Thommen George Karimpanal

GROUND ROBOT LOCALIZATION FOR INDOOR AND OUTDOOR NAVIGATION

1.0 Introduction

One of the most challenging problems in robotics today is that of developing a robust Simultaneous Localization and Mapping (SLAM) algorithm. SLAM is an algorithm wherein mobile robots, through the sensors installed on them, form a map of a previously unknown environment and at the same time estimate its location in that environment. The SLAM problem can be thought of as a 'chicken or egg' problem, in which a good map of the environment is needed for robot localization, but at the same time, good localization is needed in order to build a reliable map. The type of SLAM algorithm used depends upon a number of factors, such as the robot hardware, sensor setup and the nature of environment (indoor or outdoor). In general, all sensors have some limitations with respect to their accuracies. This, coupled with noise from the environment result in mapping and localization errors. These errors tend to build up with time, thereby greatly limiting the accuracy of the robot's localization and mapping abilities. Due to this problem, most SLAM algorithms make use of tools such as Kalman filters and particle filters, and rely on multiple sensors to obtain more accurate localization and mapping solutions. The technique of using multiple sensors to obtain a better estimate of the measured quantity is called sensor fusion.

2.0 Methods:

In this report, several aspects relating to SLAM are investigated, the first of them being the implementation of the Co-SLAM algorithm on the Evobots platform, which is elaborated in section 2.1. The maps resulting from Co-SLAM were not very accurate, making it necessary to improve the localization through a sensor fusion algorithm that makes use of the robot's sensor readings and a dynamic model of the robot. The primary portion of the work completed so far revolves around finding a good dynamic model of the robot. The model was developed using the Bond Graph technique and validated using real robot sensor test results as described in section 2.2. The remaining part of the work is ongoing, and is related to using the robot model to perform the sensor fusion algorithm in order to improve the localization. Once the localization is improved, it can be used along with the Co-SLAM algorithm to generate an improved map of the environment.

2.1Co-SLAM

Collaborative SLAM (Co-SLAM) is an open source SLAM algorithm which is able to perform 3D SLAM through monocular cameras mounted on different agents. The algorithm is also able to handle dynamic environments, by differentiating static objects from dynamic ones in the environment. The algorithm uses inter-camera pose estimation and inter-camera mapping to do this. In order to assist the inter-camera operations, the cameras are grouped together based on the overlap of the scenes they view. The general structure of Co-SLAM is shown below:

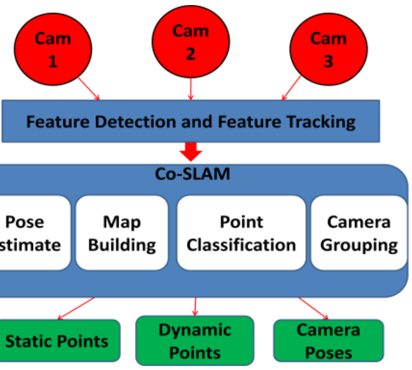


Figure 1. Schematic of the Co-SLAM algorithm

This algorithm was implemented on the Evobots using the wifi cameras mounted on them. The robots were driven around in an environment and the video data was captured and processed through the Co-SLAM algorithm.

