Robust Sensor Fusion for Robot Attitude Estimation

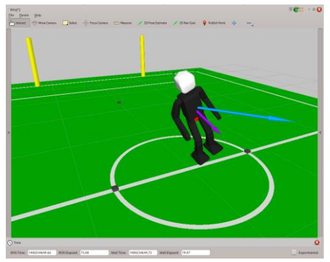
Problem Definition

Knowledge of how a body is oriented is frequently invaluable information.

We propose an **attitude estimator** that fuses 3-axis gyro, accelerometer and magnetometer data to obtain an estimate of the **orientation of a body**.

The ability to deal with **noisy sensors** and **slow processors** is key for use in low cost systems. The filter should also be **uniformly robust and stable**.

It is assumed that measurements as detailed on the right are available (models bias and zero-mean noise).



$$
\begin{aligned}
{}^{B}\boldsymbol{\Omega}_y &= {}^{B}\boldsymbol{\Omega} + \mathbf{b}_\Omega + \mathbf{v}_\Omega \\
{}^{B}\tilde{\mathbf{a}} &= {}^{B}_{G}R\,{}^{G}\mathbf{g} + \mathbf{b}_a + \mathbf{v}_a \\
{}^{B}\tilde{\mathbf{m}} &= {}^{B}_{G}R\,{}^{G}\mathbf{m}_e + \mathbf{b}_m + \mathbf{v}_m \\
{}^{B}\mathbf{z}_G &= -\frac{{}^{B}\tilde{\mathbf{a}} - \hat{\mathbf{b}}_a}{\|{}^{B}\tilde{\mathbf{a}} - \hat{\mathbf{b}}_a\|} \in S^2
\end{aligned}
$$

Complementary Filtering

1D Linear Complementary Filter

In the case of rotation within a plane, the **1D filtering** approach is

$$\dot{\hat{\theta}} = (\omega_y - \hat{b}_\omega) + k_p(\theta_y - \hat{\theta}), \quad \dot{\hat{b}}_\omega = -k_i(\theta_y - \hat{\theta}).$$

This complementary filter is unsuitable for general rotations however.

3D Nonlinear Passive Complementary Filter

A generalised 3D rotation error feedback term Ω_e is constructed and used to calculate a quaternion estimate update angular velocity Ω.

$$\boldsymbol{\Omega}_e = 2\tilde{q}_0\tilde{\mathbf{q}}, \quad \boldsymbol{\Omega} = \boldsymbol{\Omega}_y - \hat{\mathbf{b}}_\Omega + k_p\boldsymbol{\Omega}_e$$

The **filter update**, including continued gyro bias estimation, is then

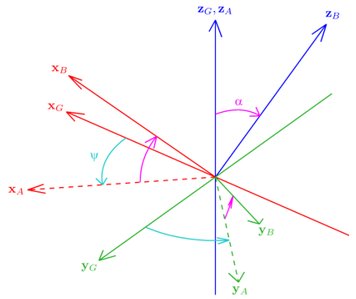
$$\dot{\hat{q}} = \tfrac{1}{2}\hat{q}\Omega, \quad \dot{\hat{\mathbf{b}}}_\Omega = -k_i\boldsymbol{\Omega}_e.$$

The **passive complementary filter** is discussed by Mahony et al. in [1].

[1] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," IEEE Trans. Automat. Contr., vol. 53, no. 5, pp. 1203–1218, 2008.

Orientation Resolution Methods

The core issue is how to calculate q_y, the quaternion orientation consistent with the sensor measurements. Three **orientation resolution methods** have been developed to address this need. One method uses the magnetometer reading ${}^{B}m$, while the other two are for cases where this is not possible. All methods strictly respect ${}^{B}z_G$ and need only calculate ${}^{B}x_G$ and ${}^{B}y_G$ to define R_y.



