2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

An Analytical Study of Motion of Autonomous Vehicles under Imperfect Sensing

Swagata Biswas[1], Himadri Sekhar Paul[1] and Saurabh Bagchi[2]

***Abstract*— A fully tested autonomous system works predictably under ideal or assumed environment. However, its behavior is not fully defined when some components malfunction or fail. In this paper, we consider automated guided vehicle (AGV), equipped with multiple sensors, executing a traversal task in a static unknown environment. We have analytically studied the system, computed a set of performance and safety metrics, and validated it with simulation results in *Webots*. We have also analyzed the effect on system performance under independent and correlated sensing errors. We have also performed sensitivity analysis to identify the most critical components in any given system; and this can be utilized to increase the reliability of the system and its conformance to safety objectives.**

***Index Terms*— Reliability, Safety, Sensing Error, Sensor Sensitivity, Autonomous Systems, Robotics.**

I. Introduction

Autonomous systems are increasingly making their way deeper into industries like automobile, manufacturing, logistics, etc. However, automation raises many concerns and the major ones stem from safety and reliability of the system while maintaining acceptable performance. For example, one of the concerns is the throughput of the system, *e.g.*, the number of orders a fulfilment center is able to complete within a given time-frame. But, from the safety perspective, it is important to estimate, as an example, the probability of a collision between autonomous vehicles on the factory floor. These concerns are, however, generic and applicable across all domains.

Mobility is an essential task for most autonomous systems. In this paper, we consider an automated guided vehicle (AGV) performing the basic task of autonomously traveling to a specified location. The travel path is determined by some independent planner, given a floor-plan with known static obstacles in it. However, there may be temporary static obstacles on the planned path, unknown to the planner. For example, in a warehouse a box or a shelf may fall on the AGV's planned path. In such scenario, the AGV's intrinsic obstacle avoidance procedure has to take appropriate action.

The problem of obstacle avoidance for an AGV in an unknown environment is a well-studied problem [16], [20]. Obstacle detection and avoidance is a local decision aided by its sensors. Imperfections in sensors result in inaccurate estimation of the AGV's environment. Further, inaccuracies may get compounded through the software stack that processes the sensor readings and may lead to unsafe actuation. Additionally, the error may accumulate over multiple processing cycles, leading to catastrophic failure. Imperfection anywhere in the processing cycle may cause collision with other entities including humans in its vicinity [19], and therefore pose safety related concerns. A more elaborate discussion on sensing error classification and their effect on a robotic system can be found in [6], [14]. The key contributions in this paper are as follows:

- We have developed a generic analytical model of traversal task of an AGV, with obstacle avoidance maneuver and which incorporates imperfections in sensors. We use this model to derive analytical solutions to estimate a set of performance metrics, which can be used to improve system performance and reliability.
- We analyzed the sensitivity of the sensor errors on these performance metrics. *Sensitivity analysis* helps in identifying the most critical components in a given system and can increase the reliability of the system even before the actual deployment enhancing the overall safety or reliability of the deployed system.
- We experimentally evaluated the proposed model with a set of different AGVs equipped with different types of sensors, under *Webots simulation framework* and also with a physical robot. We studied the effect of both independent and correlated sensing errors. We found experimental results mostly conforms our model.

We present a brief literature survey, in Sec. II, on algorithms and models on autonomous traversal. We propose a generic motion-model of an AGV in Sec. III and derive various performance metrics. In Sec. IV, we present our experiments and the results, which show that our analytical model, although simplistic, is able to describe the system with requisite fidelity. Finally, in Sec. V, we present our concluding remarks with scope of future work in this context.

II. Related Work

In this paper, we present a generic motion model of an AGV in a partially known environment and analytically evaluate the effect of sensing errors on the performance and safety of the AGV. The problem of autonomous motion and obstacle avoidance for unmanned vehicles is well studied in literature [18]. Researchers usually concentrate on a particular aspect of autonomous systems. For example, there are proposals on analytical models for obstacle characteristics [8], obstacle detection and collision avoidance [16], [20], path planning [13], motion planning and control [10],

[1]Swagata Biswas and Himadri S Paul with the TCS Research, Kolkata, India {swagata.biswas, himadriSekhar.paul}@tcs.com

[2]Prof. Saurabh Bagchi with the Dept. of Electrical and Computer Engg., Purdue University, West Lafayette, IN 47907, USA sbagchi@purdue.edu

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981329

and other aspects associated with autonomous motion. Safety is intrinsically related to these problems and is well studied in literature [3], [7]. Ensuring safety through demarcation of safety zones around robots is widely used in industrial automation domain. Distance threshold based *red-yellow-green* zone demarcation is a common approach for vehicles [15]. In [19] the authors offer a solution to ensure safety in human-robot collaborative work-space. The safety concept is capable of establishing both manually defined safety zones and dynamically generated safety zones that are based on robot joint positions and velocities, thus offering a maximum free space around the robot *w.r.t.* its users. On the other hand, techniques like formal verification perform static analysis on a certain autonomous system to certify it to be safe [11]. Modelling the system accurately is the central problem for such approaches.

To operate reliably an autonomous system must be capable to handle uncertainties arising, not only from the environment in which it is operating, but also uncertainties arising within and [21] presents an interesting discussion on these aspects. Authors in [17] present an analytical method of estimating the probability of collision under sensing error. Authors in [12] presents a safe autonomous navigation which incorporate different types of uncertainties in modelling and perception system. While the authors in [2] presents a verification framework for planned maneuvers which incorporates sensor noise among other uncertainties arising out of inexact estimation of initial state of the vehicle and frictional coefficients.

In this paper, we present a generic motion model of AGVs with obstacle avoidance, and the model is sensitive to sensing imperfections. Such a model is useful to analyze its movement in environment with unknown obstacles and imperfect sensing. This model can also be used to analyze the performance degradation of the system due to aging.

