

SLAM: Simultaneous Localization and Mapping of small sized ground robotic vehicles

Sudeshna Surabhi

(EIE, Bharath University, India)

Abstract: *We require knowing the real time location of a small sized ground robotic vehicle when it performs any task indoors or outdoors. I am implementing low cost, self contained localization system for small sized vehicles. SLAM provides most accurate real time, 3D positions in both indoor as well as outdoor environments. Unlike other localization schemes it does not require external reference, expensive hardware, and strict calibration which are capable of operating in different environments. It identifies relative movement through compass sensor and MEMS sensor. It also corrects the localization drift by GPS augmentation. The complexity of the technical processes of locating and mapping under conditions of errors and noise do not allow for a coherent solution of both tasks. Simultaneous localization and mapping (SLAM) is a concept that binds these processes in a loop and therefore supports the continuity of both aspects in separated processes.*

I. Introduction

Small sized vehicles have great potential to be deployed in situations that are either uncomfortable for humans or too tiresome. Considering the difficulty in management of external facilities, it is required to have a self contained positioning system for robot which can perform indoor and outdoor works independently.

There exist various localization schemes for ground robotic vehicles. This technique uses GPS, sensors, visual processing etc. GPS operations consume more power and cannot operate in certain environments. Vision based positioning is very costly and inconvenient to maintain. The radio based positioning technology requires a set of external devices to generate or receive radio signal having known positions. Maintaining such a positioning system can be costly and difficult in terms of addition of hardware, intensive tuning and environment management. Accuracy may be affected by other signals. Another is vision technique which relies on sophisticated techniques and the localization suffers frequent failure. So vision based positioning is relatively costly and difficult to implement or maintain. SLAM identifies the real time localization through a set of self-integrated inexpensive sensors including MEMS sensor, compass sensor and infrequent GPS-augmentation. It determines local relative position with a combination of MEMS sensor and compass sensor. It is free from many other localization schemes such as vision-technique based schemes, external reference facilities.

II. The Design Of Slam

SLAM of small sized autonomous vehicle localizes a robotic vehicle with a hybrid approach consisting of infrequent absolute positioning through a GPS receiver and local relative positioning based on compass sensor and MEMS sensor. These sensors are installed on the small robotic vehicle. The compass sensor detects the rotational movement of the robot. An embedded microcontroller is inside the robot vehicle takes central control of these sensors as well as also is responsible for computing the current absolute position of it. It infrequently uses GPS to reduce the cumulative error of the local relative positioning component. The local relative component has lower error as compared to GPS in short time interval. Virtually there is no requirement of external devices and no prior information is required. All the essential devices are attached to the robotic vehicle body which is needed to localize. Except for GPS, as it is maintained by the official organizations. An unbiased map is needed for localization while an accurate pose estimate is needed to build that map. Beyond, the answering of the two characteristic questions is not as straightforward as it might sound due to inherent uncertainties in discerning the robot's relative movement from its various types of sensors. There are various techniques to compensate errors, such as recognizing features that it has come across before, and re-skewing recent parts of the map to make sure the two instances of that feature become one. "Active SLAM" studies the combined problem of SLAM with deciding where to move next in order to build the map as efficiently as possible. The need for active exploration is especially pronounced in sparse sensing regimes such as tactile SLAM. Active SLAM is generally performed by approximating the entropy of the map under hypothetical actions.

III. Introduction To The CS420000 Compass

The Digital Compass Sensor magnetically indicates the four Cardinal (N, E, S, W) directions, and the four intermediate (NE, NW, SE, SW) directions. The Compass Sensor is compatible with the C Stamp

microcomputer's supplies and signal levels. Acquiring a direction from the sensor is made easy with AWIT's supplied software command COMPASSIN. This simple one command interface is all that is required to interrogate the Compass.