Magnetometer Method

The axes are calculated that **minimise the angular difference** between the global magnetic vector and the measured one.

$$
\begin{aligned}
{}^{B}\hat{\mathbf{m}} &= {}^{B}\mathbf{m} - ({}^{B}\mathbf{m}\cdot{}^{B}\mathbf{z}_G){}^{B}\mathbf{z}_G \\
{}^{B}\hat{\mathbf{u}} &= {}^{B}\hat{\mathbf{m}} \times {}^{B}\mathbf{z}_G \\
{}^{B}\tilde{\mathbf{x}}_G &= m_{ex}\,{}^{B}\hat{\mathbf{m}} + m_{ey}\,{}^{B}\hat{\mathbf{u}} \\
{}^{B}\tilde{\mathbf{y}}_G &= m_{ey}\,{}^{B}\hat{\mathbf{m}} - m_{ex}\,{}^{B}\hat{\mathbf{u}}
\end{aligned}
$$

ZYX Yaw Resolution Method

The axes are calculated that lead to **zero relative ZYX yaw** from G to H, the current estimated global frame.

$$
\begin{aligned}
{}^{B}\mathbf{x}_H &= (\tfrac{1}{2} - \hat{y}^2 - \hat{z}^2,\ \hat{x}\hat{y} - \hat{w}\hat{z},\ \hat{x}\hat{z} + \hat{w}\hat{y}) \\
{}^{B}\tilde{\mathbf{x}}_G &= {}^{B}\mathbf{x}_H - ({}^{B}\mathbf{x}_H\cdot{}^{B}\mathbf{z}_G){}^{B}\mathbf{z}_G \\
{}^{B}\tilde{\mathbf{y}}_G &= {}^{B}\mathbf{z}_G \times {}^{B}\tilde{\mathbf{x}}_G
\end{aligned}
$$

Fused Yaw Resolution Method

The axes are calculated that lead to **zero relative fused yaw** from G to H, for H the current estimated global frame.

$$
\begin{aligned}
{}^{H}\mathbf{z}_G &= \hat{q}\,{}^{B}\mathbf{z}_G\,\hat{q}^* = (z_{Gx}, z_{Gy}, z_{Gz}) \\
{}^{G}_{B}\bar{q}_y &= \begin{bmatrix} 1+z_{Gz} & -z_{Gy} & z_{Gx} & 0 \\ z_{Gy} & 1+z_{Gz} & 0 & -z_{Gx} \\ -z_{Gx} & 0 & 1+z_{Gz} & -z_{Gy} \\ 0 & z_{Gx} & z_{Gy} & 1+z_{Gz} \end{bmatrix} {}^{G}_{E}\bar{q}
\end{aligned}
$$

Extensions to the Estimator

The estimator has been extended with the following features:

- **Quick learning** for faster settling from large initial estimation errors
- Equations for estimation with only **2D acceleration data**
- Equations for estimation with **reduced order magnetometer data**
- **Fused yaw removal** for estimation without magnetometer data

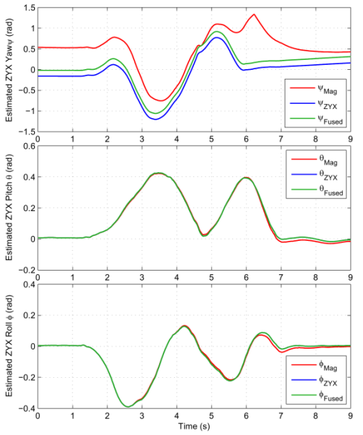
Experimental Results

The attitude estimator has been implemented in **simulation** as well as on **real robots** (e.g. NimbRo-OP).

The estimator update code takes 110 - 145ns to execute on a single 2.40GHz CPU core (≈ 6.9 - 9.1MHz).

Data from a **real robot** is shown to the right, recorded during manual disturbance of the robot's attitude.

More results demonstrating quick learning are also available.



Open-source C++ Library Implementation

The complete attitude estimator, including all extensions, has been implemented as a **C++ library** and is **freely available online**.

Features:

- Small, performant and code-efficient cross-platform library
- Ready to use, with no external dependencies
- Documented and explained using Doxygen
- Implements the full attitude estimation algorithm

URL: https://github.com/AIS-Bonn/attitude_estimator

Notation and Identities

Details: Rotation matrices, special orthogonal group, quaternions, calculation of quaternion vector rotation, parallel, antiparallel, robust rotation matrix to quaternion conversions, and vice versa.

