

Dead Reckoning Using Time Series Regression Models

João B. Pinto Neto*
joao.pinto@ifro.edu.br

Nathalie Mitton†
nathalie.mitton@inria.fr

Miguel Elias M. Campista‡
miguel@gta.ufrj.br

Luís Henrique M. K. Costa§
luish@gta.ufrj.br

ABSTRACT

Connected car technology promises to drastically reduce the number of accidents involving vehicles. Nevertheless, this technology requires the vehicle precise location to work. The adoption of Global Positioning System (GPS) as a navigation device imposes limitations to geolocation information under non-line-of-sight conditions. This work introduces the Time Series Dead Reckoning System (TedriS) as a solution for dead reckoning navigation when the GPS fails. TedriS uses Time Series Regression Models (TSRM) and the data from the rear wheel speed sensor of the vehicle to estimate the absolute position. The process to estimate the position is carried out in two phases: training and predicting. In the training phase, a novel technique applies TSRM and stores the relationship between the GPS and the rear wheel speed data; then in the predicting phase, this relationship is used. We analyze TedriS using traces collected at the campus of Federal University of Rio de Janeiro (UFRJ), Brazil, and with indoor experiments with a robot. Results show an accuracy compatible with dead-reckoning navigation state-of-art systems.

CCS CONCEPTS

• **Networks** → **Mobile networks**; Network reliability; • **Computing methodologies** → **Model verification and validation**; • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics;

KEYWORDS

Dead-reckoning, connected vehicles, time series, robot.

ACM Reference Format:

João B. Pinto Neto, Nathalie Mitton, Miguel Elias M. Campista, and Luís Henrique M. K. Costa. 2018. Dead Reckoning Using Time Series Regression Models. In *SMARTOBJECTS'18: 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects, June 25, 2018, Los Angeles, CA, USA*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3213299.3213305>

*Federal Institute of Education, Science and Technology of Rondônia, Brazil

†Institut National de Recherche en Informatique et en Automatique, France

‡GTA/Universidade Federal do Rio de Janeiro, Brazil

§GTA/Universidade Federal do Rio de Janeiro, Brazil

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SMARTOBJECTS'18, June 25, 2018, Los Angeles, CA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5857-6/18/06...\$15.00

<https://doi.org/10.1145/3213299.3213305>

1 INTRODUCTION

Connected vehicles mean applications, services, and systems that enable vehicle connectivity to its vicinity including safety applications, road side assistance, and autonomous self-driving, among others. Connected vehicle technologies are a promise to reduce vehicle accidents and casualties, producing numerous positive economic and societal impacts. By limiting the number of crashes, thousands of lives will be saved, millions of injuries prevented and billion dollars will be saved annually [7].

However, these applications must rely on an accurate vehicle localization, perception of the environment, and wireless communication means [8]. Concerning vehicle localization, automotive navigation systems usually employ Global Positioning System (GPS) receivers, combined with other sensors, to estimate the vehicle position. Sensor fusion techniques can overcome the unreliability of GPS when the vehicle drives in urban canyons, tunnels, and under foliage [16]. In extreme situations however, where GPS is completely out of service, vehicle position can still be estimated using sensors and the last reliable position. This blind navigation is called Dead Reckoning (DR).

There are numerous works exploring DR navigation using different fusion techniques combined with various sensor devices. A Dead Reckoning With Dynamic Errors (DRWDE) using Kalman Filter with dynamic covariance matrix merged with an Interacting Multiple Model (IMM) is proposed by [4] to predict vehicle position 3 s ahead of time. A single-sensor device dynamic model for DR navigation was proposed in [10]. The vehicle kinetic data acquired by an Inertial Measurement Unit (IMU) is modeled using a set of AutoRegressive models with eXogenous input (ARX). The GPS independent localization terrain based system with lane level accuracy proposed in [1], aims to avoid collisions due the lane change maneuvers. With the aid of a terrain map, the system uses information of an IMU and vehicle odometer to perform the localization algorithm. A particle filter is responsible to estimate the location, using data provided by the IMU while the vehicle drives in the same lane. [17] introduced a Wavelet Neural Network (WNN) based DR systems. The WNN is carried out when GPS is reliable where the localization is estimated with the information of a Strapedown Inertial Navigation System (SINS), i.e., an IMU device mounted rigidly to the body axes of the vehicle, a magnetometer, and GPS data fused into a Kalman filter. Authors of [2] proposed a novel hybrid approach based on neural networks (NN) and AutoRegressive Integrated Moving Average (ARIMA) models to improve the accuracy of vehicle position estimation. During the training mode, NN is trained with GPS, odometer, and gyroscope data fused by an Extended Kalman Filter (EKF), while ARIMA

works over the residuals of non-linear structures learned by NN. The aforementioned works are state-of-art DR systems. Different strategies and sensor combinations are used to improve accuracy of vehicle positioning and tracking. Regarding the sensors, only one system ([10]) acquires DR data from a single sensor (the IMU), while the remaining use two or more sensors to perform DR navigation. Regarding the applied technique, Kalman Filter was adopted by almost all works followed by neural networks. Our proposal differs from those presented by using a single data source

native in all vehicles to implement DR navigation. This allows use of the system by any vehicle regardless of brand, cost or age.