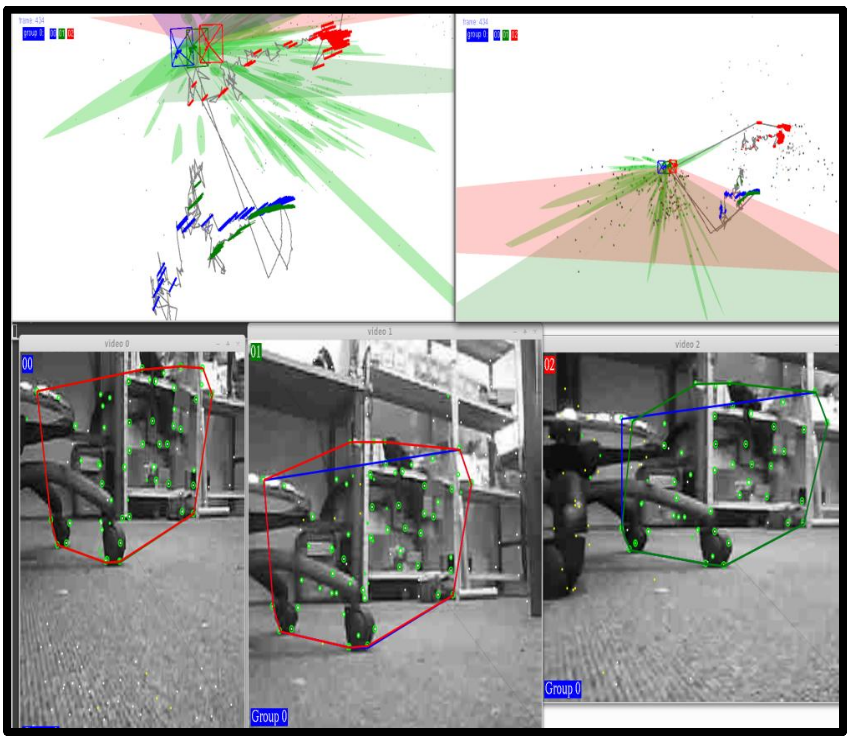


Figure 2. Co-SLAM implementation on the Evobots

It was inferred from the experiments that Co-SLAM was a promising tool for multi-agent 3D SLAM, but the accuracy of the map needed improvement. The next step was to thus improve Co-SLAM by providing it a good localization estimate. This is done by a sensor fusion process that uses the robot sensor data and a good dynamic model of the system. The next section describes the latter.

2.2 Robot Model:

The robot was modeled as a 2 point mass, with the masses concentrated at either end of the robot body as shown in the figure below.

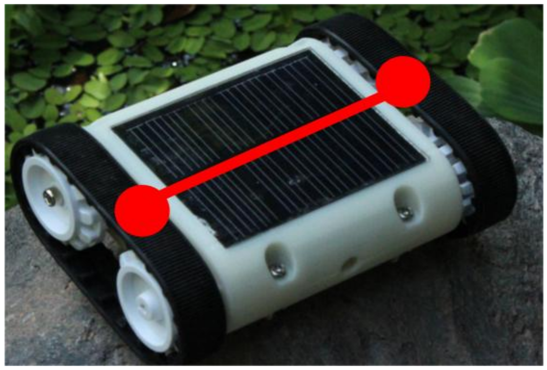


Figure 3. The Evobot platform

The bond graph model of the system is shown in figure 3. In the model, a voltage source in series with an internal resistance drives the two motors, while at the same time, powering up the other components and sensors in the robot, which are treated as parasitic components. The voltages to the motors are modulated as per their respective PWM duty cycles. The motor is modeled as a gyrator element with some inductive behavior. The motors in the robot also contain a gearbox which provides a speed reduction of 1:100, which is also modeled. The rotational motion of the motor is converted to translational motion using a transformer element whose modulus is equal to the radius of the wheel. Each motor is assumed to be attached to half the total mass of the robot. The motion of both 'halves' of the robot are coupled together with mechanical coupling (a stiffness and damper element). The equations derived from the bond graph are shown below:

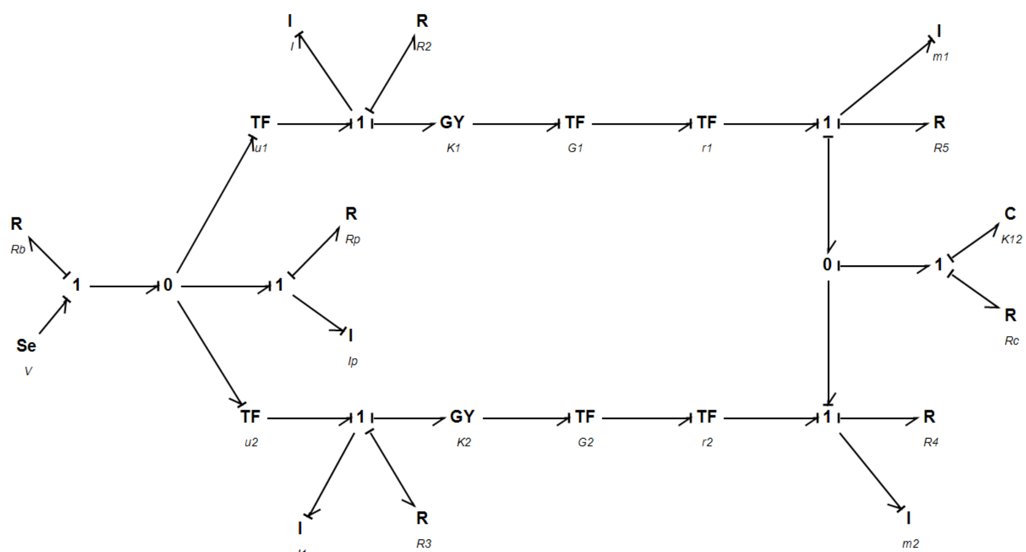


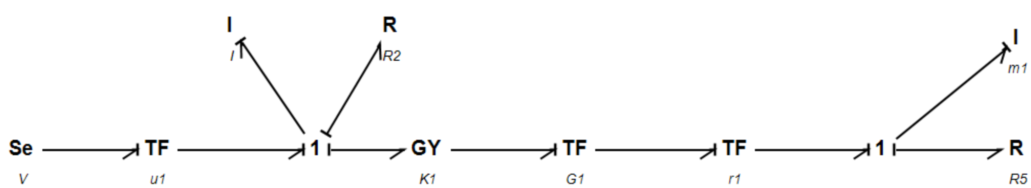
Figure 4. Bond Graph model of the robot

$$\begin{bmatrix} \dot{I_1} \\ \dot{I_2} \\ \dot{v_1} \\ \dot{v_2} \\ \dot{\Theta} \\ \dot{I_p} \end{bmatrix} = \begin{bmatrix} (-\frac{R_b}{Lu_1{}^2} - R/L) & -R_b/Lu_1u_2 & -K/GrL & 0 & 0 & -R_b/Lu_1 \\ -R_b/Lu_1u_2 & (-\frac{R_b}{Lu_2{}^2} - R/L) & 0 & -K/GrL & 0 & -R_b/Lu_1 \\ 2K/Grm & 0 & (-2R_f - 2R_c)/m & 2R_c/m & -2K_{12}/m & 0 \\ 0 & 2K/Grm & 2R_c/m & (-2R_f - 2R_c)/m & -2K_{12}/m & 0 \\ 0 & 0 & 1/l & -1/l & 0 & 0 \\ -R_b/L_pu_1 & -R_b/L_pu_2 & 0 & 0 & 0 & (-R_b/L_p - R_p/L_p) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ v_1 \\ v_2 \\ \Theta \\ I_p \end{bmatrix} + \begin{bmatrix} 1/Lu_1 \\ 1/Lu_2 \\ 0 \\ 0 \\ 0 \\ 1/L_p \end{bmatrix} [V]$$

Where :