Philipp Allgeuer and Sven Behnke

Robust Sensor Fusion for Robot Attitude Estimation

Problem Definition

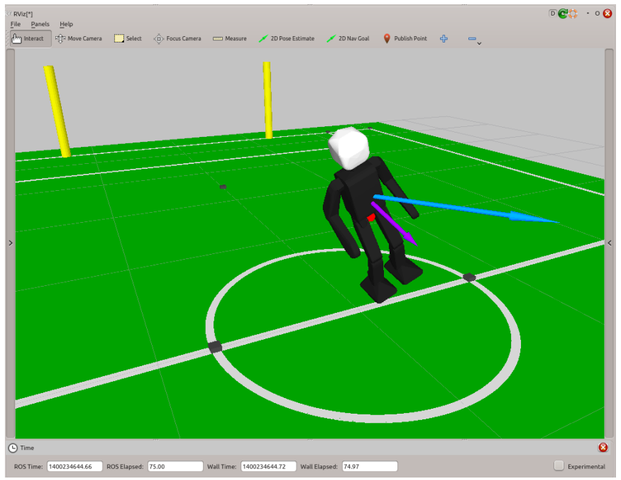
Problem Overview

Attitude estimation is the task of constructing an estimate of the full 3D orientation of a body relative to some global fixed frame, based on a finite history of sensor measurements.

Knowledge of how a body is oriented relative to the world is frequently invaluable information for the purposes of localisation and/or control.

We propose an attitude estimator that fuses 3-axis **gyroscope**, **accelerometer** and **magnetometer** data into a quaternion orientation estimate.

The estimator should be able to function well with **low cost hardware**. This means that the estimator must be **computationally efficient** (slow processor), and be able to cope with **high noise inputs** without excessively sacrificing estimator response. The estimator should also be uniformly robust and stable.



Gyroscope Measurements

This sensor is assumed to measure the **angular velocity** of the body, in body-fixed coordinates. The measurement is assumed to be affected by a gyroscope bias b_Ω, and zero-mean sensor noise v_Ω. That is,

$$^B\boldsymbol{\Omega}_y = {}^B\boldsymbol{\Omega} + \mathbf{b}_\Omega + \mathbf{v}_\Omega \in \mathbb{R}^3.$$

Accelerometer Measurements

This sensor is assumed to provide a measure of the **proper acceleration** of the body. This corresponds to the combination of inertial acceleration and gravity, where the latter component is **assumed to dominate**. The measurement is assumed to be affected by a time-invariant bias b_a, and zero-mean sensor noise v_a. An accelerometer calibration should be used to estimate ${}^Bz_G$. That is,

$$^B\tilde{\mathbf{a}} = {}^B_G R\, {}^G\mathbf{g} + \mathbf{b}_a + \mathbf{v}_a \in \mathbb{R}^3, \quad {}^B\mathbf{z}_G = -\frac{{}^B\tilde{\mathbf{a}} - \hat{\mathbf{b}}_a}{\|{}^B\tilde{\mathbf{a}} - \hat{\mathbf{b}}_a\|} \in S^2.$$

Magnetometer Measurements

This sensor is assumed to provide a measure of the strength and direction of the **Earth's magnetic field**, in body-fixed coordinates. The measurement is assumed to be affected by a time-invariant bias b_m, and zero-mean sensor noise v_m. A hard-iron magnetometer calibration is assumed. That is,

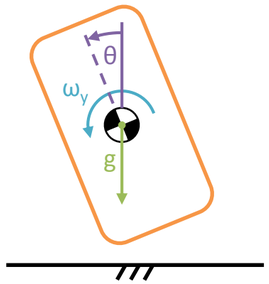
$$^B\tilde{\mathbf{m}} = {}^B_G R\, {}^G\mathbf{m}_e + \mathbf{b}_m + \mathbf{v}_m \in \mathbb{R}^3, \quad {}^B\mathbf{m} = \frac{{}^B\tilde{\mathbf{m}} - \hat{\mathbf{b}}_m}{\|{}^B\tilde{\mathbf{m}} - \hat{\mathbf{b}}_m\|} \in S^2.$$

Robust Sensor Fusion for Robot Attitude Estimation

Complementary Filtering

1D Linear Complementary Filter

For simple 1D rotations within a plane, a **complementary filter** can be constructed that fuses the **low pass** attitude information from the accelerometer with the **high pass** attitude rate data from the gyroscope. This is achieved by integrating the angular velocity data and feeding the angle information back via a **proportional-integral** (PI) controller. The filter equations are given by



$$\begin{aligned} \theta_e &= \theta_y - \hat{\theta} \\ \dot{\hat{\theta}} &= \omega_y - \hat{b}_\omega + k_p \theta_e \\ \dot{\hat{b}}_\omega &= -k_i \theta_e. \end{aligned}$$

$$\begin{aligned} \theta_e &\equiv \text{Estimated angle error} \\ \hat{\theta} &\equiv \text{Rotation angle estimate} \\ \theta_y &\equiv \text{Angle measured from accelerometer} \\ \omega_y &\equiv \text{Angular velocity from gyroscope} \\ \hat{b}_\omega &\equiv \text{Estimated gyroscope bias} \\ k_p, k_i &\equiv \text{PI compensator gains} \end{aligned}$$

The 1D filter may naïvely be implemented independently in two of three axes in the general 3D case, but this generally gives **unsatisfactory results** and fails to capture much of the dynamics of 3D rotations.

