Paper:

Sensor Data Fusion of a Redundant Dual-Platform Robot for Elevation Mapping

Avi Turgeman*, Shraga Shoval, and Amir Degani*,*****

*Technion Autonomous Systems Program (TASP), Technion
Haifa 32000, Israel
E-mail: aviturg@gmail.com
**Department of Industrial Engineering and Management, Ariel University
Ariel 40700, Israel
E-mail: shraga@ariel.ac.il
***Faculty of Civil and Environmental Engineering, Technion
Haifa 32000, Israel
E-mail: adegani@technion.ac.il
[Received June 25, 2017; accepted November 14, 2017]

This paper presents a novel methodology for localization and terrain mapping along a defined course such as narrow tunnels and pipes, using a redundant unmanned ground vehicle kinematic design. The vehicle is designed to work in unknown environments without the use of external sensors. The design consists of two platforms, connected by a passive, semi-rigid three-bar mechanism. Each platform includes separate sets of local sensors and a controller. In addition, a central controller logs the data and synchronizes the platforms' motion. According to the dynamic patterns of the redundant information, a fusion algorithm, based on a *centralized Kalman filter*, receives data from the different sets of inputs (mapping techniques), and produces an elevation map along the traversed route in the *x-z* sagittal plane. The method is tested in various scenarios using simulated and real-world setups. The experimental results show high degree of accuracy on different terrains. The proposed system is suitable for mapping terrains in confined spaces such as underground tunnels and wrecks where standard mapping devices such as GPS, laser scanners and cameras are not applicable.

Keywords: signal estimation, data fusion, robotic sensing, robot control, robotic terrain mapping

1. Introduction

The use of Unmanned Ground Vehicles (UGVs) is common in many applications and environments. UGVs are suitable for a large variety of tasks in agriculture, search & rescue, and security missions, since they can perform tedious or dangerous tasks. For instance, a UGV can travel where humans are currently unable (e.g., Mars) or wish to avoid (e.g., mine fields). An increasingly popular mission for field robotics is terrain mapping. While in some scenarios it is possible to map the terrain using Unmanned Aerial Vehicles (e.g., [1]), the use of autonomous ground vehicles has significant advantage in others. Underground or covered terrains cannot be analyzed from the air, and the ability to sense and map the surface makes the UGV an appropriate candidate for such missions. A typical example might be the mapping of a collapsed structure after a major disaster in search and rescue mission. In such a scenario, the vehicle cannot utilize GPS data, and the performance of some sensors such as laser scanners and cameras may be limited due to the complex structure and constraints of the environment. Additional important application is mapping underground tunnels and pipes (e.g., water, sewage, oil, air-conditioning etc.) where humans cannot operate, and the use of external sensors is limited or even impossible.

One of the main challenges in the operation of ground vehicles is localization, and inaccuracy in localization leads to a significant error in mapping. Non-ideal surface conditions and external effects such as weather and/or surface irregularities increase the odds for vehicle slippage, which is undetected by a standard odometric method. In order to reduce the effect of those disturbances, models for slippage estimation were developed. One model proposes investigating the kinetic behavior of a planetary rover, with attention to tire-soil traction mechanics and articulated body dynamics when the rover travels over natural rough terrains [2]. A control method uses the estimated slip ratio value to minimize odometric errors and to limit excessive tire force. Therefore, the rover is able to successfully traverse over obstacles without digging into the soil. Another method presents an algorithm for determining the nominal driving track forces of motions along a specific path at a desired speed [3]. The latter requires sufficient information regarding the terrain explored by the vehicle. This information may be unavailable or inaccurate when traveling along an unknown or complex environment. A different approach controls each of the vehicle's wheels independently [4]. Each wheel is equipped

with a motor and a rotation sensor such that the vehicle's maneuverability increases while reducing odometric error. Moreover, an additional gyroscope is placed on the vehicle to further reduce the odometric errors. Successful terrain mapping requires particular algorithms and methods such as SLAM (Simultaneous Localization and Mapping) [5, 6]. SLAM is a process that simultaneously builds a map of an unknown environment and determines the vehicle's location in that map. First, the vehicle's controller constructs a map of its surrounding, and then determines its location by identifying distinct or well-defined features in the terrain, called landmarks. The identification and estimation of the landmarks' position relative to the vehicle is crucial for the process. The vehicle constantly takes relative observations of the landmarks, while moving in the unknown environment. These landmarks are correlated with each other, and an error in the identification and/or localization of one landmark affects the other [7, 8]. One of the solutions for enhanced vehicle position estimation is the use of Extended Kalman Filter (EKF) in the SLAM process. The basis for the EKF-SLAM method is to determine the vehicle's motion with a kinematic and an observation (measurements) models. Each model contains Gaussian disturbance/error parameters that evaluate the "noise" level of the model and the observation. Each time an observation is made, the landmarks and the model parameters are updated.

Here we survey only several SLAM applications out of many. Applications using SLAM with range-only sensors [9–11] and bearing only sensors [12, 13] show that a single measurement is insufficient for a reliable landmark localization. A ground robot, equipped with a 2-D Laser Range Finder (LRF), mounted at a fixed angle facing downwards, is examined for use in terrain mapping [14]. In this system, a mapping algorithm generates a 3-D map based on the range information using Hidden Markov Models. Each step is classified as navigable or un-navigable, producing a complete map that shows the safe and unsafe areas for traveling. A distributed method is proposed in [15] by using map sharing for estimating locations and improving a global map building process.

Depth cameras are also commonly used for terrain mapping. A robot, equipped with two CCD (2-D) and one PMD (Photonic Mixer Device – able to capture 3-D static images) cameras [16] is used for mapping interior rooms and hallways. Another terrain mapping method uses stereo vision technique [17, 18]. The technique produces a quality map for far objects and wide areas. For low surface terrain, such as narrow spaces, models including additional sensors such as LRF [19] provide proper solutions. A set of binary detectors [20] have been implemented to identify different objects and surface changes, producing a compact terrain map from each frame of the stereo images. The mapping algorithm labels cells that contain obstacles as no-go regions, and encodes terrain elevation, terrain classification, terrain roughness, traversability cost, and a confidence value. The single frame maps are then merged into a world map, where temporal filtering is applied. In scenarios where geometric models are difficult to measure, different solutions such as vector-field SLAM [21] and Bayesian-field SLAM [22] are proposed, where instead of using geometric models, measurable spatial fields are exploited. Another solution for a large-scale terrain is using a Gaussian-process modeling approach [23]. When sensor information is corrupted or incomplete, a Gaussian-process is applied for estimating and interpolating information across the field. The process uses the spatial correlation of the given data points to estimate the elevation values for unknown points of interest.

There is a growing interest in the SLAM research in using only interoceptive information for localization and mapping, as some applications and environments do not allow the use of exteroceptive sensors. The dependency on external sensors such as GPS or cameras is inapplicable in many cases. As a typical example consider the Spirit Mars exploration rover. The rover was equipped, among other sensors, with a stereovision panoramic camera (PanCam). The PanCam contained two CCD high resolution cameras with special filters and infra-red sensors. In addition, the rover had two pairs of stereo black-and-white cameras under the solar panels facing to the front and the back, and a navigation camera (NavCam) with elevated 45° field of view for looking ahead. Despite this wide range of cameras and sensors, Spirit got trapped in a soft sand soil, that eventually brought the mission to its end.

The work presented in this paper uses an autonomous system, which utilizes only interoceptive sensors for terrain mapping by implementing a novel data fusion approach. As described above, the traditional SLAM methods fuse interoceptive sensing, typically odometry and IMU with exteroceptive sensing (e.g., Lidar [24, 25], camera, or depth images) to find the current pose. It then matches the information from the interoceptive and exteroceptive sensors to build a map. In our method, we enhance the interoceptive sensing by combining the redundancy of a novel vehicle with two sets of sensors and the kinematic relationship between them. This way we receive an elevation map of the terrain while reducing the localization error.