III. AGV Traversal Model

In any autonomous system, the hardware and software components are closely inter-linked. Typically the relation is non-linear and varies across components and therefore it is difficult to create a detailed analytical model of the system. Instead, we model the behavior of an AGV from the point of view of a traversal task, with the aim to produce a generic model, which incorporate the effect of sensing imperfections. The model will enable us to study the effect of sensing inaccuracies relative to a set of defined metrics.

A. Problem Definition

An AGV is given a traversal task to move from its present location $S(x_0, y_0)$ to a target location $D(x_N, y_N)$ following a planned path. The planned path is described as a sequence of N piece-wise linear segments and is represented as a sequence of way-points $[(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)]$, $[(x_i, y_i), (x_{i+1}, y_{i+1})] : 0 \leq i < N)$ denotes one line segment. It travels the segments with a constant velocity of v. The vehicle is equipped with a set of sensors, possibly heterogeneous in nature, to detect obstacles on its path. An on-board control software processes sensor readings and generates actuation signals for its driving wheels. The sensor readings are processed in a single duty cycle ($sense \rightarrow compute \rightarrow actuate$) of duration τ.

Given that some unknown obstacles are on its planned path, we need to estimate the travel time to D, including additional effort to avoid obstacles. With imperfect sensing, the vehicle may react erroneously and this not only can affect the distance to travel but also generate possibilities of collision with obstacles. We consider two performance metrics - the estimated completion time of the task *i.e.* the traversal time, and the number of possible collisions when the sensors are imperfect. The objective is to - *a)* study the performance of the system under imperfect sensors, for both uncorrelated and correlated failure models, *b)* estimate the sensitivity of each of the individual sensors, as well as identify the critical sensors.

In the following sections we present a traversal model for an AGV and derive analytical models for performance metrics.

B. Model Assumptions

In order to develop an analytical model, we make some simplifying assumptions. The obstacles are assumed to be static. The obstacles can occur randomly on the path, but follow some well-defined distribution. The distribution corresponds to the center of the area the obstacle occupies. Similarly the area the obstacles occupy (*i.e.* the size) is random, but *independent but identically distributed*. The velocity of the vehicle is also assumed to be constant in this formulation irrespective of the fact that it may be negotiating a bend. We also assume that AGVs are holonomic. In the following sub-sections, we present our model to capture the traversal behavior of an AGV and derive closed form analytical expressions as estimates of traversal times and collision count under imperfect sensing.

C. Motion Model

The model presented here is based on the perception of the AGV *w.r.t.* unknown obstacles on its path. The AGV can be in one of the three zones - *red*, *yellow* or *green*, relative to its nearest obstacle. The AGV is in *red* zone when the nearest obstacle is within r_R distance, it is in *yellow* zone when the distance is in between r_Y and r_R ($r_Y > r_R$), otherwise it is in *green* zone. Fig. 1 depicts red and yellow zones, and the rest of the space is in green zone. The vehicle follows its planned path when it is in the green zone. However, when a vehicle enters the yellow zone, first it performs evasive maneuvers to get around the obstacle and then re-plans its path to eventually merge with its earlier planned path. We assume here, the re-planned path does not differ drastically from its previous planned path. Below we describe a traversal model based on such obstacle avoidance method.

Let the length of the planned path between S and D be $\mathbf{d}$. Also let the total travel time of the vehicle be denoted as $\mathbf{T}$, which we need to estimate. $\mathbf{T}$ is a random variable dependent

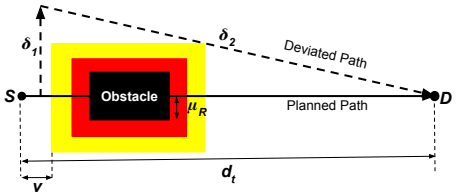


Fig. 1. Deviated path of the vehicle (marked in dashed line) due to obstacle avoidance maneuver when it detects an obstacle (solid black rectangle) in its yellow zone

on the distribution of obstacles and their sizes and we attempt to express this relation as closed form expression. Following the duty cycle of sensing, the traversal model is made discrete in time, with time quantum equal to the duty cycle τ. Therefore, there are $\mathbf{T}/\tau$ duty cycles within its traversal duration. Without loss of generality, the following analysis is performed with $\tau = 1$ for simplifying the presentation and computation. So, the following analysis estimates $\mathbf{T}$ as the number of duty cycles required to complete a traversal task. Estimation of time is, therefore, $1/\tau$ times the number of duty cycles.

At every time instance the vehicle always travels v distance. Obstacle avoidance maneuver increases the remaining traversal distance and consequently time whenever the vehicle enters the yellow or red zone. Let, at the time instance t, the estimated distance to be traversed by the vehicle to reach the target location from its present location is denoted as d_t. So, when a traversal task is initiated, the estimated distance to be traversed by the vehicle be $d_0 = \mathbf{d}$. Let the probability of detecting an obstacle in each time cycle be p. Hence, with probability p the obstacle maneuver is triggered and the vehicle takes the alternative longer path, shown as dashed line in Fig. 1. With this model of traversal, we can now present the following recurrence,

$$d_0 = \mathbf{d}, \quad d_T = 0, \quad d_{t+1} = (1-p)(d_t - v) + p(\delta_1 + \delta_2) \quad (1)$$

According to our model, the traversal algorithm triggers obstacle avoidance maneuver when the sensors identify obstacles within the yellow zone. Also any obstacle entering the red zone is assumed to collide with the vehicle. Therefore, the embedded processing endeavors to guide the vehicle in such a manner such that no obstacle enters the red zone. The navigation system is also assumed to drive the vehicle such that the obstacle is in the green zone. So the minimum distance of the nearest obstacle is always r_Y. The obstacles are modeled as convex objects with mean radius of μ_R. The estimation of deviated path length, in reference to Fig. 1, is as follows:

$$\begin{aligned}
\delta_1 &= \mu_R + r_Y \\
\delta_2 &= \sqrt{(d_t - v)^2 + \delta_1^2} = (d_t - v)\sqrt{1 + \left(\frac{\delta_1}{d_t - v}\right)^2} \\
&\approx (d_t - v)\left(1 + \frac{1}{2}\left(\frac{\delta_1}{d_t - v}\right)^2\right) \text{ [assuming } \delta_1 \ll d_t - v\text{]} \\
&= (d_t - v) + \frac{\delta_1^2}{2(d_t - v)}
\end{aligned}$$