Extension of Complementary Filtering to 3D

It is desired to formulate a complementary filter on the full 3D rotation space that retains the positive frequency characteristics of the 1D filter. Mahony et al. published the **passive complementary filter** in [1].

[1] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," IEEE Trans. Automat. Contr., vol. 53, no. 5, pp. 1203–1218, 2008.

3D Nonlinear Passive Complementary Filter

Let {B} be the **body-fixed frame**, {G} be the **global frame**, and {E} be the estimated body-fixed frame relative to the global frame {G}, based on the current quaternion orientation estimate.

Based on the accelerometer and magnetometer measurements, we first construct a 'measured' orientation q_y. This is the purpose of the orientation resolution methods. The **estimated rotation error** can then be calculated, and the axis of this is used for the corrective error feedback term Ω_e. That is,

$$ {}^E_B\tilde{q} = {}^G_E\hat{q}^* \, {}^G_B q_y \equiv (\tilde{q}_0, \tilde{\mathbf{q}}), \quad \boldsymbol{\Omega}_e = 2\tilde{q}_0\tilde{\mathbf{q}}, \quad \boldsymbol{\Omega} = \boldsymbol{\Omega}_y - \hat{\mathbf{b}}_\Omega + k_p\boldsymbol{\Omega}_e. $$

The calculated **update angular velocity** Ω is applied to the current quaternion orientation estimate, and the **gyroscope bias estimate** is updated based on the corrective error feedback term Ω_e. That is,

$$ \dot{\hat{q}} = \tfrac{1}{2}\hat{q}(0, \boldsymbol{\Omega}), \quad \dot{\hat{\mathbf{b}}}_\Omega = -k_i\boldsymbol{\Omega}_e. $$

The similarity of the passive complementary filter to the 1D filter presented on the left should become apparent when contrasting the filter equations.

$$\begin{aligned} \hat{q} &\equiv \text{Current orientation estimate} \\ q_y &\equiv \text{Measured orientation} \\ \tilde{q} &\equiv \text{Estimated rotation error} \\ k_p, k_i &\equiv \text{PI compensator gains} \end{aligned}$$

$$\begin{aligned} \boldsymbol{\Omega}_e &\equiv \text{Corrective error feedback term} \\ \boldsymbol{\Omega}_y &\equiv \text{Measured angular velocity} \\ \boldsymbol{\Omega} &\equiv \text{Update angular velocity} \\ \hat{\mathbf{b}}_\Omega &\equiv \text{Gyroscope bias estimate} \end{aligned}$$

Robust Sensor Fusion for Robot Attitude Estimation

Orientation Resolution Methods

Overview

The complementary filter requires in each cycle the calculation of a **measured quaternion orientation** q_y best fitting the sensor measurements ${}^{B}z_G$ and ${}^{B}m$. In general these two measurements suffice to construct a unique q_y, but if not, the **current orientation estimate** is also taken as an input.

In the following two methods it suffices to calculate the **directions** of ${}^{B}x_G$ and ${}^{B}y_G$, after which q_y can be calculated by converting the rotation matrix R_y into a quaternion as shown here, where R_y is constructed as

$$ {}^{B}\mathbf{x}_G = \frac{{}^{B}\tilde{\mathbf{x}}_G}{\|{}^{B}\tilde{\mathbf{x}}_G\|}, \quad {}^{B}\mathbf{y}_G = \frac{{}^{B}\tilde{\mathbf{y}}_G}{\|{}^{B}\tilde{\mathbf{y}}_G\|}, \quad {}^{G}_{B}R_y = \begin{bmatrix} {}^{B}\mathbf{x}_G & {}^{B}\mathbf{y}_G & {}^{B}\mathbf{z}_G \end{bmatrix}^T. $$