2. Hardware Configuration

In this section, we describe the hardware configuration of the UGV that enables the proposed localization and mapping methodologies.

2.1. The Redundant Dual-Vehicle Robot

The current work utilizes a design similar to the one used in [26]. This design is suitable for our proposed test experiments for algorithm validation. Different platforms (size and structure) may be applied using the proposed approach, depending on the task in hand.

The system (**Fig. 1**) consists of two platforms connected with an articulated mechanism.

Fig. 1. The research prototype. Platform size is $34 \times 40 \times 25$ cm with a 64 cm long connecting beam.

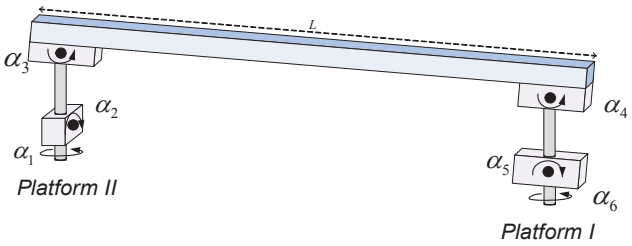

Fig. 2. The 6-DOF mechanism contains two joints for platforms' rotation, three joints for pitch movement and one joint for roll.

The connecting mechanism is a three-beam structure (**Fig. 2**) in which a rotation sensor (e.g., encoders or potentiometers) continuously measures the position (angle) of each joint. This mechanism allows relative motion measurements of each platform in relation to the other platform.

The connecting mechanism consists of six revolute joints, each with limited stroke, to enable limited independent motion of each platform. The three joints located near to the leading platform (Platform I in **Fig. 2**) enable yaw rotation of the platform along the vertical axes (α_6) and pitch rotation along the horizontal axis (α_4, α_5). The three joints located near to the rear platform (Platform II) enable pitch rotation along the horizontal axes (α_3), yaw rotation along the vertical axis (α_1) and a roll rotation along the vertical axes (α_2). This connecting mechanism allows a 6-DOF relative articulation of each platform, regardless of the motion of the other platform, particularly due to unexpected changes in the surface. For instance, if the robot reaches a ramp, joints' angles α_4, α_5 of the leading platform and α_3 of the rear platform are affected. If, in another scenario, only one side of a platform climbs over a short obstacle (e.g., a rock), then α_2 is affected. Similarly, if the leading or the rear platform rotate (e.g., due to slippage), then α_6 or α_1 change accordingly. In addition, each platform contains local sensors such as encoders attached to the driving wheels for the odometric model, and inclinometers and gyroscopes (embedded in an Inertial Measurements Unit (IMU)), which provide data on the platforms' orientation. This redundant design provides several data sets (inputs to the fusion model) to produce an accurate localization and terrain mapping.

2.2. The Control System

The control system consists of three control units: Two Local Units (LUs) and one Centralized Control Unit (CCU). The LUs control the platforms' motion and execute the odometric calculation for estimating the platforms' localization (based on the wheels' dynamics). The CCU receives the data from each platform via a Wi-Fi network, and coordinates the motion of both platforms by sending real-time commands. These commands improve the robot's maneuverability and adjust the platforms' motions in case of a slippage. In addition, the CCU logs the data received from all the sensors in the system. The control architecture and the design of the robot provide three modes of operation. The first two modes are manual: Dual and Single modes. In the Dual mode, the operator controls both platforms with the same command, while in the Single mode the operator controls each platform separately. These two modes are used for manually assisting the robot to overcome special obstacles and to perform sharp maneuvers in cluttered surroundings. The third mode is the Autonomous mode, which serves as the main operational mode. In this mode, the CCU sends motion commands to the two LUs according to the updated sensory data. The CCU identifies the relevant features of the terrain and the robot's behavior, and changes the motion pattern according to the required mission. For example, if the robot is required to travel along a straight line, the CCU corrects odometric errors by adjusting the motion of both platforms in order to maintain accurate trajectory. Alternatively, if the robot is required to perform a lateral or angular motion, the CCU determines the required motion from each platform, and verifies that these motions are executed properly.

2.3. Sensor Data Acquisition

Odometric models [27] are being used to estimate the platforms' positions based on the left and right encoders placed on each wheel. The platforms' velocities are determined by the derivation of the momentary position of the wheels. In addition, a 6-DOF IMU provides three absolute angles of yaw (ψ), pitch (θ) and roll (φ) and their respective derivatives. The raw data from both platforms are restored and processed (using the data fusion algorithm) in the CCU for producing the terrain map (both on/off-line), as described in the following section. In our approach, the platforms do not directly communicate with each other, although it is possible but due to the centrality of the algorithm we choose to handle the data in the external computer.

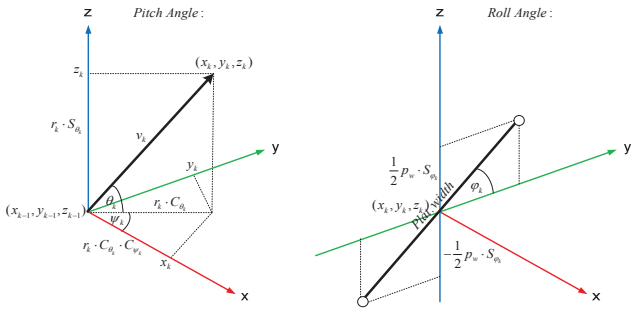

Fig. 3. Pitch and roll projection of robot position in world coordinate system.

3. Elevation Map Construction

As previously mentioned, this research focuses on constructing an elevation map in the x-z sagittal plane, assuming motion in the y-direction is constrained (e.g. inside a narrow tunnel or a pipe). Given the dual platform mechanism described in the previous section, the position coordinates of platform i in the x-z sagittal plane are given by

$$\begin{cases} x_{i,k} = x_{i,k-1} + v_{i,k-1} \cdot \cos(\theta_{i,k-1}) \\ z_{i,k} = z_{i,k-1} + v_{i,k-1} \cdot \sin(\theta_{i,k-1}), \end{cases} \quad \ldots \quad (1)$$

where k is the current time step, $v_{i,k-1}$ is the velocity command and $\theta_{i,k-1}$ is the pitch angle of the platform relative to the horizon, computed by the IMU for platform i. $i = 1,2$ refers to leading and rear platforms respectively (**Fig. 3**).

Using data from the connecting mechanism, an estimation of the position of platform 1 (leader) relative to platform 2 (rear), denoted by $(x_{21,k}, z_{21,k})$, is determined using forward kinematics, given by the homogeneous transformation matrix

$$\begin{cases} T_2^6 = T_5^6 T_4^5 T_3^4 T_2^3 = \begin{bmatrix} & R_2^6 & & d_2^6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ (x_{21,k}, z_{21,k}) = T_2^6 \cdot (x_{2,k}, z_{2,k}), \end{cases} \quad \ldots \quad (2)$$

where each homogenous transformation matrix T_i^j is composed of a rotation matrix R_i^j and a translation vector d_i^j. The position of the rear platform $(x_{2,k}, z_{2,k})$ is given by its odometric model (Eq. (1)). We call this transformation the *Forward Kinematic Transform* (FKT). Once this calculation is performed, two planner maps (in the x-z plan) are produced – the leading platform map produced by its odometric model, and the FKT map.