I_1- Current from one of the sources
I_2- Current from the other source
v_1- Velocity of a point on the rim of the wheel/velocity of one side of the robot
v_2- Velocity of a point on the rim of the other wheel/velocity of the other side of the robot
Θ – Heading/ orientation
I_p- Parasitic Current (includes IR and optical flow sensors)
L_p- Parasitic Inductance
R_p- Parasitic Resistance
R_b – Internal resistance of the battery (0.5Ω)
R- Resistance of motor windings (2 Ω)
L – Inductance of each motor (0.0015H)
K – K value of the motors (0.005V/rad/s)
G – Gear Ratio of motors (1/100)
r – Radius of the wheels (0.02m)
l – equivalent length of the robot (0.15)
m – equivalent mass of the robot (0.15kg)
V- Battery Voltage (3.8V)
u_1- PWM ratio to first motor
u_2- PWM ratio to second motor
R_f- Translational Frictional Damping Constant (7Ns/m)
R_c – Damping constant of the coupling (1Ns/m)
K_{12} – Stiffness of the coupling (1Ns/m)

The model can be simplified by treating the two halves of the robot as independent dynamic systems and by separately accounting for the parasitic losses. With this approach, it would be required to use three inputs, two PWM inputs u_1 and u_2 (corresponding voltages $V_1= u_1V$ and $V_2= u_2V$) for each of the two motors, and one constant input 1(corresponding voltage 1*V=V) that acts directly on the parasitic elements. The corresponding bond graph models for each side and for the parasitic elements are shown in figures 4 and 5 respectively.



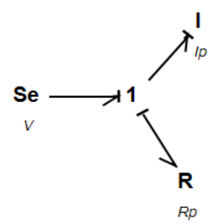


Figure 4. Alternate model of the robot

The system equations can be derived from these models and can be coupled together to obtain an alternative, simplified model as shown below:

$$\begin{bmatrix} \dot{I_1} \\ \dot{I_2} \\ \dot{v_1} \\ \dot{v_2} \\ \dot{\Theta} \\ \dot{I_p} \end{bmatrix} = \begin{bmatrix} -R/L & 0 & -K/GrL & 0 & 0 & 0 \\ 0 & -R/L & 0 & -K/GrL & 0 & 0 \\ 2K/Grm & 0 & -2R_f/m & 0 & 0 & 0 \\ 0 & 2K/Grm & 0 & -2R_f/m & 0 & 0 \\ 0 & 0 & 1/l & -1/l & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -R_p/L_p \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ v_1 \\ v_2 \\ \Theta \\ I_p \end{bmatrix} + \begin{bmatrix} V/L & 0 & 0 \\ 0 & V/L & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & V/L_p \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ 1 \end{bmatrix}$$

Either of the above mentioned models could be used for solving the optimization problem at hand. In both the models, two additional states were added, one each to determine the x and y positions of the robot. This can easily be computed as the velocities v_1 and v_2 are known.

2.2.1 Model Validation

Both the robot models were validated by performing tests on the physical robot and measuring the electrical and mechanical characteristics. The test results are summarized below in Table 1

Turning tests:

% of Max PWM voltage	Radius while turning in cm (on floor mat)	Radius while turning in cm (on table surface)	Corresponding radius on simulation in cm
25	12.5	11.5	12.5
50	20	19	22.9
75	58	55.5	53.5

Straight Line Tests:

% of max PWM voltage	Actual Velocity (cm/s)	Velocity in simulation (cm/s)
100	19.09	18.85
50	9.47	9.42
25	4.85	4.77

Electrical Tests:

Model 1

Condition:	Average Measured Current (mA)	Current in Simulation (mA)
Both motors on full speed	283	278
One Motor on full speed	240	233

Condition:	Average Measured Current (mA)	Current in Simulation (mA)
Alternate Model		
Both motors on full speed	283	288
One Motor on full speed	240	242

3.0 Sensor Fusion:

The sensor fusion process makes use of the robot model derived above, along with the sensor data obtained from the robot in order to make a good localization estimate. This estimate will later be fed into the Co-SLAM algorithm to make it more accurate and robust. The actual sensor fusion process is still in progress and will be completed soon. Once a good pose estimate of the robots is obtained, it can be merged with Co-SLAM. This will hopefully result in a more reliable and accurate map of the environment.