Magnetometer Method

This method utilises the magnetometer measurement and seeks to **minimise the angular difference** between the expected global magnetic field vector ${}^{G}m_e$ and ${}^{B}m$ by modifying q_y (i.e. modifying frame {B}). This condition is seen to be satisfied when the projections of the two vectors onto the x_Gy_G plane are **parallel**. Denoting ${}^{G}m_e = (m_{ex}, m_{ey}, m_{ez})$, this leads to

$$ \begin{aligned} {}^{B}\hat{\mathbf{m}} &= {}^{B}\mathbf{m} - ({}^{B}\mathbf{m} \cdot {}^{B}\mathbf{z}_G){}^{B}\mathbf{z}_G \\ {}^{B}\hat{\mathbf{u}} &= {}^{B}\hat{\mathbf{m}} \times {}^{B}\mathbf{z}_G \\ {}^{B}\tilde{\mathbf{x}}_G &= m_{ex}{}^{B}\hat{\mathbf{m}} + m_{ey}{}^{B}\hat{\mathbf{u}} \\ {}^{B}\tilde{\mathbf{y}}_G &= m_{ey}{}^{B}\hat{\mathbf{m}} - m_{ex}{}^{B}\hat{\mathbf{u}}. \end{aligned} $$

If this fails, ${}^{B}m$ is discarded and one of the **yaw resolution methods** is used.

ZYX Yaw Resolution Method

Let {H} be the current **estimated global coordinate frame**, relative to {B}. This method seeks to find q_y such that the **ZYX yaw** of {H} with respect to {G} is zero. This ensures that the filter update only **minimally affects the yaw**. Thus,

$$ \begin{aligned} {}^{B}\mathbf{x}_H &= (\tfrac{1}{2} - \hat{y}^2 - \hat{z}^2,\ \hat{x}\hat{y} - \hat{w}\hat{z},\ \hat{x}\hat{z} + \hat{w}\hat{y}) \\ {}^{B}\tilde{\mathbf{x}}_G &= {}^{B}\mathbf{x}_H - ({}^{B}\mathbf{x}_H \cdot {}^{B}\mathbf{z}_G){}^{B}\mathbf{z}_G \\ {}^{B}\tilde{\mathbf{y}}_G &= {}^{B}\mathbf{z}_G \times {}^{B}\tilde{\mathbf{x}}_G. \end{aligned} $$

In the rare case where this method fails, the **ZXY yaw** is zeroed instead, using

$$ \begin{aligned} {}^{B}\mathbf{y}_H &= (\hat{x}\hat{y} + \hat{w}\hat{z},\ \tfrac{1}{2} - \hat{x}^2 - \hat{z}^2,\ \hat{y}\hat{z} - \hat{w}\hat{x}) \\ {}^{B}\tilde{\mathbf{y}}_G &= {}^{B}\mathbf{y}_H - ({}^{B}\mathbf{y}_H \cdot {}^{B}\mathbf{z}_G){}^{B}\mathbf{z}_G \\ {}^{B}\tilde{\mathbf{x}}_G &= {}^{B}\tilde{\mathbf{y}}_G \times {}^{B}\mathbf{z}_G. \end{aligned} $$

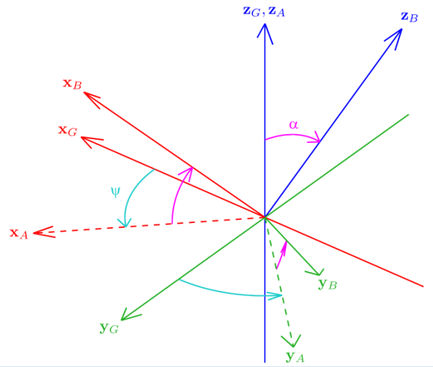
Fused Yaw Resolution Method

This method seeks to find q_y such that the fused yaw of {H}, defined as before, with respect to {G} is zero. This has a convenient quaternion solution, namely

$$ {}^{H}\mathbf{z}_G = \hat{q}\,{}^{B}\mathbf{z}_G\,\hat{q}^* = (z_{Gx}, z_{Gy}, z_{Gz}) $$

$$ {}^{G}_{B}\bar{q}_y = \begin{bmatrix} 1+z_{Gz} & -z_{Gy} & z_{Gx} & 0 \\ z_{Gy} & 1+z_{Gz} & 0 & -z_{Gx} \\ -z_{Gx} & 0 & 1+z_{Gz} & -z_{Gy} \\ 0 & z_{Gx} & z_{Gy} & 1+z_{Gz} \end{bmatrix} {}^{G}_{E}\hat{q}. $$

This method fails only in boundary cases of the **error quaternion**, where the filter is insensitive.