Now, with these estimations, Eq. 1 is expanded and simplified as follows:

$$\begin{aligned}
d_{t+1} &= (d_t - v) + p\left(\delta_1 + \frac{\delta_1^2}{2(d_t - v)}\right) \\
\implies d_t - d_{t+1} &= v - p\delta_1\left(1 + \frac{\delta_1}{2(d_t - v)}\right)
\end{aligned}$$

Since we assume $\delta_1 \ll (d_t - v)$, $\delta_1/(2(d_t = d_0 - d_{\mathbf{T}} - v)) \rightsquigarrow 0$, and therefore, approximately, $d_t - d_{t+1} = v - p\delta_1$. The recurrence can be simply solved by summing both sides,

$$\begin{aligned}
\textit{l.h.s.}, \quad \textstyle\sum_{t=0}^{\mathbf{T}}(d_t - d_{t+1}) &= d_0 - d_{\mathbf{T}} = \mathbf{d} \\
\textit{r.h.s.}, \quad \textstyle\sum_{t=0}^{\mathbf{T}}(v - p\delta_1) &= \mathbf{T}(v - p\delta_1)
\end{aligned}$$

Since, *l.h.s.* = *r.h.s.*,

$$\mathbf{T} = \frac{\mathbf{d}}{v - p\delta_1} = \frac{\mathbf{d}}{v - p(\mu_R + r_Y)} \quad (2)$$

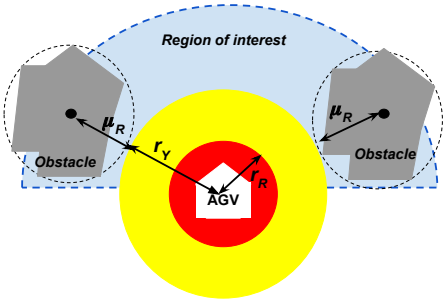


Fig. 2. Obstacles within the Region of interest (RoI) of the vehicle affects the planned path of the vehicle as it moves forward

We model the unknown obstacles to follow some stochastic distribution with mean μ_O, which defines the density distribution of centroids of obstacles in the area. As the vehicle traverses, the expected probability to encounter an obstacle is $p = A \times \mu_O$, where A is the area for the region of interest (*RoI*) to the vehicle. This area affects alteration in motion (refer to Fig. 2). The planned path of the vehicle changes when an obstacle enters the yellow region. As the vehicle traverses, the perimeter of the RoI advances, bringing possible obstacle within RoI. As the vehicle moves with a speed of v, the perimeter of RoI sweeps new area of size $v \times \pi(\mu_R + r_Y)$. We are interested to compute the probability of obstacles in the new sweep area which is $p = A \times \mu_O = v\pi(\mu_R + r_Y) \times \mu_O$ The estimation of the traversal time is the expected value of the random variable $\mathbf{T}$ and takes the following form,

$$\begin{aligned}
\mathbb{E}[\mathbf{T}] &= \frac{\mathbf{d}}{v - (v\mu_O\pi(\mu_R + r_Y))(\mu_R + r_Y)} \\
&= \frac{\mathbf{d}}{v}\frac{1}{1 - \mu_O\pi(\mu_R + r_Y)^2}
\end{aligned} \quad (3)$$

D. Faulty Sensor

In a practical scenario the readings from the sensors are not always perfect. Imperfections are due to manufacturing defects (internal) or change in environmental conditions (external), like fog or increase in suspended particles in air, *etc.*. Here, we can compute the deviation (δ_1) the vehicle

is likely to take in presence of error in its distance sensors. First, let us consider the simplest case where only one sensor is used for obstacle detection.

Let us denote the sub-region of yellow zone which is not part of the red zone, *i.e.* $[r_R, r_Y]$, as the yellow band and the width of the yellow band be $b_Y = r_Y - r_R$. The control algorithm always endeavors to ensure that the vehicle does not enters yellow or red zones. When the vehicle is able to detect the obstacle on its path while it enters the yellow zone from green zone, the vehicle needs to travel b_Y distance away from the obstacle to return to safe zone.

A collision is said to happen when the vehicle enters the red zone from the yellow zone due to sensor failure. The failure to detect obstacle while in the yellow zone, given that sensing is imperfect, is a conditional case. An obstacle is present in the yellow zone and the sensor fails to detect it to be in the yellow zone when the associated sensing error is more than the yellow band width ($\epsilon > b_Y$). So, the probability of failure to detect obstacle in yellow zone, when it is there, is denoted as $Pr(\overline{D_Y}O_Y)$, where O_Y is the event that an obstacle is in its yellow zone and $\overline{D_Y}$ denotes event of failure to detect the obstacle in yellow zone and where, F_ϵ is the CDF of the error distribution associated with the sensor.

$$Pr(\overline{D_Y}O_Y) = F_\epsilon(\epsilon > b_Y) = 1 - F_\epsilon(b_Y)$$

In Fig. 2, we endeavor to elucidate the condition for a RoI for the vehicle's sensing system. The figure shows the case when the planned traversal path leads the vehicle through two obstacles on both sides. In order to pass through the narrow path between them, the vehicle need to sense their presence. The boundary condition of the place of the obstacle is when their periphery just touches the yellow zone as the AGV passes through them. If the relative distance between the obstacle is more than that, sensing one of them is sufficient. We define the area marked with bold dotted blue as the RoI when the vehicle begins its movement through the narrow path between the obstacles. The sensing system should detect obstacles at the periphery of the region to be identified as them being in the yellow region. Since obstacles behind do not interfere with the traversal, the RoI is a semi-circular space. The obstacles at the periphery are of interest to start an obstacle avoidance maneuver. Since the obstacles have mean diameter of $2\mu_R$, the obstacle can be present in a semicircular band of width $2\mu_R$ at distance of r_Y from the vehicle. So the probability of obstacles in this periphery is,

$$\begin{aligned} Pr(O_Y) &= F_O\left(\frac{\pi}{2}\left((2\mu_R + r_Y)^2 - r_Y^2\right)\right) \\ &= F_O\left(2\pi\mu_R(\mu_R + r_Y)\right) \end{aligned}$$

where, F_O is the CDF of the distribution of the obstacle centroid in the given space.

