Generalized Attitude Heading Reference System (AHRS) - an open hardware and software project

markgross@thegnar.org

2/2/2008

Dead Reckoning (DR) can be hard if you don't know heading reliably.

- Most Robo-Magellan's use encoders and compasses to implement DR.
  - My robot did.
  - My robot drifted into the curb after 50'
  - My robot was off by 10deg when executing an "out and back" test on level ground.
- Odometry can't be used reliably to compute heading.
- Simple compasses have issues with tilt, and power lines.

The Generalized AHRS will provide robust, accurate heading, that is stable WRT tilt, and ambient magnetic noise.

- One of the goals of the effort is to provide a good tilt invariant compass for robotic applications.
- We are shooting for a modest BOM of < 200$
- Generalized AHRS is also a *lot* more than just a tilt proof compass

Advanced navigation robotics takes more than good DR

- Data fusion and the prediction of location with error estimates are what the big boys are doing.
- The implementations of good AHR-Systems use techniques to compute attitude, heading, and inertial state.
- This is cool stuff.

The output of this effort will:

- Provide a design and reference implementation of a stand alone tilt invariant compass that is stable, reliable and accurate for < $200 BOM
  - Competitive with the $1500 Microstrain.
- Provide a hardware and software platform for advanced navigation methods beyond dead reckoning.

Talk Outline

- High level project goals
- High level design goals (requirements)
- PSAS relationship
- OSS licenses issues
- Technical aspects this project
- Status
- Conclusion

High level project goals

Cheapest hardware and free software to enable 1% accurate outdoor navigation.

Project goals

- Make modern navigational technologies accessible to amateur robotics
- Solve the heading problem affordably for Robo-Magellan builders.
- Enable the development of new algorithms and methods.

Project goals

- Create an Open Source / Open Hardware project that will have a lifetime longer than the attention span of its principles.
- Enable 3rd parties to produce hardware for sale (PARTS?, Sparkfun?, bystander?)
- Enable 3rd parties to evolve the design.

High level design goals (requirements)

design goals

- Support robots using processors with USB host controllers with full speed USB communications
- Support robots without USB host using serial communications (I2C, SPI or RS232)
- Support basic heading reporting
- Support full sensor data reporting for advanced algorithmic techniques.

design goals

- 100 to 500Hz data update rates
- calibration design and implementation.
- 3-axis MEMS accelerometers
- 3-axis MEMS rotation rate
- 3-axis magnetometer
- MCU can interface with robotics inputs and sensors (TBD)
  - integrate GPS, encoder and sonar data.

design goals

- Sensor / interface MCU to support serial and USB interface to user.
- MCU can do data gathering and communication to the host with limited local processing.
- MCU can do data some fusion processing locally
  - Basic tilt proof compass implementation minimum.
  - This is so that users could connect the device as a serial (slave) to their robot controller and us it as a compass.

design goals

- Host software for implementing data fusion available under OSS licenses.
- Calibration automation design, platform, and related software, available under OSS compatible licensing.
- Schematics and Gerber files under OSS compatible licensing.
- Public documentation.

PSAS relationship

PSAS

- http://psas.pdx.edu/
- Solving very similar problems
- Using very similar technologies and sensors
- Much higher data rate than we need
- Different orders of magnitude accelerations.
- Working to solve the controlled orbit insertion of a micro-satellite.

PSAS

- We are participating with the PSAS IMU and avionics development.
- Wherever possible we'll cross pollinate.
- Hope to reuse MCU and tool chain knowledge.
- Hope to reuse calibration and MEMS experiences
- Expect to share data fusion and algorithm design and learning's.

PSAS

- You should check out the PSAS activities.

OSS licenses issues

Viral or not?

- Its not an easy choice.
  - If too viral then the work may not be accessible to small start-ups building on the IP they add to parts of the design, or it could prevent integration of this work into a larger system that could be contaminated by the viral nature of the licenses.
  - If not viral building a grass roots community to participate can be hard, and the life cycle of the project may become limited to the attention span of the original principles to the project.

licensing

- Leaning toward viral OSS licensing for applications and non-viral for libraries.
- Where customary licensing of the implementation environment exist, we should use those.
  - i.e. python libraries should be released under the python license.
- MCU firmware may need to be non-viral. (TBD)
- Input and advice welcome!
  - One recomendation is to use a “non-viral copyleft with explicit patent grant” styled license such as EPL or CDDL.

Technical aspects this project

Finally!