Robust Sensor Fusion for Robot Attitude Estimation

Extensions to the Estimator

Quick Learning

It is desired for the attitude estimate to **settle quickly** from large estimation errors, yet simultaneously provide adequate general **noise rejection**.

Quick learning allows for two sets of PI gains to be tuned—one set that provides suitably fast transient response (*quick*), and one set that provides good tracking and noise rejection (*nom*).

A parameter λ on the unit interval is used to fade linearly between the two sets of gains over a given **quick learning time**, ending in the nominal tracking gains. The gain fading scheme is given by

$$(k_p, k_i) = \lambda(k_p^{nom}, k_i^{nom}) + (1-\lambda)(k_p^{quick}, k_i^{quick}).$$

The effect of quick learning can be seen in the Experimental Results section.

Estimation with Two-Axis Acceleration Data

If only **two-axis X-Y accelerometer data** is available, then the missing part of the acceleration vector can be synthesised by assuming that the norm of the measured acceleration should be the **magnitude of gravity**, g.

In mathematical form this is

$$a_z = -\sqrt{\max\{g^2 - a_x^2 - a_y^2, 0\}}\,.$$

This equation only allows for estimates in the **positive z-hemisphere** as the sign of the missing a_z component has to be assumed. For many applications, such as bipedal walking, this can be sufficient.

Estimation with Reduced Magnetometer Data

If only **two-axis X-Y magnetometer data** is available, then this can still be used for ${}^B\mathbf{m}$ if the third unknown component is left to zero. This generally still produces satisfactory results due to the **projection operation** at the beginning of the magnetometer resolution method.

If magnetometer data is only available in terms of a **relative heading angle** ψ, then the required three-axis data can be constructed using

$${}^B\mathbf{m} = (\cos\psi,\ \sin\psi,\ 0).$$

Estimation without Magnetometer Data

If **no magnetometer data** is available, then the attitude estimation is entirely reliant on the selected yaw-based orientation resolution method. This still leads to quality estimation results in the pitch and roll dimensions by setting ${}^B\mathbf{m}$ to zero and removing the **fused yaw** of the output quaternion using

$$\tilde{\hat{q}}_s = (\hat{w}, 0, 0, -\hat{z})\,\hat{q}, \quad \hat{q}_s = \frac{\tilde{\hat{q}}_s}{\|\tilde{\hat{q}}_s\|}$$

It is **not recommended** to remove the Euler ZYX yaw instead, as this leads to unexpected behaviour near the not uncommon case of ±90° pitch rotations.

Note that the open-loop yaw produced by the estimator **remains stable** with each update of q_y due to the yaw-zeroing approach used. Small drift velocities in the yaw can still unavoidably result however, due to the gyroscope biases.