A third map is produced using a *Time-Transform* (TT) method. Since each platform constructs its own map along the travelled trajectory using its IMU and odometric model, and assuming that the rear platform follows the trajectory of the leading platform, each point along the trajectory is "visited" by both platforms with a time lag between them. The two maps, with the time lag can be used to generate a third map. For instance, if at time (step) $k = 10$ the leading platform reached the location of $(x_{1,10}, z_{1,10})$, after time lag τ, the rear platform reaches position $(x_{2d,10}, z_{2d,10}) = (x_{2,10+\tau}, z_{2,10+\tau})$ and in an environment with no disturbances $(x_{1,10}, z_{1,10}) = (x_{2d,10}, z_{2d,10})$.

In a more general way

$$\begin{aligned} (x_{2d,k}, z_{2d,k}) &= TT(x_{2,k}, z_{2,k}) \\ &= \begin{cases} (x_{2,k+\tau}, z_{2,k+\tau}), & k \le k_{\max} - \tau \\ (x_{1,k}, z_{1,k}), & \text{else.} \end{cases} \end{aligned} \quad (3)$$

In an "*ideal*" world, the three maps are identical. However, this is not the case in most scenarios. Due to the sensitivity and the noise level of the sensors, tolerances and differences in the mechanical structure (for example - different air pressure in the wheels), and odometric effects (longitudinal and lateral slippage, bumps, holes etc.), the three maps are different. In order to obtain a more reliable elevation map of the terrain, there is a need to fuse the data gathered from all sensors' sets. For that, we use a *Centralized Kalman Filter* (CKF), as discussed in the following section.

4. Centralized Kalman Filter (CKF)

4.1. Fusion Algorithm

A *Discrete Kalman Filter* (DKF) is suitable for data fusion and filtering of systems in which the models and observations can be described by a linear connection, or when a model linearization has minimum effect on the results. In our system, the robot's motion model is non-linear (e.g., forward kinematics is trigonometric), therefore we use an extension of the DKF known as *Extended Kalman Filter* (EKF). EKF is used in problems that require filtering and data fusion from different sensors with non-linear relations between their state vector variables. Sensor fusion adds some requirements compared with the standard Kalman Filter method, and several modifications have been suggested [28, 29] (i.e., sensor fusion model and estimated state mean). In our system, the CKF method is chosen due to its adaptive fusion technique. The CKF calculates the mean of the Kalman gain and the measurement vector differential equation. The Kalman gain represents the relations between the relative measurement and the predicted state estimate. When the gain is high, the filter increases the weight of the measurements; otherwise, it follows the model predictions more closely. The CKF structure is more flexible and allows the algorithm to perform an evaluation of the measurements quality, and therefore to adjust its effect at each step. By doing so, the CKF reduces the total error, as will be shown in the next sections. In our current system, the state and measurement vectors are governed by the non-linear stochastic difference equations [30]:

$$\begin{cases} s_k = f(s_{k-1}, u_{k-1}, w_{k-1}) \\ m_k = h(s_k, v_k), \end{cases} \quad \ldots \quad (4)$$

where w_{k-1} and v_k are random variables representing the process and measurements noises. In practice, those values are unknown at each step, and the following approximated state model is being used:

$$\begin{cases} \hat{s}_k = f(\hat{s}_{k-1}, u_{k-1}, 0) = f(\hat{x}_{k-1}, \hat{z}_{k-1}, \hat{\theta}_{k-1}, \nu_{k-1}) \\ \hat{m}_k = h(\hat{s}_k, 0), \end{cases} \quad (5)$$

where $\hat{x}_{k-1}, \hat{z}_{k-1}$ are the robot's positions along the trajectory, $\hat{\theta}_{k-1}$ is the pitch angle and ν_{k-1} is the velocity command. The state function f is according to Eq. (1) and the measurement function h is simply $h(\hat{s}_k, 0) = [\hat{x}_k, \hat{z}_k]^T$. To estimate a process with non-linear difference and measurement relationships, a new set of linearized equations are developed:

$$\begin{cases} s_k \approx \hat{s}_k + A \cdot (s_{k-1} - \hat{s}_{k-1}) + W \cdot w_{k-1} \\ m_k \approx \hat{m}_k + H(s_k - \hat{s}_k) + V \cdot v_k, \end{cases} \quad . \ . \quad (6)$$

where s_k, s_{k-1} and m_k, m_{k-1} are the actual state and measurement vectors, $\hat{s}_k$, $\hat{s}_{k-1}$ and $\hat{m}_k$, $\hat{m}_{k-1}$ are the approximate state and measurement vectors and w_{k-1}, v_k are the process and measurement noise random variables. The matrices A, W, H, V are the Jacobean matrices of the partial derivative of f with respect to s, w and h with respect to s, v:

$$\begin{cases} A^k_{i,j} = \dfrac{\partial f_i}{\partial s_j}(\hat{s}_{k-1}, u_{k-1}, 0), \quad W^k_{i,j} = \dfrac{\partial f_i}{\partial w_j}(\hat{s}_{k-1}, u_{k-1}, 0) \\ H^k_{i,j} = \dfrac{\partial h_i}{\partial s_j}(\hat{s}_k^-, 0), \qquad V^k_{i,j} = \dfrac{\partial h_i}{\partial v_j}(\hat{s}_k^-, 0), \end{cases}$$

$$. \ . \quad (7)$$

These matrices are calculated at each step for every variable in the state vector. The CKF algorithm is implemented in two steps per iteration (**Fig. 4**).

1. A-Priori Estimation (Prediction):
The a-priori estimation step is given by

$$\begin{cases} \hat{s}_k^- = f(\hat{s}_{k-1}^-, u_{k-1}, 0) \\ P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T, \end{cases} \quad . \ . \ . \ . \quad (8)$$

where $\hat{s}_k^-$ and P_k^- are the a-priori state vector and error covariance estimates at step k respectively. The matrices A_k and W_k are the Jacobean matrices of the partial derivatives of f with respect to $\hat{s}_k^-$ and w_k respectively. w_k is the process white noise with normal probability distribution of $p(w_{k-1}) \sim N(0, Q_k)$, where Q_k is the process noise covariance.

2. Post-Priori Estimation (Measurement update):
The post-priori estimation step is given by

$$\begin{cases} K_{i,k} = P_k^- H_k^T (H_k P_k^- H_k\, T + V_k R_{i,k} V_k^T)^{-1} \\ \hat{s}_k = \hat{s}_k^- + \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} K_{i,k} \cdot \left(m_{i,k} - h(\hat{s}_k^-, 0)\right) \\ P_k = \left(I - \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} K_{i,k} H_k\right) P_k^-, \end{cases} \quad (9)$$

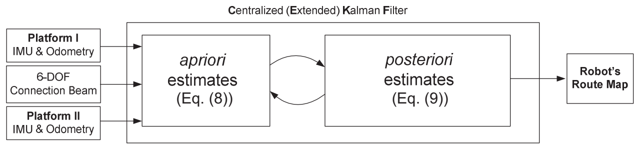

Fig. 4. CKF algorithm procedure.

where $K_{i,k}$ is the Kalman gain (*blending factor*) that minimizes the posteriori error covariance for each mapping technique ($i = 1 : N$). In our case, $N = 3$: Leading platform, Rear platform *Forward-Kinematics Transform* (FKT) and Rear platform *Time-Transform* (TT). $\hat{s}_k$ and P_k are the post-priori state and error covariance estimates at step k respectively. The measurement vector $m_{i,k}$ is determined according to Eqs. (1), (2), and the time-lag (TT)

$$\begin{cases} m_{1,k} = (x_{1,k}, z_{1,k}) \\ m_{2,k} = (x_{21,k}, z_{21,k}) = FKT(x_{2,k}, z_{2,k}) \\ m_{3,k} = (x_{2d,k}, z_{2d,k}) = TT(x_{2,k}, z_{2,k}), \end{cases} \quad . \quad (10)$$

where FKT is the Forward Kinematics Transform and TT is the Time Transform.