Drilling down into the good stuff

- Tool chains MCU's and communications
- Digital design
- Analog design
- Calibration
- Algorithmic design

Tool chains and MCU's

- Tool chains and communications stacks can be a pain in the butt.
  - Choosing a MCU to use in the interfacing of devices to USB and serial can become a project of its own.
  - Risking feature creep from desire to also make this MCU capable of running the robot.
- Today we have a working UBW (pic18F2455) with the SDCC.
- The LPC 2148 is used by PSAS and has attractive 32 bit processing with head room for algorithm computations.
  - The PSAS folks have the tool chain and JTAG tools working.
  - Sample USB implementation needs to be verified.
  - I like the idea of using olemex header board for lpc2148 (http://www.sparkfun.com/commerce/product_info.php?products_id=676)

Tool chains MCU's and USB communications

- Today we have a working UBW with crude USB stack.
- The LPC 2148 has an available OSS bulk USB stack we can try.
- FTDI chip serial to USB used with an AVR or some other part is an option.
  - Target data rates between the host and device is less than ((3 + 6 + 9) * 16 bits) at 100 to 500 updates per second or 28800bps to 144000bps
  - Higher rates will be needed if additional sensor input is incorporated into data stream.

Tool chains MCU's and serial communications

- Serial slave mode needed for integration into a MCU based robot design.
- Serial transport needs to support reliable communications at 1.5 meters.
- Anticipated data rate and processing capabilities of serial master call for filtered data to be sent
  - Such as the heading output from a Kalman filter running on the MCU.
- MCU firmware needs to support this use. Hopefully MCU will be able to handle running the serial output and the raw USB output concurrently.

Digital design

- Digital design issues around interfacing different components together. Some parts are GPIO, I2C, others are SPI.
- Voltage issues around having a 3.3 or 5V part talk to a 5 or 3.3V part reliably and at the maximum bandwidth.
  - tuning pull ups for buses
  - tuning for line noise.
- Clean digital signals in an electrically noisy environment (motor drivers near by)
- Other stuff I don't know a lot about…

Analog design

- Analog design issues around interfacing A2D's and the base sensors is also an area for design review an challenges.
- To do well the sensors need to be sampled with more than 10 bits of precision (16 bits is the target). There are issues to doing a good job at connecting the sensors to the A2D's and then the A2D's to the MCU.
- We want between 150 and 500 good samples per second from the sensors to send to the host.

Analog design

- Some of the MEMS devices sample at different rates than others and we'll have to accommodate that
- Scaling signals with op-amps may be needed and tuned for maximum dynamic range in data.
- TBD challenges in doing a good job at this.

Calibration

- Device calibration turns out the be a step that separates good IMU's from shake sensors.
  - We will need to calibrate and tune the hardware to get good results.
- We need to create a procedure for calibration.
- We need to design hardware for executing the calibration procedure.

Calibration

- Calibration tends to be unique per device.
  - Will likely require some flash or storage for calibration values.
- We need to figure out where and how to associate the calibration information onto the device.
  - HW serial numbers?
  - flash / EEPROMs?

Algorithmic design

- Filter design / data fusion is needed before the calibrated sensor information is more than just numbers with a crude correlation or physical environment.
- The host software that consumes the 28.8Kbs to 144Kbs data stream from the device will convert the numbers into initial state and heading values.
- This information along with additional sensor data can be used to compute configuration and dynamical state.

Algorithmic design

- A number of existing Kalman filter implementations exist today that need to be experimented with
- Bayesian filtering (generalizations to Kalman filtering) will be experimented with.
- Other related methods to be explored such as, http://wiki.cs.pdx.edu/forge/bmpf.html.
- I'm sure others will come out of the wood work as this effort evolves.

Status

- We are just getting started and in early design and requirements definition phases of the work contributors, participants and ideas at all levels welcome

- Current principle contributors (alphabetically):
  - Jim Remington <jremington@uoxray.uoregon.edu>,
  - Mark Gross <markgross@thegnar.org>
  - Robert Scheer <rfscheer@speakeasy.net>,
  - Timothy Ressel <madhun2001@yahoo.com>,
  - Zac Wheeler <zac@poor-robot.com>

Status

Current web site:

- http://www.thegnar.org/ahrs/AHRS.html
- mailing list : Pending licensing choice.
  - For now contact markgross@thegnar.org and we'll start CC'ing new folks.
- IRC : TBD

Conclusion

- We hope this effort will develop a life of its own and produce useful hardware and software technologies for amateur robotics as well as promote research into autonomous navigation.
- We have a small team today, we are just getting stared, we are not terribly organized yet.
- Now is a good time to join the fun.

The end.