In this context, we aim to support a significant number of vehicles not protected by safety systems. We propose a DR navigation system, called Time Series Dead Reckoning System (TedriS) that uses a conventional GPS as navigation device and the data of the rear wheels of the vehicle, provided by the native ABS speed wheel sensors through the interface with the Controller Area Network (CAN), i.e., it does not require any additional sensor. TedriS comprises a training phase and a predicting phase working in a collaborative way to estimate the vehicle location when GPS fails. During the training phase, the system collects data from the GPS and CAN interface to feed a time series regression model to generate a database with the relationship between the two variables in small blocks of data. This data is later used in the predicting phase to estimate the vehicle location, only using a simple implementation of a time series regression model and CAN interface data. TedriS performance was evaluated with real car traces and experiments with a robot. The results show that TedriS accuracy during a 80 s interval is situated within the range achieved by the state-of-art DR systems. The results also show better accuracy with robot experiments, confirming that TedriS can also be applied to robot DR navigation.

This work is organized as follows. Section 2 describes the system modeling. Section 3 describes TedriS training and predicting phases and time series regression model. Section 4 evaluates the performance of TedriS. Finally, Section 5 provides closing remarks and future work directions.

2 SYSTEM MODELING

The design of a DR system faces several challenges to achieve accurate vehicle localization. This section details the tools and techniques employed in the system modeling.

2.1 Dynamic Model

We assume that dynamic model state variables are referenced from car-like front wheel drive with no lateral slipping of the tires. We define a coordinate system whose origin is attached to the center of gravity of the vehicle (CG) as the body frame where x axis is aligned with the vehicle longitudinal axis. We also define an east-north-up coordinate system as navigation frame where the heading φ is the angle from east to north, the same GPS heading rule (Fig. 1). The relationship between front (φ_F) and rear (φ_R) yaw rates and wheels speeds is [9]:

$$\dot{\varphi}_F = \frac{(V_{FR} - V_{FL})R_F}{b_F \cos \delta} \text{ and } \dot{\varphi}_R = \frac{(V_{RR} - V_{RL})R_R}{b_R}, \quad (1)$$

where $V_{FR}, V_{FL}, V_{RR}, V_{RL}$ are linear speeds of front and rear wheels, b_F, b_R are front and rear track width, and R_F, R_R are front and rear

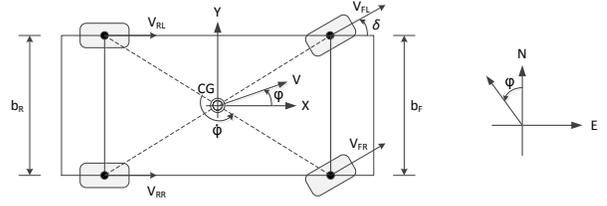


Figure 1: Front wheel drive vehicle dynamic model with the body frame coordinate system attached in the Center of Gravity (CG) and the navigation frame coordinate system adopted in Dead Reckoning System design. The heading angle φ follows the same GPS heading rule (from [9]).

wheel radius, respectively. We use φ_R to estimate the vehicle's CG yaw rate without the need of the steering angle δ . Thus, the kinematic model of the vehicle is $\dot{x} = -V \sin \varphi$ and $\dot{y} = V \cos \varphi$, where V is the instantaneous speed. There are many sources of errors that

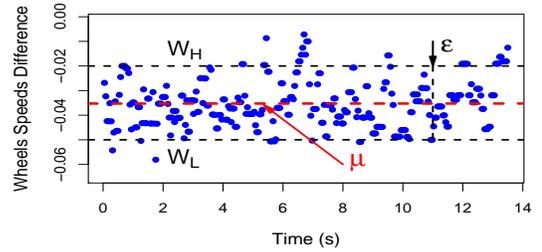


Figure 2: Typical wheels speeds difference acquired when a vehicle describes a straight line trajectory where ϵ is the estimated noise amplitude interval, μ the mean value, W_L and W_H are the lower and the upper bounds.