The difference $\left[m_{i,k} - h(\hat{s}_k^-, 0)\right]$ is called the measurement innovation (or *the residual*). The measurement noise covariance matrix $R_{i,k}$ is defined by

$$R_{i,k} = \begin{bmatrix} \sigma^2_{i,x_k} & 0 \\ 0 & \sigma^2_{i,z_k} \end{bmatrix}, \quad . \ . \ . \ . \ . \ . \ . \ . \ . \ . \ . \quad (11)$$

where σ^2_{i,x_k} and σ^2_{i,z_k} are the measurement noise covariance parameters in step k for each platform.

The matrix $R_{i,k}$ is dynamic and is updated during the process. The covariance represents the reliability of the measurement – the higher it is, the less confident the current measurement is. During the process, the covariance parameters are updated according to the following parameters:

- $\sqrt{z^2_{i,k} - z^2_{i,k-1}} > \varepsilon_1$: The error between the current and the previous steps. The error (ε_1) is subjected to a significant difference between the two steps (e.g., due to noisy measurement). This verification applies for each platform independently.

- $\sqrt{z^2_{i,k} - z^2_{j,k}} > \varepsilon_2$: The error between the different measurements according to each mapping technique. This calculation is performed at each step. Once the differences are higher than an empiric parameter (ε_2), all three $R_{i,k}$ are being affected by increasing the standard deviation values.

The matrices H_k and V_k are the Jacobean matrices of the partial derivatives of h with respect to $\hat{s}_k^-$ and v_k respectively. v_k is the measurement white noise with normal probability distribution of $p(v_k) \sim N(0, R_k)$.

As described above, the fusion procedure is defined by the measurement noise covariance matrix $R_{i,k}$ and the Kalman gain $K_{i,k}$ for each platform. Averaging is conducted to calculate the post-priori state and error vectors

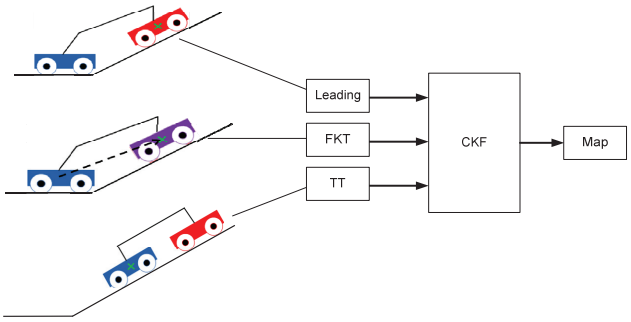

Fig. 5. Mapping procedure scheme relatively to the leading platform.

$\hat{s}_k$ and P_k. Later, we show that the mapping errors produced by the three techniques are reduced by adapting the correlation parameters in the CKF algorithm. The measurement noise covariance matrix $R_{i,k}$ considers the weight of each platform in the mapping procedure at the current step. **Fig. 5** illustrates the three data inputs of the CKF fusion algorithm. In this figure, the parallel inputs are divided as follows. In the top input, the front platform produces the leading platform's map. In the middle input, the front platform performs the forward kinematics transformation, estimating the position of the leading platform based on the connecting mechanism data and the position of the rear platform. In the bottom input, the rear platform represents the time transformation, i.e., the measurements of the rear platform once arriving at the position of the leading platform after a time-lag.

4.2. Adaptive Model Parameters for Improved Mapping

The CKF equations, as described above, are updated every time step:

$$[s_k, P_k] = CKF\left(f, s_{k-1}, P_{k-1}, h, m_k, Q, R, w_k, v_k\right), \quad (12)$$

where the measurement noise covariance matrix $R_{i,k}$ (Eq. (11)) is dynamic. In order to further enhance the performance, an adaptive velocity command is implemented. Due to gravity and the coupling mechanism between the platforms, the robot consumes more power while climbing compared with moving down, resulting in different speed. In order to maintain constant speeds, the CCU interferes with the velocity commands of the LUs when a slope (slows down) or an incline (increases speed) are detected by the IMU system.

5. Experimental Results

This section describes the experimental results of the proposed system. The experiments consist of dynamic simulations and actual experiments in laboratory and real world scenarios. For validation, the mapping algorithm output is compared with the actual terrain geometry. In addition, the algorithm performance is compared with the leading platform odometric mapping results, which serve as a benchmark performance value.

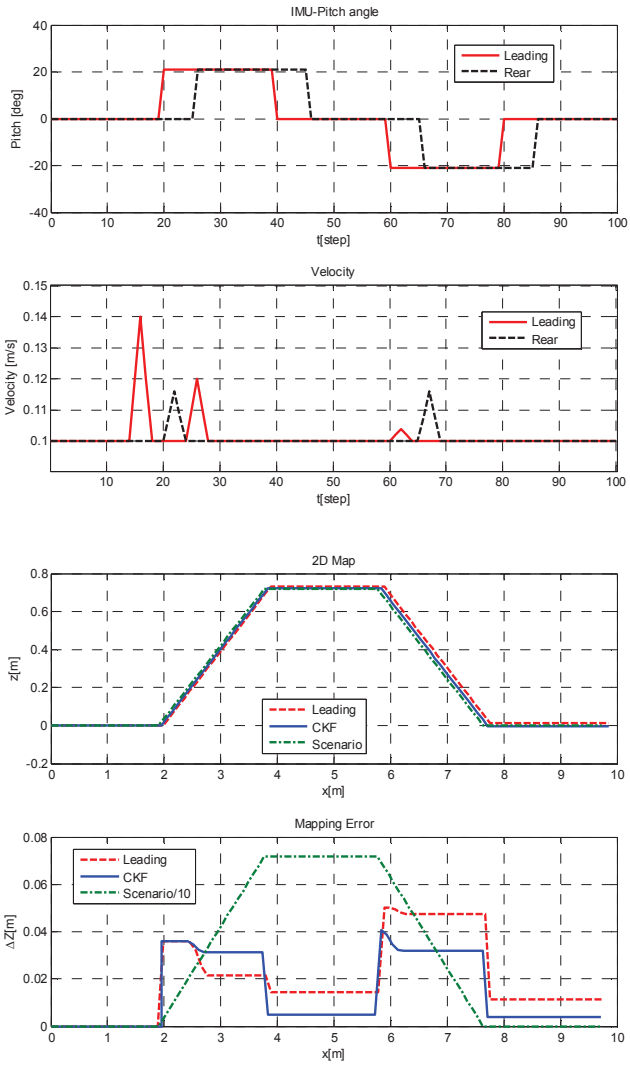

Fig. 6. Multiple slippages simulation. The algorithm manage to reduce the error and produces a reliable map.

5.1. Simulation Results

This section presents simulation results of several scenarios, analyzing the sensitivity of the mapping algorithm to odometric errors such as slippage, and to sensors errors such as bias and drift. In addition, the CKF algorithm calibration and robustness are examined as a preparation for the actual real-world experiments.