Therefore the conditional probability, that the sensor fails to detect an obstacle given the object is in the yellow zone, is expressed as $Pr(\overline{D_Y}|O_Y)$ and is computed as,

$$Pr\left(\overline{D_Y} \mid O_Y\right) = \frac{Pr\left(\overline{D_Y}O_Y\right)}{Pr(O_Y)} = \frac{1 - F_\epsilon(b_Y)}{F_O(2\pi\mu_R(\mu_R + r_Y))} \quad (4)$$

The imperfect sensing affects obstacle avoidance maneuver and hence its path; and introduces possibility of collision. First we incorporate modification required to cater in sensing imperfections to our estimation of task completion time. When the obstacle is not detected within the yellow zone and it enters the red zone, the vehicle is required to maneuver in such a way that the obstacle is no longer found in its red zone before performing obstacle avoidance maneuver. In this case it traverses additional b_Y distance. So, $\delta_1 = (1 - Pr(\overline{D_Y}|O_Y))(\mu_R + r_Y) + Pr(\overline{D_Y}|O_Y)\ (\mu_R + r_Y + b_Y) = (\mu_R + r_Y) + Pr(\overline{D_Y}|O_Y) \times b_Y$ (refer to Fig. 1). The recurrence solution in Eq. 3 is modified as follows,

$$\begin{aligned} \mathbb{E}[\mathbf{T}] &= \frac{\mathbf{d}}{v - v\mu_O(\mu_R + r_Y)\ \left(\mu_R + r_Y + Pr(\overline{D_Y}|O_Y)\ b_Y\right)} \\ &= \frac{\mathbf{d}/v}{1 - \mu_O(\mu_R + r_Y)\left(\mu_R + r_Y + b_Y Pr(\overline{D_Y}|O_Y)\right)} \quad (5) \end{aligned}$$

Let us estimate the number of collisions. For this we note that the semantics of the safety regions around the vehicle implies that the vehicle tries to ensure that no obstacle is found in the red zone. Semantically any obstacle entering the red zone is equivalent to collisions and we count the expected number of times obstacle enters the red-zone as the number of collisions. The estimated distance traveled by the vehicle is $v \times \mathbb{E}[\mathbf{T}]$. So, the area under surveillance during its traversal is $v\mathbb{E}[\mathbf{T}] \times 2(\mu_R + r_Y)$. Since the density of obstacles in the arena is μ_o, the expected number of obstacles in the path is $2v\mathbb{E}[\mathbf{T}](\mu_R + r_Y) \times \mu_o$. Hence the expected number of collisions is:

$$\begin{aligned} \mathbb{E}[CC] &= 2v\mu_O(\mu_R + r_Y)\ \mathbb{E}[\mathbf{T}]\ Pr(\overline{D_Y}|O_Y) \\ &= \frac{2\mathbf{d}\ \mu_O\ (\mu_R + r_Y)\ \ Pr(\overline{D_Y}|O_Y)}{1 - \mu_O(\mu_R + r_Y)(\mu_R + r_Y + b_Y Pr(\overline{D_Y}|O_Y))} \quad (6) \end{aligned}$$

E. Multi-sensing and Sensitivity Analysis

In this section we consider scenarios where multiple sensors are installed on a vehicle for sensing obstacles around it. This is usually the design of most of the autonomous vehicles where multiple heterogeneous sensors are used to facilitate more accurate perception of the environment around. The importance of a sensor reading is based on the motion of the vehicle. For example, when the vehicle moves forward, the sensors which detect obstacles ahead of the vehicle are more important than the rear sensors; while the rear ones are more important when the vehicle is backing up.

In this context we define the sensitivity of a sensor as the deviation in the estimated performance measure of the system with the deviation of the sensor reading from its expected value due to error. The sensing error is modelled as a random value taken from a distribution with a mean μ_ϵ and deviation be σ_ϵ. For sensitivity analysis, deviation of the error is more important than its mean, since a known mean error can be easily handled by the processing unit treating it as an offset.

When there are multiple sensors, and each of these sensors is associated with its own error distribution model with variation σ_{ϵ_i} (for the i^{th} sensor). The mean μ_ϵ of error distribution for the overall system, as discussed earlier, are related to those of the individual sensors (μ_{ϵ_i}) by some transfer function and is denoted in our model as $\sigma_\epsilon = \phi(\vec{\sigma_\epsilon})$, where $\vec{\sigma_\epsilon} = \{\sigma_{\epsilon_i} : 0 \leq i < n\}$.

Given a system performance measure $\mathcal{M}$ which is a function of sensing error, expressed as $\mathcal{M}(\sigma_\epsilon)$, sensitivity of a sensor is defined as,

$$\frac{\partial \mathcal{M}(\sigma_\epsilon)}{\partial \sigma_{\epsilon_i}} = \frac{\partial \mathcal{M}(\sigma_\epsilon)}{\partial \sigma_\epsilon}\ \frac{\partial \phi(\vec{\sigma_{\epsilon_i}})}{\partial \sigma_{\epsilon_i}}$$

For example, when the performance measure is the travel time of the vehicle, sensitivity of a sensor is expressed as,

$$\frac{\partial \mathbb{E}[\mathbf{T}]}{\partial \sigma_{\epsilon_i}} = \frac{\mathbb{E}[\mathbf{T}]^2 v \mu_O (\mu_R + r_Y)}{\mathbf{d}} \frac{\partial Pr(\overline{D_Y}|O_Y)}{\partial \sigma_\epsilon} \frac{\partial \phi(\vec{\sigma_{\epsilon_i}})}{\partial \sigma_{\epsilon_i}} \quad (7)$$

In the following section we present our experiments, the specific model for the experimental setups, and compare the estimations from the model with the experimental results.