Robust Sensor Fusion for Robot Attitude Estimation

Experimental Results

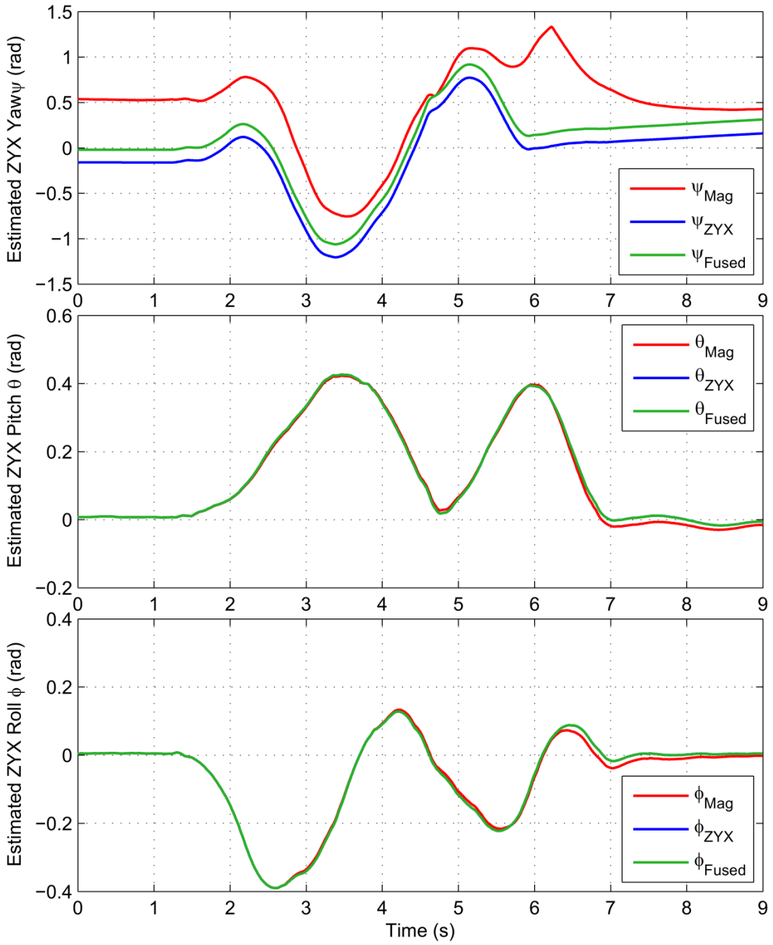


Figure 1: Estimation results on the NimbRo-OP robot using the magnetometer, ZYX yaw and fused raw orientation resolution methods. The attitude of the robot was manually disturbed for a number of seconds.

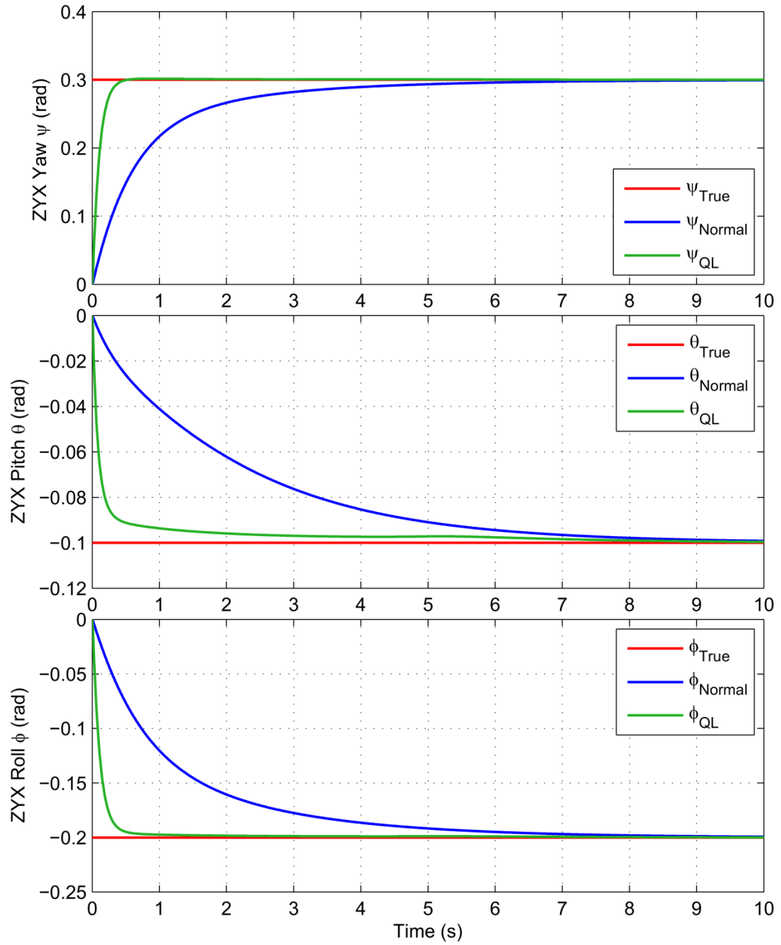


Figure 2: Simulated attitude estimation results demonstrating the effect of quick learning (with a quick learning time of 3.0s) on the filter's transient response. The simulated estimator was subjected to a step in orientation.