First, a simulation with multiple slippages of the leading and rear platform (**Fig. 6**) is conducted while traversing over a bump. This simulation examines the effects of multiple slippages on the raw data and the way the algorithm manages to reduce the total errors. In this scenario, the differences between the maps produced by the leading platform and the CKF are minor, with small error reduction by the CKF, mainly at the end of the bump. In the next simulation set, a white noise of the IMU simula-

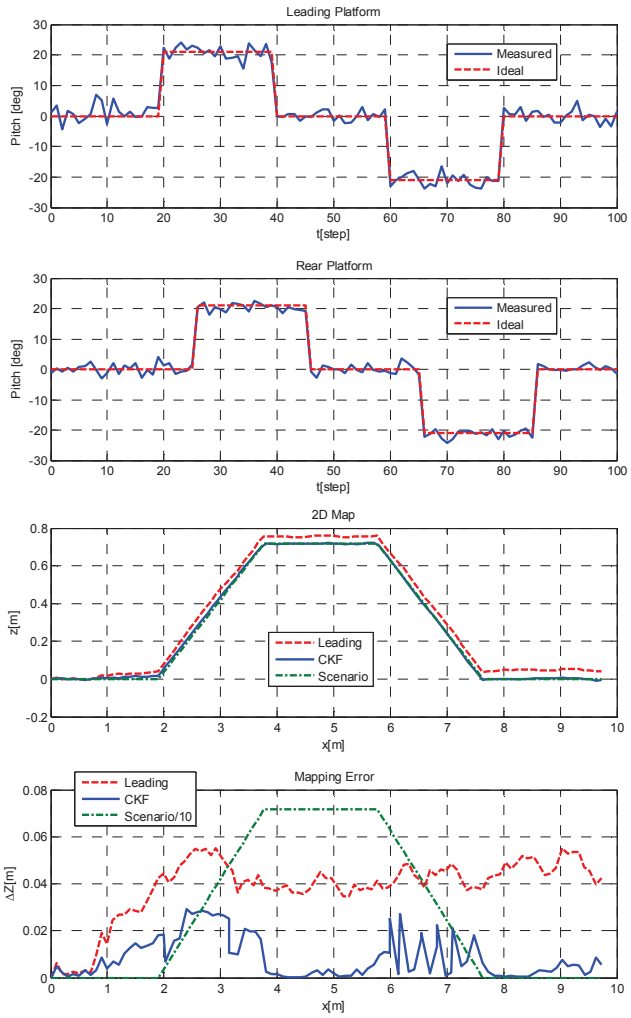

Fig. 7. White noise in both IMU sensors. The CKF behaves as a low-pass filter which reduces the overall error.

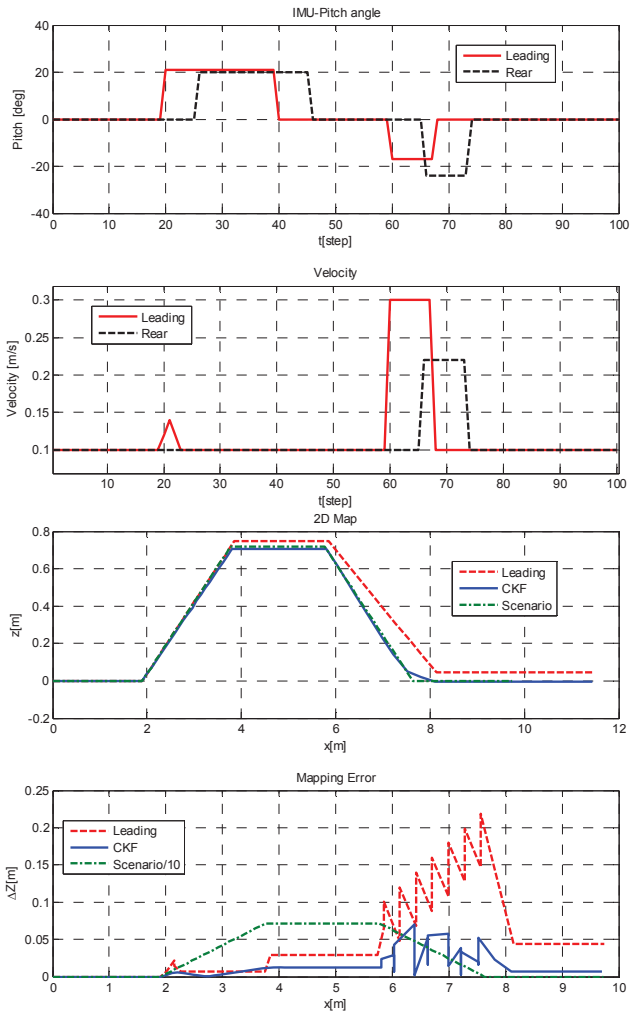

Fig. 8. Multiple errors – bias in IMU measurements and multiple slippages in both platforms. While the leading platform error is constantly increasing, the algorithm manages to maintain a significantly lower error.

tion (**Fig. 7**) is added to observe the performance of the CKF in case of noisy measurements. The CKF reduces the mapping error by over 200% due to the Kalman filter "original" characteristics. A simulation of multiple errors (**Fig. 8**) is a combination of the two former scenarios. It simulates multiple slippages for both platforms, combined with IMU errors. While the leading platform error increases, the CKF algorithm manages to maintain the error at a constant level. The final simulation examines a scenario of multiple bumps (**Fig. 9**). It consists of two bumps, each with a different geometry, and a horizontal section between them. This scenario is more challenging compared with the previous scenarios, due to slippage and pitch error measurements on both platforms. As observed in the previous simulation scenarios, the CKF algorithm presents better mapping results.

5.2. Indoor and Outdoor Experimental Results

This section describes the results of indoor and outdoor experiments. The purpose of these experiments is to examine the performance of the mapping algorithm in a realistic and controlled environment. At each scenario, after the robot completes its motion, the fusion-mapping algorithm produces an elevation map using the obtained raw data. Each scenario is repeated ten times and a map of the averaged results with standard deviation is produced.

The initial experimental setup includes a single bump, consisting of a 14.5 cm height with 20° slopes and a 2 m horizontal section (**Fig. 10a**). The total route distance is 3.7 m. Due to the bump's structure, the robot faces four points of significant-changes in the terrain. The mapping results of the leading platform and the CKF are shown in **Fig. 11**. The leading platform obtains different results when repeating this scenario, but the CKF manages to maintain similar outputs. In the multiple bumps scenario, the terrain consists of two different bumps. The first bump consists of 11.5 cm high with 11° slope and a 2.5 m horizontal part, and the second bump consists of 14.5 cm high with 20° slope and a 2 m horizontal section (**Fig. 10b**). Again, each experiment is repeated ten times at each direction. The combined mapping results are shown in **Fig. 12**. As shown, the CKF algorithm re-

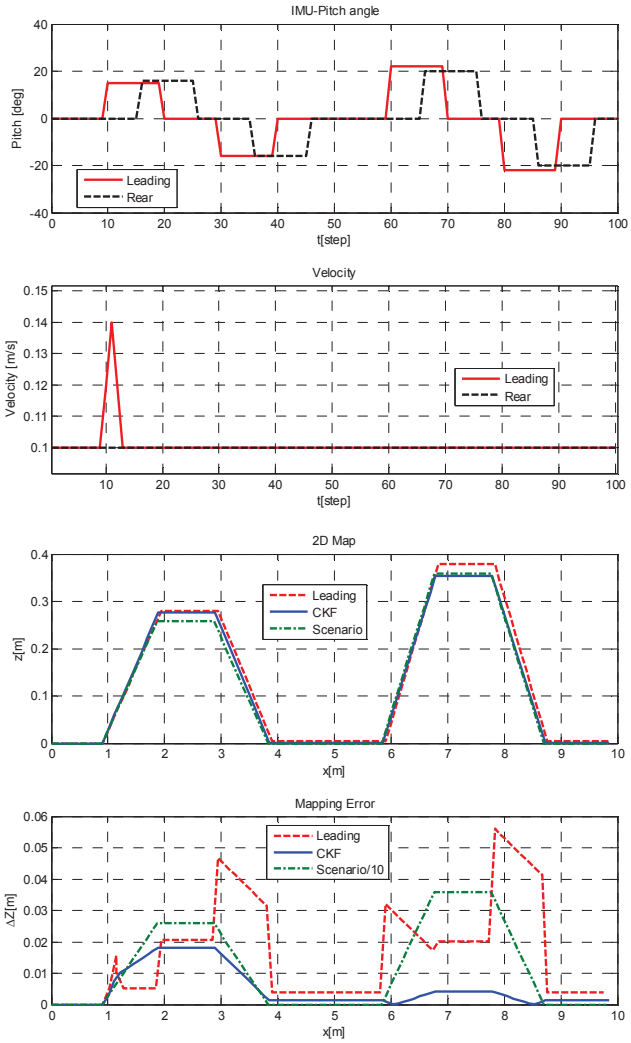

Fig. 9. Multiple bumps simulation including errors such as sensors bias and slippage. Here, the CKF manages to decrease the error toward the second bump. In addition, the simulation shows that the error is reduced after crossing the bump due to opposing measurements (incline vs. decline).