IV. Experiment and Discussion

To validate our model we performed various experiments both with a simulation system as well as with a physical robot. We model the obstacle distribution as *gamma distribution* and sensing errors as *Gaussian distribution*. The closed form solutions of the performance metrics, described in Sec. III, are computed in Sec. IV-A. Sec. IV-B presents our experiment setups, followed by comparative analysis of our model with experimentally collected data.

A. Model Specific to Experiment

We assume obstacles are randomly distributed following gamma distribution with parameters (k, θ). Therefore, $\mu_O = k\theta$. Also, $k, \theta \geq 0$. Since the obstacles are detectable by sensing systems, they are comparable to the vehicle in dimension. This implies that the gamma distribution is not a flat distribution and hence $k \geq 1$. Since, we represent μ_O as the density of obstacle in the arena, the additional constraint is $0 \leq \mu_O \leq 1$. The CDF of gamma distribution is expressed as,

$$F_O(x) = \frac{1}{\Gamma(k)} \gamma\left(k, \frac{x}{\theta}\right)$$

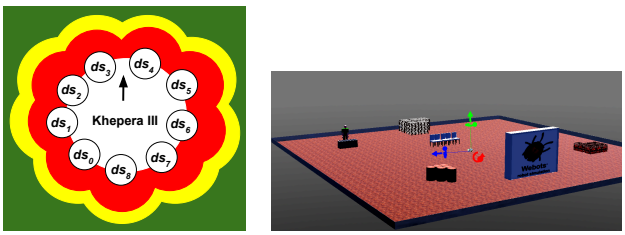
Also we assume the sensing errors follow normal distribution $N(\mu_\epsilon = 0, \sigma_\epsilon^2)$ [9]. The CDF of normal distribution is,

$$F_\epsilon(x) = \frac{1}{2}\left[1 + erf\left(\frac{x}{\sigma_\epsilon\sqrt{2}}\right)\right]$$

The conditional probability of detection of failure can be computed as follows.

$$Pr(\overline{D_Y} \mid O_Y) = \frac{\Gamma(k)}{2} \frac{1 - erf(b_Y/\sigma_\epsilon\sqrt{2})}{\gamma(k, 2\pi\mu_O\mu_R r_Y(\mu_R + r_Y)/k)} \quad (8)$$

where, $\theta = \mu_O/k$ replaced in the gamma distribution.



(a) Proximity sensors and safety zone demarcation for Khepera III

(b) Simulated arena with heterogeneous obstacles

Fig. 3. Experiment setups

B. Experimental Setup

Our simulation were performed in the *Webots simulation framework* [5] using three different AGVs - Khepera III (*K3*), Pioneer 3-DX (*P3-DX*), and Pioneer 3-AT (*P3-AT*), equipped with different sensors described in Table I. The working models of these AGVs along with the sensors are already available with the Webots system. We generated an 25 m-sq arena bounded by walls in the simulation system, in which the AGVs perform assigned traversal tasks of various lengths. Different number of obstacles, with random dimensions, are placed randomly in the arena. The same set of experiments were performed both under perfect and imperfect sensing system. Out of these three AGVs, only Khepera III is equipped with an array of proximity sensors and their positions are depicted in Fig. 3(a), and hence this AGV is used for our experiments for correlated sensing error condition described in Sec. IV-C.4. The following subsection outlines behaviour of the control subsystem driving the robots in Webots system.

Input : $D = (x_d, y_d)$; `// Destination location`
Data : $\overrightarrow{Th_R} = \{Th_R^i : 0 \leq i < n\}$; `// Red-zone`
Data : $\overrightarrow{Th_Y} = \{Th_Y^i : 0 \leq i < n\}$; `// Yellow-zone`
begin
1 **while** *Current location is not* D **do**
2 **SENSE:**
$\vec{S}$ = Read proximity sensor values;
$zone = green$, $yellow$, or red, based on proximity readings, $\vec{S}$, indicating the distance between the vehicle and the nearest obstacle;
3 **COMPUTE:**
if $zone = Green$ **then**
Align pose towards D using GPS and Compass;
Move forward;
else
4 **if** $zone = Yellow$ **then**
Apply *Braitenberg* algorithm [4] to determine direction to move;
else
// $zone = Red$
Move backward;
if $signaturePath = NULL$ **then**
Update $signaturePath$;
else
5 **if** $replannedPath = signaturePath$ **then**
Align pose randomly;
Update $signaturePath$;
end
end
end
end
6 **ACTUATE:**
Perform movement;
end
end

Algorithm 1: Traversal algorithm of AGV

Control System Description: The task of the control system is to guide the AGV to a target location following

AGVs	Max vel. (m/s)	Avg vel. (m/s)	Proximity sensor type	Proximity sensor no.	Max range (m)	Field of View	r_R (m)	r_Y (m)	Other sensors
Khepera III	0.3	0.21	Infrared	9 (ds0-ds8)	0.5	360°	0.026	0.086	Ground, Ultrasonic, GPS, Compass
Pioneer 3-DX	1.2	0.45	MS Kinect	1	3.5	70.6°× 60°	0.5	1	Sonar, GPS, Compass
Pioneer 3-AT	0.7	0.46	Sick LMS 291 Lidar	1	80	180°	1	1.5	Sonar, GPS, Compass

TABLE I

DESCRIPTION OF AGVS USED IN OUR EXPERIMENTS

a planned path, avoiding collision with obstacles on its path. An outline of the control objective is described in Algorithm 1. When the vehicle detects an obstacle in its path from its sensor readings, an obstacle avoidance maneuver is triggered, for example a *Braitenberg* [4] (at line 4 of Algorithm 1). The actuation algorithm sets motor speed to individual motors appropriately to drive the vehicle in the intended direction. Sometimes the vehicle gets trapped within a barricaded region which is formed due to awkward positioning of a group of obstacles such that an AGV cannot move forward, left or right without colliding. The only option is to backtrack sufficiently and then travel in a different direction to avoid the group of obstacles (line 5 of Algorithm 1). When the sensors are non-faulty, the collision avoidance functionality is able to correctly detect and navigate the AGV away from the obstacles. When sensing is imperfect, there is a probability of delay in triggering the collision avoidance algorithm, which may result in a collision.