can compromise the efficiency of the dynamic model, such as speed sensors noise, difference of wheels radii, and position of the GPS antenna in relation to CG. Several studies to estimate the radius of wheels [3, 13] confirm the impact of this factor over the position accuracy. Thus, our focus is to overcome this source of error using a technique to estimate this difference dynamically. Fig. 2 shows the typical behavior of the difference of rear wheels speeds in relation to time by a vehicle describing a straight-line trajectory. Noise amplitude ϵ delimited by W_L and W_H is represented by the black dashed lines and the mean μ is represented by the dashed line. Suppose that the noise is a white noise like and the wheels have exactly the same radius, the mean μ would be zero. This does not occur in this case due to tire different pressure, unbalanced tire wearing and electronic circuitry issues, including sensors misalignment. In order to reduce these issues effects, TedriS evaluates periodically the predicted position accuracy. A dynamic correction factor generated at each evaluation point is used to improve DR navigation performance.

2.2 Time Series Regression Models (TSRM)

Time series data is characterized by the correlation displayed by observations and their temporal sequence [5]. TedriS explores TSRM in order to determine the relationship between the heading angle rate ($\dot{\theta}$) acquired from the GPS and the vehicle yaw rate ($\dot{\varphi}$) acquired from rear wheels speeds information through the CAN interface. These two independent collections of values, sampled within a uniform time interval, fit the time series definition [11]. Thus, the use of time series regression will give the temporal relation between these series both in the matter of delay and amplitude. The design of TSRM starts by the discretization of the series to investigate their relationship. Assume φ_t and θ_t are the discrete values of $\dot{\varphi}$ and $\dot{\theta}$ at time t , we define $\omega_t = \text{diff}(\varphi_t)$ and $\gamma_t = \text{diff}(\theta_t)$, where diff is the difference operator.

To know whether past values of ω_t can be used to predict γ_t and how this relationship happens, sample Cross Correlation Function (CCF) is used to generate a set of sample correlations, or lags, of ω_t . These values are used to predict γ_t as input of a transfer function model that consists of an ordinary linear regression model. Hence, according to [14], the relationship between ω_t and γ_t is:

$$\gamma_t = \gamma_0 + \alpha_0\omega_t + \alpha_1\omega_{t-1} + \alpha_2\omega_{t-2} + \alpha_3\omega_{t-3} \dots \alpha_n\omega_{t-n} + \eta_t, \quad (2)$$

where γ_0 is the mean value of γ_t when $\omega_{t-h} = 0$, $h = 0 \dots n$, $\alpha_1, \alpha_2, \alpha_3 \dots \alpha_n$ are the weights of each correlated lag and η_t is the error. The analysis of this error or residual will help to decide if we have to correct the model using an Autoregressive (AR) or a Moving Average (MA) approach [14].

3 TEDRIS DESCRIPTION

TedriS is composed of two distinct stages. The first one, the training phase, collects, treats and processes the input data of GPS receiver and CAN interface and in addition it generates and stores

the data that will be used later on. The second one defined as DR navigation phase uses the data generated by the training phase to predict the trajectory of vehicle, based only on CAN data.

3.1 System Training Phase

TedriS training phase must be performed a priori when GPS receiver data is reliable. The dynamic characteristics between GPS and CAN are estimated by TSRM while the vehicle performs a road course. Nevertheless, instead of applying the linear regression on the entire set of data captured after the training phase, TedriS applies it in small consecutive blocks of data in order to capture the non-linearities more accurately. The training phase is depicted in Fig. 3. The CAN yaw rate (ω_t) and GPS heading (θ) are acquired after the synchronization to overcome the differences of sample rates. Controlled by the fastest sample rate clock, the buffer data feeds a First-in-First-out queue. Once the queue is full, the data is transferred to the sample register at each clock pulse, i.e. new data available in the buffer. As θ varies from 0 to 360 degrees, it is necessary to remove its discontinuity when a transition occurs (0 to 360 or 360 to 0). After that, the GPS yaw rate ($\gamma_t(\tau)$) is obtained by applying a difference operator to θ . The relationship between γ_t and ω_t is carried out using a set of k ordinary linear regressions whose inputs are τ consecutive values of ω_t as in:

$$\hat{\gamma}_k = a_{0k} + a_{1k} \cdot \text{median}(\omega_{t_{k-1}}, \omega_{t_k}, \omega_{t_{k+1}}, \dots, \omega_{t_{k+\tau-1}}), \quad (3)$$

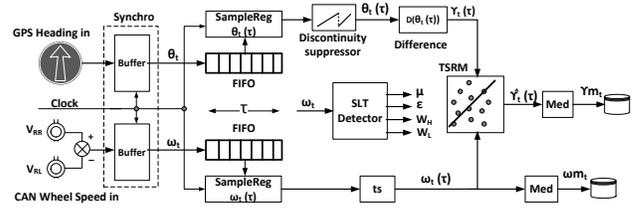


Figure 3: Block Diagram of TedriS training phase. Each consecutive τ data block of GPS heading (θ_t) and CAN yaw rate (ω_t) are processed by the TSRM to generate the tuple ($\gamma_{m_t}, \omega_{m_t}$) database. Straight Line Trajectory (SLT) provides parameters related to the difference in wheels speeds data.

where t_0 is the start time and $\hat{\gamma}_k$ is the predicted variable.