Fig. 10. Experiments setup. Single bump (a) flat wooden plate placed of a flat road. Multiple bumps (b) the first bump is lower and longer than the second bump. Rough terrain (c) the gravel ground is constantly changing making it difficult for mapping. The terrain includes three different bumps (height and width).

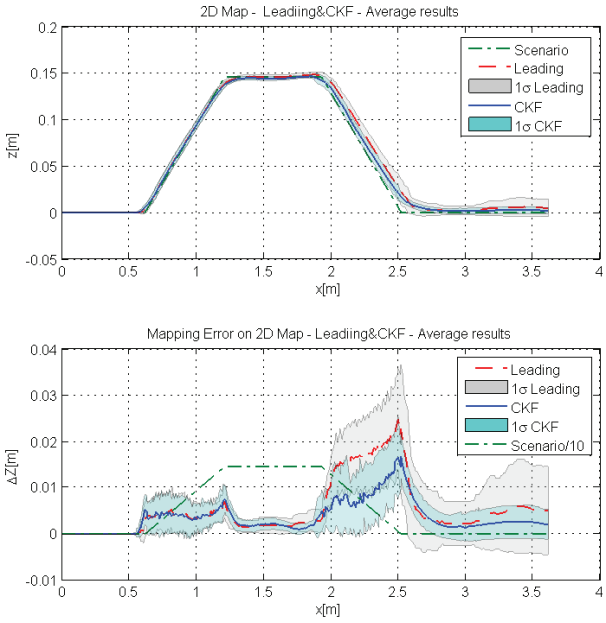

Fig. 11. Single bump mapping results. The dashed and solid lines represent the average result of the leading platform and the CKF respectively. The grey and dark grey areas around those lines represent the standard deviation. The significant differences between the leading platform and the CKF is significant in the bump's slope (greater than 60%). The average mapping error of the CKF is lower than the leading platform by around 50%.

duces the mapping error compared with the leading platform results.

The final experiment is conducted in a rough terrain environment set-up (**Fig. 10c**) along a 16 m trajectory that includes three natural bumps. After the robot completes its motion, the travelled surface geometry is accurately measured, using an orthogonal surface height measurement device. Due to the rough terrain, the robot tends to slip, resulting in noisy IMU measurements. **Fig. 13** shows the raw data inputs and the mapping results. Although the raw data has many errors, the CKF reduces these errors and produces a map close to the surface of the real rough terrain.

5.3. Discussion

The results of the experiments for a single (**Figs. 6** and **11**) and multiple bumps (**Figs. 9** and **12**) scenarios are similar to the simulation forecast. Similar behaviors in the slope area (peak at the beginning of the slope) and reduced error at the end of the bump are observed. As examined in the simulation, various errors such as slippages (**Fig. 6**) and noisy IMU measurements (**Figs. 7** and **8**) provide an explanation to this behavior. When slippage occurs during extreme change of the train, in addition to IMU measurement error during slope descending, the mapping error is reduced. This error reduction is caused by gravity and the gradient of the slope [31]. During motion along the slope, the robot accelerates, which results in a shorter descent time compared with ascending the inclination. This

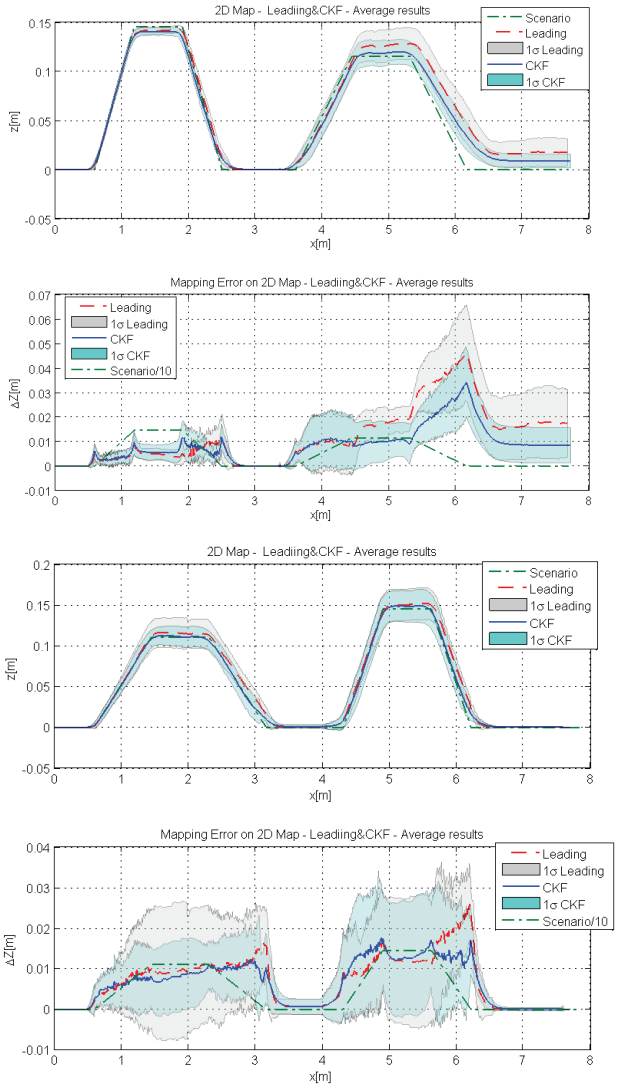

Fig. 12. Multiple bump mapping results. Again, both average error and standard-deviation are lower for the CKF case than the leading platform.

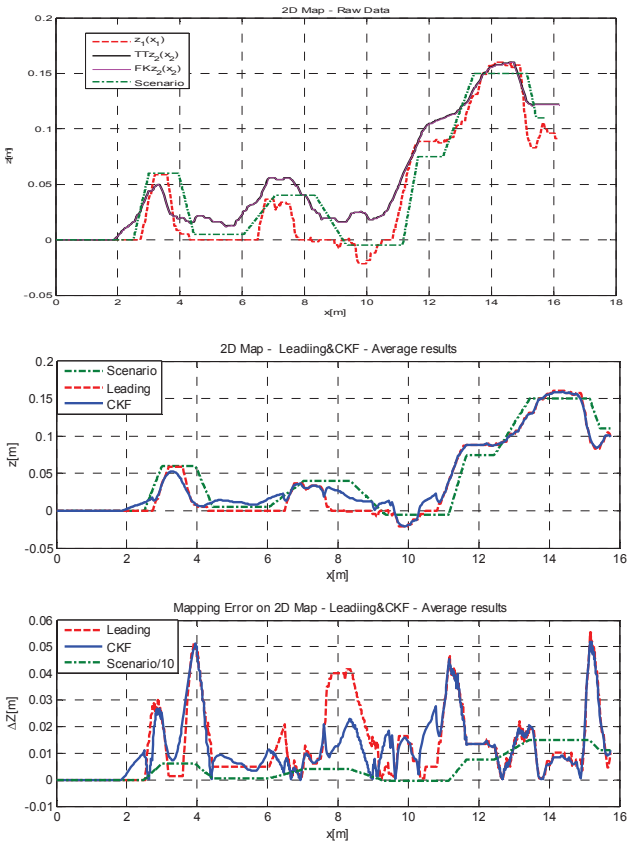

Fig. 13. Rough terrain experiment results. One can see that although the leading platform maps the second bump with high error, the CKF manages to map the bumps with small errors in the width and height.

produces a complementary bias, which reduces the total error.