Experiment with Physical Robot: We have also collected data and compared observations for *Double-3* [1], a telepresence robot in our laboratory. For our experiments the robot was placed on a pre-determined place in our lab and the destination was manually specified to the robot, while some obstacles were placed randomly on its path. *Double-3* has its own in-built obstacle avoidance procedure, which is aided by its in-build 3D sensing array. This robot was used only for verifying estimates from our model for the perfect sensing case only.

The following section presents our experimental results for both perfect and imperfect sensing scenarios.

C. Results and Discussion

In this section we compare and contrast our model with experimentally collected data. Each experiment was performed 10 times and the measurements were averaged for presentation. The simulation results are contrasted with those estimated from our model, both under error-less and erroneous sensor reading conditions in Sec. IV-C.1 and Sec. IV-C.2 respectively.

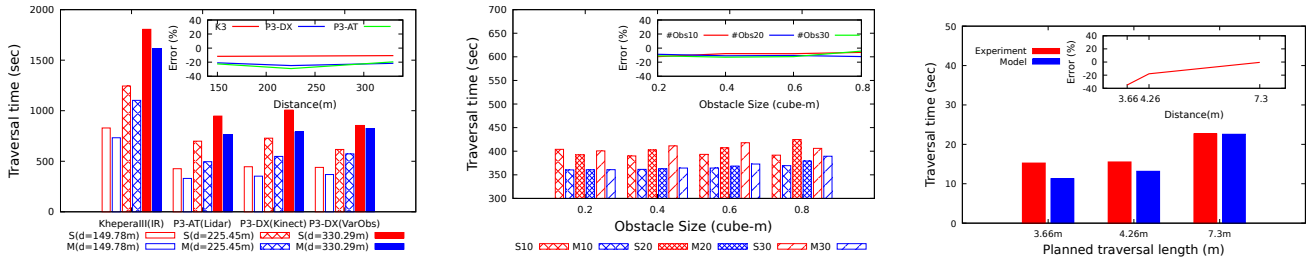
1) Experiment-I: Perfect Sensing: In this first set of experiments we attempt to validate our proposed model in a perfect sensing condition. We compare the estimated traversal time from the model with that obtained from experiments for various traversal distances with different number and size of obstacles. The first three clusters in Fig. 4(a) corresponds to three AGVs travelling in an arena with 20 obstacles of size 0.6 m-cube randomly placed, while the last cluster corresponds to *P3-DX* travelling in a domestic environment consisting of various heterogeneous obstacles as shown in Fig. 3(b) . The inset plot shows deviation of the estimation as error percentage. A general observation from these results is that the model always underestimates the travel time. It has its root in the approximation performed in derivation of Eq. 2 by ignoring an additive term with small value. The ignored value is comparable for smaller traversal distance and induces high error, but as the distance increases the effect of this reduces and is evident from Fig. 4(a) *i.e.* the model is more accurate over longer paths.

Fig. 4(b) contrasts this model for various number and size of obstacles for Khepera-III. The plot shows clusters with the same obstacle sizes, and within a cluster the number of obstacles increases. Both the experimental results, as well as model estimations, show that the traversal time of the AGV increases as the obstacles becomes larger. This is expected since the AGVs need to travel more distance to avoid them. When the obstacles sizes are small (0.2 m-cube) there are sufficient free space for the robot to navigate around. Increasing the number of obstacles does not drastically affect the path length except when more number of obstacle may appear on the path. In either case the increase in travel distance is small since the size of the obstacles are also small. Such a variation can be seen in experimental results within all the clusters in the figure, but the model absorbs this variation as mean under its stochastic treatment. However, as the size of the obstacles increases, free space in the arena decreases. This results in the path length to be lengthier to avoid obstacles. The model can capture this effect as evident from the cluster corresponding to the mean obstacle size 0.8 m-cube in Fig. 4(b).

The simulation result from the other configurations exhibit similar pattern and are not shown here. In fact our experiment with the physical robot, *Double-3*, also show the same pattern and is depicted in Fig. 4(c).

2) Experiment-II: Single Sensor Independent Error: In this section we present experimental results for imperfect sensing, *i.e.* the sensed values contain error but the error is independent of error in sensor readings from other sensors. We simulate the error in sensor readings by injecting white Gaussian noise (mean, $\mu = 0$). Fig. 5 compares time and collision estimations for Pioneer 3D-X for different traversal distances. Although our model underestimates the metrics, it follows the trend of experimental results. A similar trend is visible for our experiments with Pioneer 3-AT and are not shown here. This implies that the model also sincerely estimates the metrics when sensors are imperfect.

We have also studied the performance of our model as the variation of the sensing error increases ($0.1 \leq \sigma \leq 0.5$).



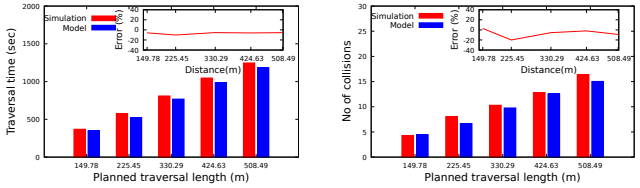
(a) Results clustered according to the AGVs

(b) Clustered by obstacle size and within cluster obstacle number varies (for Khepera III)

(c) Traversal time comparison for the physical robot Double-3

Fig. 4. Comparison of time estimations when sensing is perfect. S and M represent simulation and model results respectively. Si and Mi represent the simulation and model results respectively for i number of obstacles.