Thus, at each block of τ samples of ω_t , a linear regression is made to predict the value of γ_t . The coefficients a_{mn} are calculated using Eq. 2 with the highest correlation lag of ω_t . During all the training phase long, the tuple ($\text{Med}(\omega_t), \text{Med}(\hat{\gamma}_t)$) is stored to be used in the DR navigation phase. The use of the median rather than the average value to estimate the central tendency is justified by the small number of samples and to avoid the influence of outliers.

In addition, in this phase the values of ϵ , μ and the upper (W_H) and lower (W_L) bound of wheel speeds difference noise (Fig. 2) are estimated by the Straight Line Trajectory (SLT) detector. Defining $[t_s, t_e]$ as a SLT time interval and ω_{se} the set of all values of ω_t where $t \in [t_s, t_e]$, we have: $\mu = \text{mean}(\omega_{se})$, $W_H = \max(\omega_{se})$, $W_L = \min(\omega_{se})$, and $\epsilon = W_H - W_L$.

The tuples ($\hat{\gamma}_t, \omega_t$) must be selectively stored to provide fast retrieval of data when requested by the DR phase. In order to meet the dynamic range of ω_t and limit the size of the databases, the storage is divided into small repositories corresponding to intervals

of ω_t values delimited by W_L and W_H . Thus, for values of ω_t belonging to the interval $[W_L, W_H]$, the tuples ($\hat{\gamma}_t, \omega_t$) will be stored into the repository R_0 , and for values of ω_t greater than W_H they will be stored in R_n for $\omega_t \in (W_H + (n-1)\epsilon, W_H + n\epsilon]$.

The same applies for values of ω_t smaller than W_L to generate $R_{-1}, R_{-2}, \dots, R_{-n}$. Generalizing, if $\omega_t \in [-k, k]$, the number of repositories will be $2k/\epsilon$. Hence, the total storage required for the training phase can be easily estimated by delimiting a minimum and a maximum amount of data for each repository. The training phase completes when all repositories are filled at least with the minimum amount of data. TedriS executes the training phase periodically updating the repositories in conformity with eventual changes in the vehicle dynamical model.

3.2 System Dead Reckoning (DR) Phase

TedriS DR phase (or predicting phase) is the main focus of the system. All previous techniques and procedures are designed to predict vehicle position with the highest accuracy. This phase is performed when the GPS receiver data is unreliable and periodically after the training phase to update the dynamic correction factor (D_{CF}). DR phase uses the difference of rear wheels, speeds acquired from the CAN interface and the data stored selectively by the training phase to predict the vehicle heading angle as shown in Fig. 4. The input sequence is similar to the training phase where τ consecutive block

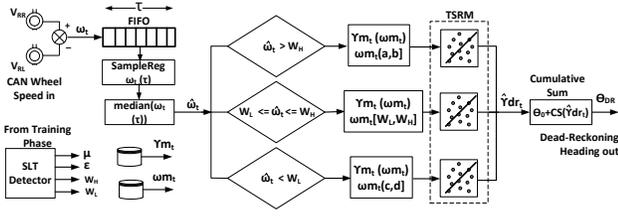


Figure 4: TedriS dead reckoning (predicting) phase. The median ($\hat{\omega}_t$) of consecutive τ data blocks of ω_t is the index to retrieve data from the repositories where tuples ($\gamma_{m_t}, \omega_{m_t}$) are stored. All the data generated by the training phase feeds the appropriate TSRM to generate the DR heading.