3.1 Compass Sensor

The Digital Compass Sensor magnetically indicates the four Cardinal (N, E, S, W) directions, and the four intermediate (NE, NW, SE, SW) directions. The Compass Sensor is compatible with the C Stamp microcomputer's supplies and signal levels. Acquiring a direction from the sensor is made easy with A-WIT's supplied software command COMPASSIN. This simple one command interface is all that is required to interrogate the Compass. The Sensor is a combination of a sub miniature rotor jewel suspended in combination with solid state Hall-effect IC's. The Sensor was designed to indicate the direction of the horizontal flux pattern (Compass component) of the Earth's field thus becoming a compass. The Sensor is damped to give the same speed indication as a liquid filled compass, that is, 0.5 second response from 90° displacement without over-swing. It has built-in hysteresis to prevent indication "flutter" when near a switching direction. The Sensor will accept input power between 5.0 and 20.0 Volts, but should be polarity and "spike" protected from a vehicular power supply. The output is an "open collector" NPN transistor which will sink up to 25 milliamps per direction. The Sensor A - W I T T E C H N O L O G I E S I N C. 3 is low cost and reliable. The Sensors is based on Hall-effect technology and can be mounted remotely. Also, all Sensor outputs can be used in stand-alone applications. The Sensor may be operated from input voltage of 5 to 20 Volts DC with 8 to 13 Volts recommended. Input should be both "spike" and polarity protected. Power requirement is approximately 1 to 30 milliamps. Each output will sink up to 25 milliamps. The Sensor will switch so that no more than two adjacent output channels are asserted at any one time. Each output is an open collector NPN transistor pulling the output to ground, and thus it does not add to the input current requirements. The Sensor is internally designed to respond to directional change similar to a liquid filled compass. It will return to the direction indicated from a 90° displacement in approximately 0.5 to 1.0 seconds with no over-swing. The Sensor should be operated in a vertical position. The Sensor indicates the horizontal component or compass component of the Earth's magnetic field. If off the vertical position, some of the vertical component of the earth's field is introduced which may create some directional error. Generally, tilt up to 12° is acceptable with little error. The Sensor is manufactured for pins down operation, but it operates equally well pins up or down.

The Sensor weighs approximately 2.25 grams. The operating temperature is -20° C to +85° C. The sensor may be stored without damage in wider temperature limits and may be subjected to high flux levels (up to 1000 gauss) without permanent damage.

The pins are on 0.050 inch centres but may be distorted for 0.100 inch spacing without damage to the sensor or its measurements. The four VCC as well as four grounds may be common connected.

3.2 Travelled Distance

After inferring the orientation we require to know the relative movement. The compass sensor attached to the motor measures the rotating angle. Let us suppose r be the rotation angle in degrees, d be the diameter of wheel then the distance travelled by the wheel is $\frac{\pi * r * d}{360}$. Robotic vehicle may take turns by adjusting its two sides of motors at different speeds. It is a question how to calculate distance when there are different sensors, one on each side. Firstly it is observed that any small segment of movement can be taken as part of a circular movement in certain origin. A short local curve can be approximated which will reflect the turning of curve at a point. Locally approximating the curve gives negligible computing distance.

3.3 Mapping

SLAM in the mobile robotics community generally refers to the process of creating geometrically consistent maps of the environment. Topological maps are technique of environment representation which captures the connectivity (i.e., topology) of the environment rather than creating a geometrically accurate map. The resulting algorithms that create topological maps are not referred to as SLAM. SLAM is tailored to the available resources, therefore not aimed at perfection, instead at operational compliance. The published perspective is employed in unmanned aerial vehicles type, autonomous vehicles of underwater, planetary rovers, new upcoming domestic robots and even inside the human body.

IV. Introductions To PIC

The microcontroller that has been used for working with compass and MEMS sensor in this project is from PIC series. PIC microcontroller is the very first RISC based microcontroller fabricated in CMOS (complementary metal oxide semiconductor) that uses separate bus for instruction and data allowing simultaneous access of program and memory of data. The advantage in use of CMOS and RISC combination is low power consumption resulting in a very small chip size with a small pin counting. The main advantageous of

using CMOS is that it has immunity to noise than other fabrication techniques. PIC (16F877) has various microcontrollers having different types of memories. Among EEPROM, EPROM, FLASH etc., few memories of which FLASH is the most recently developed. Technology which is used in pic16F877 is flash technology, so that data can be retained even when the power is switched off. Easy Programming and Erasing are other additional features of PIC 16F877.

4.1 PIC Start Plus Programmer

The PIC start plus development system from microchip technology provides the product development engineer with a highly flexible low cost microcontroller design tool set for all microchip PIC micro type devices. The PIC start plus development system includes PIC start plus development programmer and mplug ide. The PIC start plus programme provides the product developer ability to program user software in to any of the supported microcontrollers. The PIC start plus software working under mplug provides for full interactive control over the programmer.