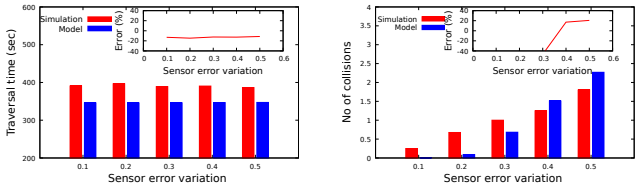
Fig. 5(a) compares travel time estimations with experimental results for increasing error variations and this also closely matches. However, the collision count estimations, as shown in Fig. 5(b) does not match very closely. However, when these values are rounded to its closest integral values (since the number of collisions is an integral value), the match is exact. For collision avoidance, the importance of correct reading increases (analogously, the disastrous effect of anomalous reading increases) inversely with the distance from the obstacle. So, the timing of an anomaly in reading for obstacle distance is important, and this timing factor which is not well captured in the model.



(a) Comparison of traversal time

(b) Comparison of collision count

Fig. 5. Single faulty sensor in P3-DX, with error distributed as $N(\mu = 0, \sigma = 1)$, in an arena with 20 homogeneous obstacles of size 0.6 m-cube.



(a) Comparison of traversal time

(b) Comparison of collision count

Fig. 6. Single faulty sensor in P3-DX, with normal error distribution with $\mu = 0$ but different variances, in an arena with 15 homogeneous obstacles of size 0.4 m-cube.

3) Experiment III: Sensor Sensitivity: It is easy to see that when the sensor readings are composed as a linear function, $\phi(\cdot)$ (the transfer function defined in Sec. III-E) is a linear function. The transfer function used in control system in Khepera-III is modeled as a weighted average function of sensor readings, *i.e.* $\sum_{i=0}^{n-1}(w_i r_i)$ such that $\sum_{i=0}^{n-1} w_i = 1$. Then, ϕ is also the same linear weighted function defined as $\phi(\vec{\mu_{\epsilon_i}}) = \sum_{i=0}^{n-1}(w_i \mu_{\epsilon_i})$. When the sensor composition is defined as weighted average as defined above, the last factor in Eq. 7 is $\partial\phi(\vec{\mu_{\epsilon_i}})/\partial\mu_{\epsilon_i} = w_i$. With this sensitivity of the i^{th} sensor stands as follows,

$$\frac{\partial \mathbb{E}[\mathbf{T}]}{\partial \sigma_{\epsilon_i}} = w_i \frac{\mathbb{E}[\mathbf{T}]^2 v \mu_O (\mu_R + r_Y)}{\mathbf{d}} \frac{\partial Pr(\overline{D_Y}|O_Y)}{\partial \sigma_\epsilon}$$

Since the other factors are constant, this implies that the sensor associated with the highest normalized weight is most sensitive.

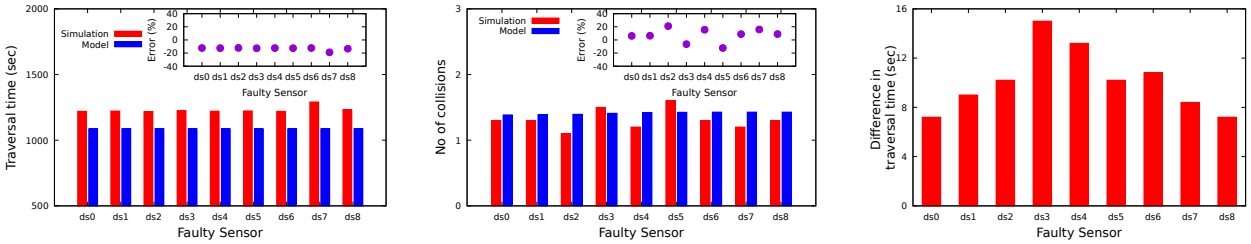
In this experiment we study the effect on the expected traversal time when one of the proximity sensors of Khepera-III is faulty. In this experiment, the faulty sensor is injected with Gaussian distributed error ($\mu_{\epsilon_i} = 0, \sigma_{\epsilon_i} = 1$). Fig. 7(c) shows the additional travel time required due to error. According to our experimental results, additional task completion times are higher when the front sensors (*ds3* and *ds4*) are faulty. The weights associated with the sensors in the control system is as follows and this shows that *ds3* and *ds4* carry higher weights, in order.

$$\vec{w} = [0.016, 0.1, 0.133, \mathbf{0.233}, 0.216, 0.149, 0.1, 0.016, 0.033]$$

Sensitivity analysis from our experimental results shown in Fig. 7(c) conform with the model-based sensitivity analysis. This implies that to improve the overall performance of the system, the reliability of these front proximity sensors should be improved.

4) Experiment-IV: Multi-Sensor Correlated Error: Khepera III is equipped with an array of proximity sensors and values from these sensors are fused for proximity measurement in the control system. In these experiments, we study the performance of the system under correlated sensing errors. The correlation of sensing error distributions is represented as a co-variance matrix $\mathbb{C} = [[\rho_{ij}]]$ and treated under the general framework of co-variance computation. The co-variance matrix $\mathbb{C}$, for n sensors, is an $n \times n$ matrix drawn from $[-1, 1]$, where $1, 0$, and -1 represents perfect correlation, independence, and anti-correlation respectively. In this context, for practical purposes, the sensors are not anti-correlated, and hence $\rho_{ij} \in [0, 1]$. For normally distributed co-related errors, mean: $\mu_\epsilon = \sum_i w_i \mu_{\epsilon_i}$ and variance: $\sigma_\epsilon = \sum_i w_i^2 \sigma_{\epsilon_i} + 2\sum_{i<j}\sum_j w_i w_j \rho_{ij}$ such that ρ_{ij} is the co-variance of error between the i^{th} and the j^{th} sensors.