samples of ω_t are acquired after the full charge of the FIFO queue. In the same way that the training phase data was computed, the median value of $\omega_t(\tau)$ ($\hat{\omega}_t$) is used to predict $\hat{\gamma}_{dr_t}$. In order to allow a fast recovery of stored data, TedriS determines from which repository the set of tuples ($\gamma_{m_t}, \omega_{m_t}$) is used to apply TSRM. If $\hat{\omega}_t \in [W_L, W_H]$, the repository data R0 will predict $\hat{\gamma}_t(\tau)$ according to:

$$\hat{\gamma}_{dr_t} = \alpha_0 + \alpha_1 \text{Med}(\omega_{m_t-h}) + D_{CF}, \quad (4)$$

where α_0, α_1 are the TSRM coefficients, h is the lag of ω_{m_t} corresponding to the highest correlation coefficient of $\text{CCF}(\gamma_{m_t}, \omega_{m_t})$ and D_{CF} is the dynamic correction factor. For values of $\hat{\omega}_t$ greater than W_H , $\hat{\gamma}_{dr_t}$ is calculated also using Eq. 4 and the index number n of the repository Rn is $n = \lceil (\hat{\omega}_t - W_H) / \epsilon \rceil$ and the corresponding lower (a) and upper (b) bounds are $a = W_H + (n-1)\epsilon$ and $b = a + \epsilon$. Similarly for values of $\hat{\omega}_t$ lower than W_L the index number n of the repository Rn is $n = \lfloor (\hat{\omega}_t - W_L) / \epsilon \rfloor$ and the corresponding lower (c) and upper (d) bounds are $c = d - \epsilon$ and $d = W_L - n\epsilon$.

Once TSRM is applied, $\hat{\gamma}_{dr_t}$ is calculated by making $D_{CF} = \mu$. This is the dynamic correction factor initial value that will be periodically updated by the system. Based on the value of $\hat{\omega}_t$, the appropriate data is retrieved from the repository and $\hat{\gamma}_{dr}$ is obtained using Eq. 4. Hence, applying the cumulative sum operator (CS), the DR heading is given by:

$$\theta_{DR_k} = \theta_0 + \sum_{k=1}^m \hat{\gamma}_{dr_k}. \quad (5)$$

During the DR update mode, TedriS adjusts the D_{CF} to minimize the DR navigation error (E_{DR}) over a fixed time interval k . Considering θ_k and θ_{DR_k} , GPS and DR heading at instant k , we have:

$$E_{DR_{Min}} = \min(\text{RMSE}(\theta_k - \theta_{DR_k})). \quad (6)$$

The value of D_{CF} that minimizes E_{DR} is used in the DR phase predicting mode until a new update is performed.

3.3 TedriS Algorithm Sequence

DR functions are performed depending on GPS heading data status. When the GPS is operational, the training phase is executed, producing the data needed for the next phase. When all repositories of training data have enough data, the DR update phase is executed, generating D_{CF} . This cycle runs continuously until a GPS failure occurs and triggers the DR predicting phase.

4 TEDRIS PERFORMANCE EVALUATION

The evaluation of TedriS performance was carried out in two distinct environments: a vehicle and a robot.

4.1 Vehicle Performance

The data used in this evaluation comes from real traces captured at the campus of Federal University of Rio de Janeiro. The vehicle's on-board unit was equipped with a U-blox M8 GNSS device ([15]) that provides the heading angle at 4 samples per second and with a CAN interface that provided the 4 wheels speed at 25 samples per second. The input data of the training phase was acquired while the vehicle was traveling on the road stretch shown in Fig. 5. The

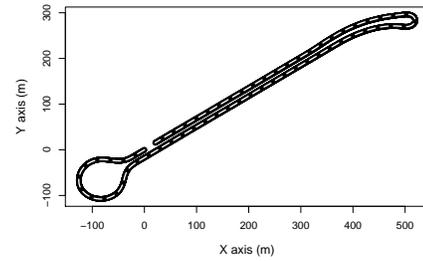


Figure 5: TedriS vehicle training phase road stretch with straight sections, light and sharp curves and roundabout. GPS is the black solid line, TedriS is the white dashed line.

black solid line is the GPS heading data and the white dashed line is the training phase estimate heading. Temporal data analysis reveals a strong correlation between ω_{m_t} and γ_{m_t} registering peak values at lags -2 and -3, highlighted in Fig. 8a (these values are determined dynamically using CCF). An analysis of residuals for regression correction using the AR and MA models revealed significance which is not desirable, confirmed with the Box-Pierce statistic test. Thus, choosing lag -2 the Time Series regression model is shown in Table 1.

Table 1: TedriS vehicle regression model. The coefficients α_0 and α_1 satisfies the relation $\gamma_{m_t} = \alpha_0 + \alpha_1 \omega_{m_{t-2}} + \eta_t$.