When the robot reaches the highest point on the ramp, it experiences a sharp pitch angle change, and a significant error is observed. The reason for this phenomenon is the rigid robot structure, which makes corners' mapping a challenging task. Once the leading platform's front two wheels pass the corner, it has a momentary fall in the pitch direction until the two rear wheels pass the corner. It also produces a force on the rear platform due to the linking beam. This phenomenon repeats in all the scenarios. Due to the filter's characteristics, the CKF algorithm is less sensitive to noisy measurements, presenting better results in all scenarios. This ability also contributes to peaks measurements handling. Rough terrain experiment (**Fig. 13**) is performed to observe mapping behavior in more realistic conditions. This rough surface produces many slippages and sensors errors. However, the robot successfully maps the three different bumps in the terrain. In this experiment, the error plot includes many fluctuations. An explanation for this behavior is the measurements accuracy of the sensors. The sensors noise level and inner low pass filter affect the robot's measurement quality. The IMU sensor used in our experiments comes with small SNR (Signal Noise Ratio), low resolution and increased bias. When moving along a rough terrain, constant errors are measured and the algorithm needs to be selective.

A summary of the errors calculation, expressed by $z_{relative} = \frac{1}{N}\sum|z_{Real} - z_i|/z_{\max}$ [%], is presented in **Table 1**.

As shown, the CKF algorithm has 38% better mapping results than the leading platform. In addition, the CKF standard deviation (STDEV) is lower than the leading platform. This series of experiments and simulations prove the robustness and efficiency of the CKF mapping algorithm to the overall performance of the robot.

6. Conclusions

This research presents a method for terrain mapping along a pre-defined trajectory without external sensors, using two identical platforms, coupled by an articulated connecting mechanism. It specifically aims at detect-

Table 1. Experiments error.
{average [cm]}; (maximum [cm]); [relative [%]]

Scenario	CKF	Leading platform	Single robot	Constant velocity torque
Ramp	{0.5} (1) [3.33%]	{0.6} (1) [4%]	–	–
Bump	{0.36} (1.5) [2.4%]	{0.5} (2.4) [3.33%]	{0.9} (2.2) [6%]	Leading: {1} (1.5) [6.66%] CKF: {1.2} (2.2) [8%]
Multiple bumps	{1} (1.5) [6.66%]	{1.6} (3.5) [11%]	–	–
Crosswalk	{0.6} (6.5) [4%]	{0.8} (1.8) [5.33%]	–	–
Rough terrain	{2} (5) [13.33%]	{2.5} (5) [16.66%]	–	–
Total	{0.91} [6%]	{1.26} [8.4%]	–	–

ing irregularities in narrow environments such as underground tunnels and pipes where the use of external sensors is limited or impossible. During the terrain exploration, the robot gathers information from the two platforms and the connecting mechanism to generate three maps. The first two maps are constructed by using odometric data from the leading platform and the rear platform (time shifted), and the third map is constructed using the connecting mechanism and a forward-kinematics model. These three maps behave as input sets to a *Centralized Kalman Filter* (CKF) data fusion algorithm. By analyzing the quality of the measurements at each time step, the CKF grades the data according to the correlation factor, and estimates the current step values according to previous data. Once completed, an elevation map in the x-z plane of the terrain is produced. In order to examine the mapping performance of the CKF algorithm, a series of simulations and experiments with a prototype robot were conducted. The simulations include measurements and slippages errors in various environments, and multiple bumps and rough terrain surfaces. Based on preliminary results, the mapping algorithm was modified by performing adaptive velocity control and adaptive algorithm parameters, and the mapping results were compared with the results of a single platform as a reference. Based on these satisfactory results, we conclude that inertial mapping in challenging terrains is a viable solution in environments and scenarios that cannot utilize other navigation and interoceptive and exteroceptive sensing. The CKF algorithm is a suitable tool for data fusion in such mapping tasks. By including the dynamic model of the robot, along with slippage identification and classification, the enhanced CKF algorithm produces better mapping results. Moreover, the unique structure of the robot and the CKF algorithm design reduce the effect of slippages on the output results.

Future work will strive to construct a complete 3-D map by using a simple coverage algorithm (forward and backwards), or by utilizing multiple robots. For a complete 3-D map, additional data such as platforms' roll and yaw angles, along with calculation of the y-axis position of the robot, should be added to the estimation model as part of the state vector variables. It is imperative to distinguish between lateral movement of each platform according to the world coordinate and the relation between each other. However, this can be achieved by using the joints angles on the connecting beam in addition to the 6-DOF IMU data mounted on each platform. The measurement noise covariance matrix will then have an additional weight parameter concerning the y-axis and its effect on the estimation procedure and the output map.

When external sensing is possible, our method does not need to completely replace current methods such as LRF or stereo vision. The information using these methods can then be fused into our maps to increase accuracy and reliability. Including additional sensors with similar data can expand the estimation model with more inputs that will enhance the estimation procedure.

References:

[1] T. Suzuki, Y. Amano, T. Hashizume, and S. Suzuki, "3D Terrain Reconstruction by Small Unmanned Aerial Vehicle Using SIFT-Based Monocular SLAM," J. Robot. Mechatron., Vol.23, No.2. pp. 292-301, 2011.

[2] K. Yoshida and H. Hamano, "Motion Dynamics of a Rover with Slip-Based Traction model," Int. Conf. on Robotics and Automation, pp. 3155-3160, 2002.

[3] Z. Shiller and W. Serate, "Trajectory Planning of Tracked Vehicles," ASME J. of Dynamic Systems, Measurement and Control, Vol.117, Np.4, pp. 619-624, 1995.

[4] L. Ojeda and J. Borenstein, "Methods for the Reduction of Odometry Errors in Over-Constrained Mobile Robots," Proc. of the UGV Technology Conf. at the SPIE AeroSense Symposium, Orlando, FL, April 21-25, 2003.

[5] H. D. Whyte and T. Bailey, "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms," 2006.

[6] T. Bailey and H. D. Whyte, and T. Bailey, "Simultaneous Localization and Mapping (SLAM): Part II State of the Art," 2006.

[7] R. Smith, M. Self, and P. Cheeseman, "Estimating Uncertain Spatial Relationships in Robotics," Autonomous Robot Vehicles, Springer-Verlag, pp. 167-193, 1990.

[8] J. J. Leonard and H. D. Whyte, "Simultaneous Map Building and Localization for an Autonomous Mobile Robot," Proc. IEEE Int. Workshop on Intelligent Robots and Systems (IROS), pp. 1442-1447, Osaka, Japan, 1991.

[9] L. Doitsidis, A. Renzaglia, S. Weiss, E. Kosmatopoulos, D. Scaramuzza, and R. Siegwart, "3-D Surveillance Coverage Using Maps Extracted by a Monocular SLAM Algorithm," Intelligent Robots and System (IROS), IEEE/RSJ Int. Conf. San Francisco, 2011.

[10] J. J. Leonard and R. J. Rikoski, "Incorporation of Delayed Decision Making into Stochastic Mapping," Int. Symposium on Experimental Robotics, 2000.

[11] J. J. Leonard, R. J. Rikoski, P.M. Newman, and M. C. Bosse, "Mapping Partially Observable Features from Multiple Uncertain Vantage Points," Int. J. of Robotics Research, Vol.21, Issue 10-11, pp. 943-975, 2002.