In our experiment, nine proximity sensors (ref. Fig. 3(a)) situated at the base of the Khepera-III are injected with



(a) Variation in traversal time

(b) Variation in collision count

(c) Variation in additional distance travelled shows that sensor ds3 is most sensitive

Fig. 7. Primary sensing error, from Gaussian distribution with $(\mu = 0, \sigma = 1)$, is injected in one of the sensors of Khepera III, while errors for other sensors are computed based on the correlation matrix $\mathbb{C}$. The AGV was operating in an arena of size 25 m-sq with 15 homogeneous obstacles of size 0.4 m-cube randomly placed. These results depict variations against primary sensing error injection.

correlated errors, computed from a correlation matrix, $\mathbb{C} = [[\rho_{ij}]]$, whose one representative row is $\rho_5 = [0.2, 0.4, 0.6, 0.8, \mathbf{1.0}, 0.8, 0.6, 0.4, 0.2]$ and the other rows are simple rotational permutations. Here ρ_{ij} represents the correlation coefficient of error among the i^{th} and j^{th} sensors. The rationale behind the design of such a correlation matrix is based on the assumption that a fault in one sensor affects failure in neighboring sensors and the effect dies down with larger separation between the pair. So the geometry of the sensor positions (ref. Fig. 3(a)) corresponds to such correlations. The sensor error vector is generated as $E_c = LE$, where the lower triangular matrix L is obtained by Cholesky decomposition such that $\mathbb{C} = LL^T$, and the $E = [e_0, e_1, \ldots e_8]$, where e_i represents the independent error value generated for the i^{th} sensor. The computed errors are then injected to individual distance sensors. If we want to inject an ϵ amount of error to the sensor *ds0*, then *ds1* sensor to be induced with 0.8ϵ error, *ds2* by 0.6ϵ, and so on.

Fig. 7 shows comparison of estimations for correlated sensing errors. Here the faulty sensor denotes the primary sensor which is injected with a Gaussian distributed error with $\mu_\epsilon = 0$ and $\sigma_\epsilon = 1$. The remaining proximity sensors are injected with correlated error as described above. Comparisons of traversal time and collision count estimations are shown in Fig. 7(a) and Fig. 7(b) respectively. These have the similar trends as imperfect sensing as discussed in Sec. IV-C.2. The sensitivity to error *w.r.t.* the performance (in this traversal time) of the Khepera-III, follows our intuition that *ds3* is the most influential sensor. Error in this sensor results in longer path to travel than any other sensor in the AGV.

V. Conclusion and Future Work

In this work we proposed a generic traversal model for an AGV under imperfect sensing. This model enables us to analytically study some important performance metrics *e.g.* task completion time, collision count, *etc.* and its sensitivity to various kinds of errors in the sensors. Experiments show that our model, although simple, has acceptable accuracy of estimating these performance metrics. As a future study we intend to extend this model to remove some of the assumptions and also extend this for 3-D space, which will be applicable to autonomous drone systems.

References

[1] Double Robotics - Telepresence Robot for the Hybrid Office. `https://www.doublerobotics.com`. Accessed : 2022-04-26.

[2] M Althoff and J M Dolan. Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4):903–918, 2014.

[3] C. Bila et al. Vehicles of the future: A survey of research on safety issues. *IEEE Trans. on Intelligent Transportation Systems*, 18(5):1046–1065, 2016.

[4] V Braitenberg. *Vehicles: Experiments in synthetic psychology*. MIT press, 1986.

[5] Cyberbotics Ltd. Webots. `https://https://www.cyberbotics.com`. Accessed: 2022-04-13.

[6] C Ferrell. Failure recognition and fault tolerance of an autonomous robot. *Adaptive behavior*, 2(4):375–398, 1994.

[7] J Guiochet, M Machin, and H Waeselynck. Safety-critical advanced robots: A survey. *Robotics and Autonomous Systems*, 94:43–52, 2017.

[8] G Klančar et al. Modelling and simulation of a group of mobile robots. *Simulation Modelling Practice and Theory*, 15(6):647–658, 2007.

[9] J H Kotecha and P M Djuric. Gaussian sum particle filtering. *IEEE Transactions on signal processing*, 51(10):2602–2612, 2003.

[10] J P Laumond et al. *Robot motion planning and control*, volume 229. Springer, 1998.

[11] M. Luckcuck et al. Formal specification and verification of autonomous robotic systems: A survey. *ACM Computing Surveys*, 52(5):1–41, 2019.

[12] Nadhir M et al. Safe and adaptive autonomous navigation under uncertainty based on sequential waypoints and reachability analysis. *Robotics and Autonomous Systems*, 152, 2022.

[13] T T Mac et al. Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86:13–28, 2016.

[14] R R Murphy and D Hershberger. Classifying and recovering from sensing failures in autonomous mobile robots. In *Proc. of the Natl. Conf. on Artificial Intelligence*, pages 922–929, 1996.

[15] N Nikolakis, V Maratos, and S Makris. A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace. *Robotics and Computer-Integrated Manufacturing*, 56:233–243, 2019.

[16] A Pandey et al. Mobile robot navigation and obstacle avoidance techniques: A review. *Intl. J. of Robotics Automation*, 2(3), 2017.

[17] S Patil et al. Estimating probability of collision for safe motion planning under gaussian motion and sensing uncertainty. In *IEEE Intl. Conf. on Robotics and Automation*, pages 3238–3244, 2012.

[18] M. Radmanesh et al. Overview of path-planning and obstacle avoidance algorithms for UAVs: A comparative study. *Unmanned systems*, 6(02):95–118, 2018.

[19] C Vogel et al. Safe human-robot cooperation with high-payload robots in industrial applications. In *11th Intl. Conf. on Human Robot Interaction*, pages 529–530. IEEE Press, 2016.

[20] X Yu and M Marinov. A study on recent developments and issues with obstacle detection systems for automated vehicles. *Sustainability*, 12(8):3281, 2020.

[21] Q Zhu et al. Know the unknowns: Addressing disturbances and uncertainties in autonomous systems. In *Proc. of the 39th Intl. Conf. on Computer-Aided Design*, pages 1–9, 2020.