	Estimate	Standard Error	t.value	p.value
α_0	0.068529	0.003772	18.17	0.0000
α_1	2.251480	0.022151	101.64	0.0000

Nevertheless, TedriS applies this model not in the entire set of the tuple ($\gamma_{m_t}, \omega_{m_t}$), but in τ blocks of data. The best τ value was found empirically by varying τ until the training phase error below 1 m threshold level is achieved. The study shows that small values of τ increases training phase performance. The output data of this phase, provided dynamically by the data set ($\gamma_{m_t}, \omega_{m_t}$) is $\tau = 6$, $\max(\hat{\omega}_t) = 0.281 \text{ rd/s}$, $\min(\hat{\omega}_t) = -0.607 \text{ rd/s}$, $\mu = -0.0352 \text{ m}$, $W_L = -0.0581 \text{ m}$, $W_H = -0.01 \text{ m}$ and $\epsilon = 1.5(W_H - W_L) = 0.0721 \text{ m}$

Fig. 6 shows both the update and predicting phases applied to two distinct road stretches. The corresponding DR Error can be

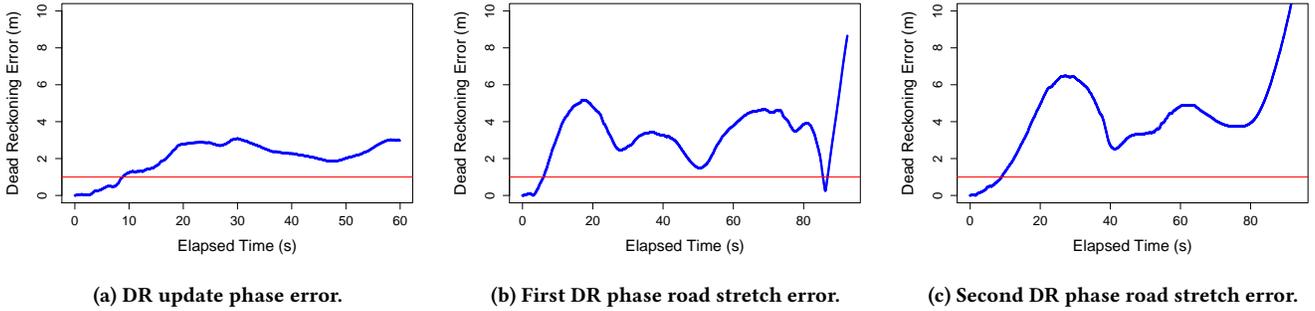


Figure 6: Vehicle DR phase performance. DR update, first and second predicting phases and their corresponding DR errors. The DR mean error below 5 meters was held by 80 s. The Red line is 1m DR threshold mark error.

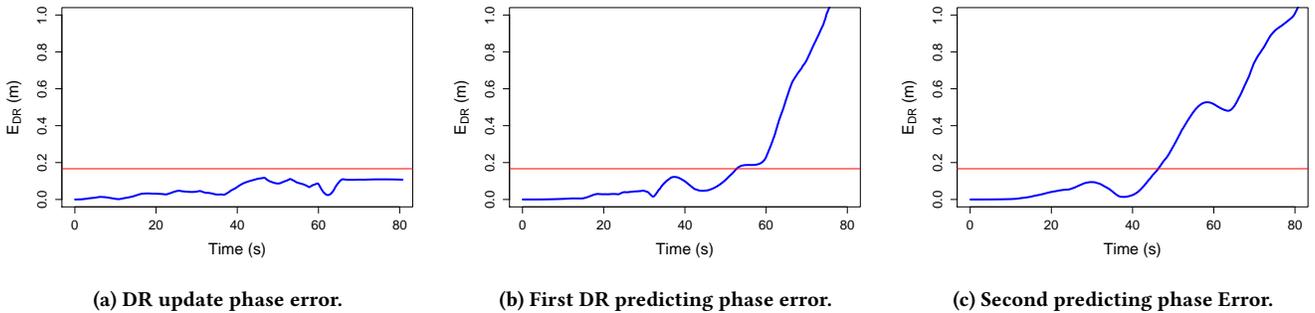


Figure 7: Robot DR phase performance (DR update, first and second predicting phases with DR errors).

evaluated against the 1 m threshold red line on the bottom. This threshold level is the desired performance of DR navigation systems. The DR update phase has produced a dynamic correction factor, $D_{CF} = -0.03755$, slightly different from the value of the mean, $\mu = -0.0352$, that guarantees the smallest E_{DR} (Fig. 6a). This value of D_{CF} has resulted in the best algorithm response and was used in both road stretches of the DR predicting phase. For the first road segment where TedriS predicted the position of the vehicle, the mean DR error is close to 3 m for little more than 80 s, as shown in Fig. 6b. The second DR navigation test where the vehicle drives more than 80 s presented a mean DR error close to 4 m (Fig. 6c).