4.0 Demonstrating map usage: Energy-Optimal Path Planning and Navigation

Experiments were conducted to explore possible applications once a good map of the environment is obtained. Since the Evobots are a swarm of energy-starved robots which need to be used for long term missions, the motion of the robots need to be as energy-economical as possible. For this, optimization theory was made use of and an algorithm was developed to demonstrate an energy efficient path planning algorithm. The optimization problem was formulated by constructing and minimizing a cost function which comprises of the power consumed by the robot, the environment costs (obtained from the map information) and a terminal constraint dictated by the final destination of the robot. The constraints for the optimization were the inputs to the robot (duty cycle to the motors) and the dynamics of the system itself. The results of the optimal path planning algorithm are shown (in simulation and through experiments) in the figure below.

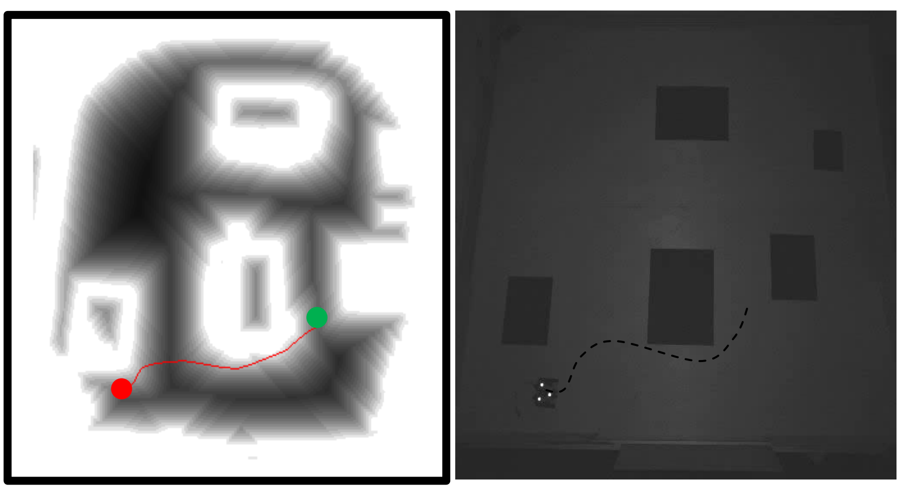


Figure 5. Energy-Optimal path planning and navigation in simulation (left) and in reality (right)

5.0 Conclusion

The project investigated the feasibility of Co-SLAM for a swarm of ground robots. Co-SLAM was found to be suitable for the application, but map of the environment needed to be improved. A dynamic model of the system was formulated using the Bond Graph approach, and the resulting model was validated with experiments. The sensor fusion process was started and is still in progress. The resulting localization will be used with the Co-SLAM algorithm to improve it. A sample application of energy efficient navigation was demonstrated to show some of the possible applications once a good map is available.

References:

[1] "CoSLAM: Collaborative Visual SLAM in Dynamic Environments", Danping Zhou and Ping Tan, IEEE trans on Pattern Analysis and Machine Intelligence, 2012

[2] “Probabilistic Robotics”, Sebastian Thrun, Wolfram Burgard, Dieter Fox, MIT press, 2005

[3] http://www.bondgraph.org/

[4] “System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems”, Dean C. Karnopp, Donald L. Margolis and Ronald C. Rosenberg, John Wiley & Sons Inc., 4th Edition, 2006

[5] “Digital Image Processing”, R.C. Gonzalez and R.E. Woods, Prentice Hall, Third Edition, 2008

[6] “Optimal and Efficient Path Planning for Partially-Known Environments”, Antony Stentz in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)*, May, 1994, pp. 3310 - 3317.

[7] “RRT∗-Smart: Rapid convergence implementation of RRT∗ towards optimal solution” Nasir J., Malik U., Ayaz Y. and Hasan O. in Mechatronics and Automation (ICMA), 2012 International Conference on