I/O Ports:

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. Generally when a peripheral is made enabled, that pin may not be used as a general purpose I/O pin. Additional Information on I/O ports can be found in the IC micro™ Mid-Range Reference Manual.

PORTA And The TRISA Register:

PORTA is a 6-bit wide bi-directional port. The corresponding direction of data register is TRISA. Setting a TRISA bit (=1) will make the corresponding PORTA pin an input, i.e., putting the corresponding output driver in a Hi-impedance mode. Clearing a TRISA bit (=0) will make the corresponding PORTA pin an output, i.e., putting the contents of the output latch on the selected pin. Reading the PORTA register enables reading the status of the pins whereas writing to it will write to the port latch. All the operations of write are read-modify-write operations. Hence a write to a port implies that the port pins are read; this value is changed, and then written to the port data latch. Pin RA4 is then multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI is a pin which is Schmitt Trigger input and an open drain output. All the other pins of RA port have the TTL input levels and full CMOS output drivers. Other pins of PORTA are multiplexed with analog inputs and analog VREF input. The working of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

The TRISA register controls the direction of the RA pins, even if they are used as analog form inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs.

PORTB And TRISB Register:

PORTB is an 8-bit wide bi-directional port. The corresponding direction of data register is TRISB. Setting a TRISB bit (=1) will be making the corresponding PORTB pin an input, i.e., putting the corresponding output driver in a hi-impedance mode. Clearing a TRISB bit (=0) will make the corresponding PORTB pin an output, i.e., putting the contents of the output latch on the pin which is selected. Three pins of PORTB are multiplexed with the Low Voltage Programming function; RB3/PGM, RB6/PGC and RB7/PGD. The other works of these pins are described in the Special Features Section. Each of the PORTB pins consist internally weak pull-up. A single control bit has the capacity to turn on all the pull-ups.

This is performed by clearing bit RBPU (OPTION_REG<7>). The weak pull-up is turned off automatically when the port pin is configured as an output. The pull-ups are made disabled on a Power-on Reset. Four of PORT B's pins, RB7:RB4, have an interrupt on changing the characteristic. Only configured pins as inputs can make this interrupt to occur (i.e. any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>). This interrupt can wake the device from SLEEP mode. In the interrupt service routine, the user can clear the interrupt in the following way:

- a) Any read or write of PORTB. The mismatch condition ends.
- b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Mismatch condition can be ended by reading the PORTB, and allow flag bit RBIF to be cleared. On change feature the interrupt is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt if feature is changed. Polling of PORTB is not advised during the use of the interrupt on change feature. This interrupt on different feature, along with the software configure, pull-ups on these four pins, allow the interface easily to a keypad and make it possible for wake-up on key depression.

PORTC And The TRISC Register:

PORTC is an 8-bit wide and is bidirectional port. The corresponding path of data register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORTC pin as input, i.e., making the corresponding driver of output in a hi-impedance mode. Clearing a TRISC bit (=0) will make the corresponding PORTC pin an output, i.e., contents of the output latch are placed on the pin which is selected. PORTC is multiplexed along with several peripheral functions. A PORTC pin contains Schmitt Trigger input buffers.

When the I2C module is enabled, the PORTC (3:4) pins can be configured with normal I2C levels or with SMBUS levels by using the CKE bit (SSPSTAT <6>). When enabling peripheral functions, precaution should be taken in defining TRIS bits for each pin of PORTC. Some peripherals overtake the TRIS bit to make a pin an output; where as other peripherals overtake the TRIS bit to make an input pin. As the TRIS bit overriding while the peripheral is enabled, read-modify write instructions (BSF, BCF, XORWF) with TRISC as destination must be avoided. Reference of the corresponding peripheral section for the correct TRIS bit settings should be taken by the user.

PORTD And TRISD Registers:

This section is not applicable to the devices of 28-pins. PORTD is an 8-bit port whose input buffers is Schmitt Trigger. Each pin separately can be configured as an input or output. PORTD can be arranged as 8-bit wide microprocessor Port (parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

PORTE And TRISE Register:

PORTE has three pins RE0/RD/AN5, RE1/WR/AN6 and RE2/CS/AN7, are configured individually as inputs or outputs. These pins are having input buffers as Schmitt Trigger.