4.2 Robot Performance

Experiments with a Turtlebot2 robot were carried out in the dependencies of Inria at Villeneuve d’Ascq, France. The Turtlebot2 is an open source hardware platform and mobile base equipped with a 110 degrees/s factory-calibrated gyroscope [12]. Software packages to control the Turtlebot2 were developed using the Indigo distribution of the Robot Operating System (ROS) platform [6]. With Turtlebot2, TedriS worked with gyroscope and wheel encoder ticks. An advantage is that the two signals come from the same source at 10 Hz sample rate and no synchronization was needed. All the trajectories the robot speed was maintained constant at 0.3 m/s. To provide data for the training phase, the trajectory shown in Fig. 8b

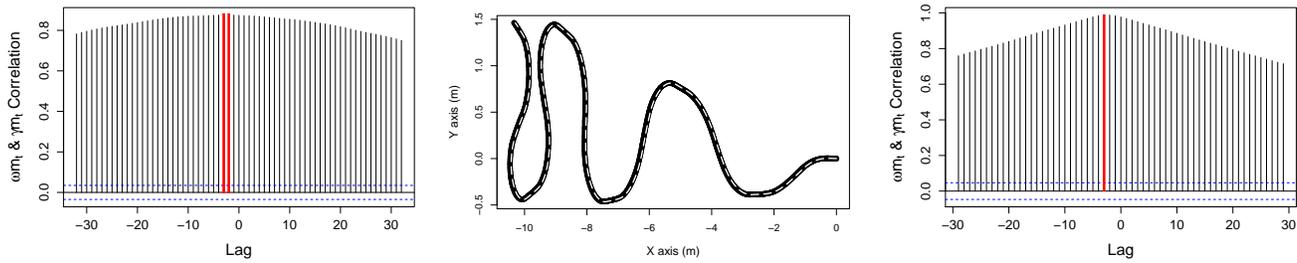
was generated in the same way as in previous experiments. Black solid line is the gyroscope heading data and the white dashed line is the training phase heading. The average system error was below 0.16 m. Note that the trajectory was shaped to provide a wide range of yaw rate values. It initiates with light curves and gradually the curves become sharper. The correlation between γm_t and ωm_t , showed a peak value at lag -3 , is highlighted in Fig. 8c.

Analyzing the residuals, we identify a moving average 2 (MA2) process which resulted in the correction of the original model. The new coefficients and parameters that characterize the MA2 structure for the residuals are shown in Table 2.

Table 2: TedriS robot MA model. The new coefficients α_0 and α_1 was used in the TSRM DR predicting phase.

	Estimate	Standard Error	t.value	p.value
MA2	0.6333	0.0137	46.3037	0.0000
α_0	0.0000	0.0001	-0.2277	0.8199
α_1	0.0492	0.0004	140.3375	0.0000

Previous tests with this new model produced better performance without selective data storage, i.e., $\epsilon = \infty$, thus the remaining data provided by this phase is $\tau = 6$, $\max(\hat{\omega}_t) = 0.482 \text{ rd/s}$, $\min(\hat{\omega}_t) = -0.526 \text{ rd/s}$, $\mu = -0.0047m$. Using the same pattern of the vehicle



(a) Temporal correlation between ωm_t and γm_t from the vehicle training phase. One can observe a strong yaw rate range. Gyro position is the black line, the robot training phase. Like in the vehicle training correlation with peak values at lags -2 and -3. (b) TedriS robot training path. It includes a wide range of paths. TedriS training position is the white dashed line. (c) Temporal correlation between ωm_t and γm_t from the TedriS robot training phase. A strong correlation is revealed at lag -3.

Figure 8: Vehicle and Robot Temporal correlation (a,c) and TedriS Robot training phase (b).

performance evaluation (Section 4.1), the entire DR phase, including the update phase and the predicting phase with two different robot pathways is depicted in Fig. 7. Proportionally, the threshold red line at 0.159 m respects the rate between robot and vehicle rear track width (0.25/1.57). All the DR phase experiments carried out with the robot, show more stable behavior and better DR navigation accuracy than vehicle DR phases, as depicted in Fig. 7a where the robot performs DR update phase trajectory keeping E_{DR} below the red mark. This is due to better training data, a controlled environment, and constant speed that we can achieve with robot platforms but bit easily with a car. The minimum E_{DR} was yielded with a $D_{CF} = 0.00005$, quite different of $\mu = -0.0047$. To simulate a common vehicle trajectory, the first DR predicting phase path was designed with straight paths, light and sharp curves. TedriS kept the DR error below the red mark during almost 60 s (Fig. 7b). The second DR predicting phase trajectory is an inverse version, but exhibited a slightly lower accuracy (Fig. 7c).