[12] M. Deans and M. Hebert, "Experimental Comparison of Techniques for Localization and Mapping Using A Bearing-Only Sensor," Int. Symposium on Experimental Robotics, 2000.

[13] T. Bailey, "Constrained Initialisation for Bearing-Only SLAM," IEEE Int. Conf. on Robotics and Automation, Taipei, Taiwan, 2003.

[14] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard, "Autonomous Terrain Mapping and Classification Using Hidden Markov Models," Int. Conf. on Robotics and Automation, Barcelona, Spain, 2005.

[15] F. Hashikawa and K. Morioka, "Convenient Position Estimation of Distributed Sensors in Intelligent Spaces Using SLAM for Mobile Robots," J. Robot. Mechatron., Vol.27, No.2, pp. 191-199, 2015.

[16] A. Prusak, O. Melnychuk, and H. Roth, "Pose Estimation and Map Building with a PMD-Camera for Robot Navigation," "3-D PoseMap" project (KO-2044/3-1 and RO-2384/1-1), 2007.

[17] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey, "Outdoor Mapping and Navigation Using Stereo Vision," Proc. of Int. Symp. on Experimental Robotics (ISER), Rio de Janeiro, Brazil, 2006.

[18] A. A. Souza and L. M. G. Gonçalves, "3D Robotic Mapping with Probabilistic Occupancy Grids," Int. J. of Engineering Sciences & Emerging Technologies, 2012.
[19] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation," Autonomous Robots 18, pp. 81-102, Springer Science and Business Media, 2005.
[20] A. L. Rankin, A. Huertas, and L. H. Matthies, "Stereo Vision Based Terrain Mapping for Off-Road Autonomous Navigation," Unmanned Systems Technology XI, Proc. of the SPIE, Vol.7332, 2009.
[21] J. S. Gutmann, E. Eade, P. Fong, and M. E. Munich, "Vector Field SLAM – Localization by Learning the Spatial Variation of Continuous Signals," IEEE Trans. on Robotics, Vol.28, Issue 3, pp. 650-667, 2012.
[22] H. N. Do, M. Jadaliha, M. Temel, and J. Choi, "Fully Bayesian Field Slam Using Gaussian Markov Random Fields," Asian J. of Control, Vol.18, No.5, pp. 1-14, 2016.
[23] S. Vasudevan, F. Ramos, E. Nettleton, and H. D. Whyte, "Gaussian Process Modeling of Large Scale Terrain," J. of Field Robotics, Vol.26, Issue 10, pp. 812-840, 2009.
[24] R. Tanabe, Y. Sasaki, and H. Takemura, "Probabilistic 3D Sound Source Mapping System Based on Monte Carlo Localization Using Microphone Array and LIDAR," J. Robot. Mechatron., Vol.29, No.1, pp. 94-104, 2017.
[25] A. Sujiwo, T. Ando, E. Takeuchi, Y. Ninomiya, and M. Edhario, "Monocular Vision-Based Localization Using ORB-SLAM with LIDAR-Aided Mapping in Real-World Robot Challenge," J. Robot. Mechatron., Vol.28, No.4, pp. 479-490, 2016.
[26] S. Shoval and A. Shapiro, "Dual-Tracked Mobile Robot for Motion in Challenging Terrains," J. of Field Robotics, JWUS107s7A/ROB-10-0128.R3 July 24, 2011.
[27] J. Borenstein, "Control and Kinematic Design for Multi-Degree-of-Freedom Mobile Robots with Compliant Linkage," IEEE Trans. on Robotics and Automation, Vol.11, No.1, pp. 21-35, 1995.
[28] T. Digaňa, "Kalman Filtering in Multi-Sensor Fusion," Master's Thesis for the degree of M.Sc., Department of Automation and Systems Technology, Helsinki University of Technology, 2004.
[29] T. G. Lee, "Centralized Kalman Filter with Adaptive Measurement Fusion: its Application to a GPS/SDINS Integration System with an Additional Sensor International," J. of Control, Automation, and Systems, Vol.1, No.4, 2003.
[30] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Department of Computer Science, University of North Carolina, 2006.
[31] P. G. Jayasekara, G. Ishigami, and T. Kubota, "Particle Filter based 3-D Position Tracking for Terrain Rovers using Laser Point Clouds," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), September 14-18, Chicago, IL, USA, 2014.

Name:
Avi Turgeman

Affiliation:
Research Assistant, Institute of Control Systems, Hamburg University of Technology

Address:
Eissendorfer Str. 40, 21073 Hamburg, Germany

Brief Biographical History:
2005-2009 B.Sc. in Electrical Engineering, Technion (Israel Institute of Technology)
2011-2015 M.Sc. in Autonomous Systems and Robotics, Technion (Israel Institute of Technology)
2016- Research Assistant, Institute of Control Systems, Hamburg University of Technology

Main Works:
• "Mission Control – Combined Solutions for Source Seeking and Level Curve Tracking in a Time-Varying Field," 2017 American Control Conf., pp. 4268-4273, 2017.

Name:
Shraga Shoval

Affiliation:
Associate Professor and Dean, Faculty of Engineering, Ariel University

Address:
Faculty of Engineering, Ariel 707000, Israel

Brief Biographical History:
1990-1994 Ph.D., University of Michigan, USA
1994-2000 Assistant Professor, Technion (Israel Institute of Technology)
2000- Associate Professor, Ariel University

Main Works:
• S. Shoval and A. Shapiro, "Dual Tracked Mobile Robot for Motion in Challenging Terrains," J. of Field Robotics, Vol.28, No.5, pp. 769-791, Sep. 2011.
• S. Shoval, M. Efatmaneshnik, and M. J. Ryan, "Assembly sequence planning for processes with heterogeneous reliabilities," Int. J. of Production Research (IJPR), Vol.55, No.10, pp. 2806-2828, 2017.
• S. Hacohen, S. Shoval, and N. Shvalb, "Dynamic Model for Pedestrian Crossing in Congested Traffic Based on Probabilistic Navigation Function," Transportation Research, Part C, Vol.86, pp. 78-96 2018 (in press).

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)
• The International Council on Systems Engineering (INCOSE)

Name:
Amir Degani

Affiliation:
Assistant Professor, Faculty of Civil and Environmental Engineering, Technion's Autonomous Systems Program, Technion (Israel Institute of Technology)

Address:
609 Rabin Building, Technion, Haifa 32000, Israel

Brief Biographical History:
2002- B.Sc. (Summa Cum Laude), Mechanical Engineering, Technion (Israel Institute of Technology)
2010- Ph.D., Robotics Institute, Carnegie Mellon University, USA
2011- Joined the Technion (Israel Institute of Technology)

Main Works:
• A. Degani, H. Choset, and M. T. Mason, "DTAR – A Dynamic Tube Ascending Robot," IEEE Trans. on Robotics, Vol.27, Issue 2, pp. 360-364, 2011.
• A. Degani, A. W. Long, S. Feng, H. B. Brown, R. D. Gregg, H. Choset, M. T. Mason, and K. M. Lynch, "Design and Open-Loop Control of the ParkourBot, a Dynamic Climbing Robot," IEEE Trans. on Robotics, Vol.30, Issue 3, pp. 705-718, 2014.
• N. Shemer and A. Degani, "A Flight-Phase Terrain Following Control Strategy for Stable and Robust Hopping of a One-Legged Robot Under Large Terrain Variations," Bioinspiration & Biomimetics, Vol.12, Issue 4, 2017.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE) Robotics and Automation Society (RAS)
• Israeli Robotics Association (IROB)
• Israeli Society of Agricultural Engineering (ISAE)