results:

Filtering odometry data with a Kalman filter can improve the accuracy and reliability of the estimated position and orientation of a mobile robot. Odometry data is typically subject to noise and drift, which can accumulate over time and cause errors in the estimated position and orientation. From the acquired odometry data, we can be assured that the Kalman filter fulfilled its algorithm and tried to diminish the prediction error covariance in each of the collected data whether it’s the x-data, y-data, or theta-data collected. The states were updated with each iteration. It is noted that the discrete form of the modelling of the mobile robot was used to conduct the filter. The error covariance was predicted and then the Kalman gain was calculated. The Kalman gain indicates whether we trust the data collected by the sensor or not in this case the filtered data. If the Kalman gain is equal to one then this indicates that the sensor data is trusted. On the contrary, if it is equal to zero, then the predicted data is trusted. Regarding the obtained Kalman gain data we collected; it was 0.3 which indicates that we trust our predicted data more than the sensor data.

After that, the state of the robot was updated followed by the predicted error covariance. The Kalman filter is repeated for several iterations until the predicted error covariance is a very small number which gives an indication that we can trust the data collected from the sensor. From the plots and graphs, the sensor data can be trusted as the noise was mostly filtered diminishing the error covariance.