5 CONCLUSION AND FUTURE WORK

This work presented TedriS, a dead reckoning navigation system designed to enable any vehicle, independent of brand, model, cost or age, to have a DR navigation system, a gap left open by current DR systems that only address new high-end car models. To achieve this goal, TedriS uses an ordinary GPS as navigation device (or a gyroscope) and vehicle rear speed wheel sensors captured from CAN interface combined with a novel absolute position prediction technique. The core of TedriS is the relationship estimation between GPS and CAN yaw rate by a time series regression model using small blocks of data. This approach combined with parameters acquired dynamically during the DR update phase is responsible to set TedriS performance within the range achieved by state-of-art DR navigation systems, confirmed by the error margin acquired in the two road stretches of the DR predicting phase.

The experiments with the robot, attest the TedriS efficiency to predict localization in DR navigation. Analyzing proportionally, the results with the robot presented error margins below the results with the vehicle. The controlled environment, the smaller number of embedded devices and the constant speed, contributed that the

system reaches out these levels. Nevertheless, experiments with a robot proved that TedriS methodology can be applied to any type of vehicle on wheels. As future work, we intend to use other techniques like multiple linear regression models and autoregressive models with exogenous input in order to improve TedriS performance. Our main goal is to achieve sub-metric error margins for longer times.

REFERENCES

- [1] H. Ahmed, M. Tahir, and K. Ali. 2016. Terrain Based GPS Independent Lane-Level Vehicle Localization Using Particle Filter and Dead Reckoning. In *VTC-Fall*.
- [2] I. Belhajem, Y. Ben Maissa, and A. Tamtaoui. 2017. A Hybrid Machine Learning Based Low Cost Approach for Real Time Vehicle Position Estimation in a Smart City. In *Advances in Ubiquitous Networking 2*. Singapore.
- [3] M. Boufadene, A. Rabhi, M. Belkheiri, and A. Elhajjaji. 2016. Vehicle online parameter estimation using a nonlinear adaptive observer. In *American Control Conference (ACC)*. 1006–1010.
- [4] B. Chandrasekaran, S. Gangadhar, and J. M. Conrad. 2017. A survey of multisensor fusion techniques, architectures and methodologies. In *SoutheastCon 2017*. 1–8.
- [5] A. Gasparrini and B. Armstrong. 2018. Time Series Regression Analysis. (2018). Retrieved March 10, 2018 from <http://csm.lshmt.ac.uk>
- [6] P. Goebel. 2014. *ROS By Example*. Lulu.
- [7] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, and J. Wang. 2014. Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application. (2014).
- [8] F. Jiménez, J. E. Naranjo, J. J. Anaya, F. Garcia, A. Ponz, and J. M. Armingol. 2016. Advanced Driver Assistance System for Road Environments to Improve Safety and Efficiency. *Transportation Research Procedia* 14 (2016), 2245 – 2254.
- [9] U. Kiencke and L. Nielsen. 2000. *Automotive Control Systems: For Engine, Driveline and Vehicle* (1st ed.). Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [10] E. I. Laftchiev, C. M. Lagoa, and S. N. Brennan. 2015. Vehicle Localization Using In-Vehicle Pitch Data and Dynamical Models. *IEEE Transactions on Intelligent Transportation Systems* 16, 1 (Feb 2015), 206–220.
- [11] G. Mahalakshmi, S. Sridevi, and S. Rajaram. 2016. A survey on forecasting of time series data. In *ICCTIDE*.
- [12] Inc Open Source Robotics Foundation. 2014. Turtlebot2 Robot. (2014). Retrieved March 1, 2018 from <http://turtlebot.com>
- [13] E. Ozkan, C. Lundquist, and F. Gustafsson. 2011. A Bayesian approach to jointly estimate tire radii and vehicle trajectory. In *ITSC*.
- [14] Shumway Robert H. and Stoffer David S. 2017. *Time Series Analysis and Its Applications* (4st ed.). Springer International Publishing.
- [15] Inc U-blox america. 2018. U-blox-M8. (2018). Retrieved March 1, 2018 from <https://www.u-blox.com/en/product/neo-m8-series>
- [16] S. Zair, S. Le Hégarat-Masle, and E. Seignez. 2016. A-Contrario Modeling for Robust Localization Using Raw GNSS Data. *IEEE Transactions on Intelligent Transportation Systems* 17, 5 (may 2016), 1354–1367.
- [17] T. Zhang and X. Xu. 2012. A new method of seamless land navigation for GPS/INS integrated system. *Measurement* 45, 4 (2012), 691 – 701.