The PORTE pins become control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. The user should make surity that the TRISE<2:0> bits are set (pins are configured for inputs which are digital). Assuring ADCON1 is configurable for digital input or output. The buffers for input are TTL in this mode.

PORTE pins are multiplexed with analog inputs. As it is selected as an analog input, the port pins will read as '0's. TRISE controls the direction of the RE pins, although inputs are in analog form. Making sure the user should keep the pins configured as inputs when using them as analog inputs.

V. Evaluation

5.1 Implementation And Evaluation

To implement SLAM, I used a low cost LEGO MINDSTORMS EV3 vehicle robot as mini robot and HTC Legend Smartphone. The HTC Legend phone is placed on the robot, to supply a set of sensors: MEMS sensor, a compass sensor and a GPS. The HTC is powered by six AA batteries, LEGO NXT EV3 moves on its two servo motors (one for left and one for right). The servo motors can rotate, either in same direction or in reverse direction, providing flexible movement. Its rotating speed can be changed by user programs at any time. The LEGO EV3 has a set of built in rotation sensors to measure continuous rotating distance of each motor. The program controls the motor's movement and collect data from the sensors and also GPS. I performed experiments in both environments indoor and outdoor from (1*1) m to (50*50) m areas. The LEGO EV3 robot randomly moves from its minimum speed to its maximum speed and can change speed or direction any moment. To follow curved turns and path it can operate two motors reversely. Each experiment takes approximately 20 minutes. The programmed robot random takes decision for next movement after certain time interval. The two approaches, SLAM and accelerometer based approach, were executed both simultaneously. The raw data of GPS were taken during each experiment. The experiment indicated that accelerometer based experiment cannot get satisfactory results within the localizing area of a ground robotic vehicle like LEGO EV3 robot did. The SLAM has a low cost setting, and more accurate positioning in both indoors as well as outdoors. Although SLAM shows drifting effect but it is compensated by the infrequent GPS augmentation.

5.2 Evaluation Of Slam With GPS-Augmentation

I performed few experiments in outdoor environment in area about (50*50) m. To obtain the reality, the GPS is turned on and computes position once at least every 3 seconds. The GPS data can be viewed locally as Cartesian coordinates. The SLAM produced coordinates are compared against GPS trace. The analysis shows that the SLAM's local relative position system produces an inaccuracy about 18m range where as in one GPS augmentation the inaccuracy is up to 8m range. The one time GPS adjustment from GPS data corrects the drift at large amount.

VI. Conclusions

I propose SLAM, a low cost, self contained, more accuracy localization system of small sized ground robotic vehicles along with GPS receiver and local positioning system based on MEMS sensor and compass sensor. SLAM fuses information from sensors to know the movement of the robot in short time interval which is further corrected by infrequent GPS augmentation. The hardware devices used by SLAM are of very low cost. SLAM does not require any external management facility and also does not require prior information. SLAM provides satisfactory performance while working indoor as well as outdoor environments. The experiments of various scales with SLAM gave accuracy. Along with GPS augmentation SLAM succeeds in maintaining the cumulative error greatly.

Reference

- [1] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
- [2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. *Fastslam: A factored solution to the simultaneous localization and mapping problem*. In 2002 IEEE ICRA, Workshop W4 Notes, 2002.
- [3] S. Thrun, D. Fox., and W. Burgard. *A probabilistic approach to concurrent mapping and localization for mobile robots*. *Machine Learning*, 31:29-53, 1998.
- [4] Durrant-Whyte, H. Bailey, T. (2006). "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms" *Robotics and Automation Magazine*.
- [5] Karlsson, N.; Di Bernardo, E.; Ostrowski, J.; Goncalves, L.; Pirjanian, P.; Munich, M. (2005). "The vSLAM Algorithm for Robust Localization and Mapping". *Int. Conf. on Robotics and Automation (ICRA)*.
- [6] Leonard, J.J.; Durrant-whyte, H.F. (1991). "Simultaneous map building and localization for an autonomous mobile robot". *Intelligent Robots and Systems' 91. Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on: 1442-1447*.