Robust Sensor Fusion for Robot Attitude Estimation

Open-source C++ Library Implementation

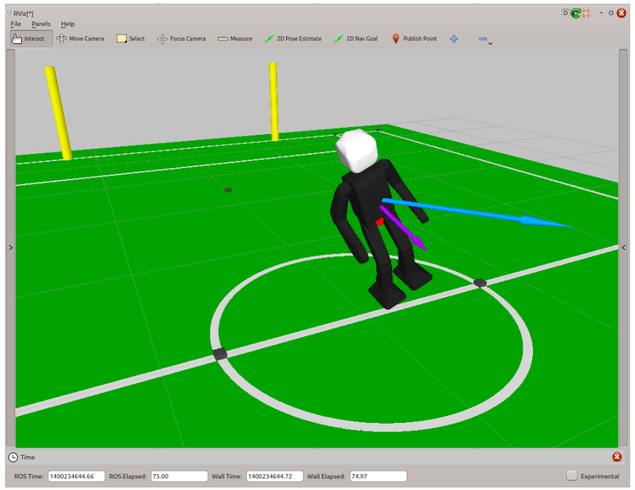
Attitude Estimator

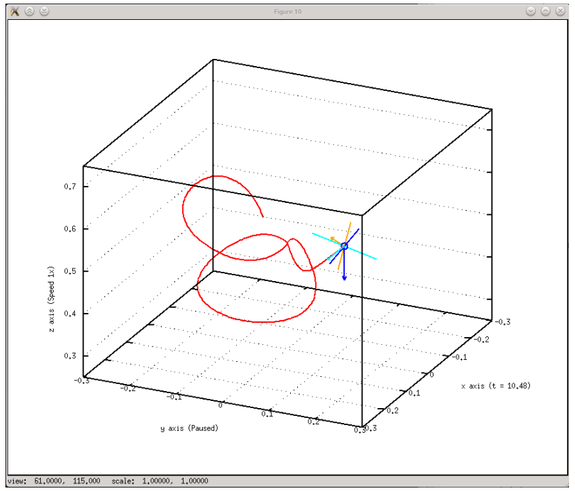
A C++ implementation of a nonlinear 3D IMU fusion algorithm

https://github.com/AIS-Bonn/attitude_estimator

Attitude Estimator is a generic platform-independent C++ library that implements an IMU sensor fusion algorithm. Up to 3-axis gyroscope, accelerometer and magnetometer data can be processed into a full 3D quaternion orientation estimate, with the use of a nonlinear Passive Complementary Filter. The library is targeted at robotic applications, but is by no means limited to this. Features of the estimator include gyro bias estimation, transient quick learning, multiple estimation algorithms, tuneable estimator parameters, and near-global stability backed by theoretical analysis.

Great emphasis has been placed on having a very efficient, yet totally numerically and algorithmically robust implementation of the filter. The code size has also been kept to a minimum, and has been extremely well-commented. The programmatic interface has also been made as easy as possible. Please refer to the extensive documentation of the library for more information on its capabilities and usage caveats.





GitHub This repository Search Explore Features Enterprise Blog Sign up Sign in

AIS-Bonn / attitude_estimator ★ Star 0 Fork 0

A C++ implementation of a nonlinear 3D IMU fusion algorithm

5 commits 1 branch 0 releases 1 contributor

branch: master attitude_estimator / +

Makefile/Doxyfile addendum, modify .gitignore/README.md

pallgeuer authored 17 days ago latest commit 44580181c2

doc	Makefile/Doxyfile addendum, modify .gitignore/README.md	17 days ago
src	Release: Attitude Estimator v1.2.0	3 months ago
test	Release: Attitude Estimator v1.2.0	3 months ago
.gitignore	Makefile/Doxyfile addendum, modify .gitignore/README.md	17 days ago
LICENSE	Release: Attitude Estimator v1.0.1	6 months ago
Makefile	Makefile/Doxyfile addendum, modify .gitignore/README.md	17 days ago
README.md	Makefile/Doxyfile addendum, modify .gitignore/README.md	17 days ago

Code Issues 0 Pull Requests 0 Pulse Graphs

HTTPS clone URL https://github.com/

You can clone with HTTPS or Subversion.

Clone in Desktop Download ZIP

README.md

Attitude Estimator

Author: Philipp Allgeuer

Version: 1.2.0

Date: 14/08/14

General Overview

Attitude Estimator is a generic platform-independent C++ library that implements an IMU sensor fusion algorithm. Up to 3-axis gyroscope, accelerometer and magnetometer data can be processed into a full

Notation and Identities

Definition: 2-Sphere

The **2-sphere** S^2 is the set of all unit vectors in $\mathbb{R}^3$, that is,

$$S^2 = \{\hat{\mathbf{u}} \in \mathbb{R}^3 : \|\hat{\mathbf{u}}\| = 1\}.$$

Definition: Rotation matrices

A **rotation matrix** is a 3x3 orthogonal matrix with determinant +1 that can be used to represent a rotation. The set of all rotation matrices is the **special orthogonal group** SO(3), and is given by

$$\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times 3} : R^T R = \mathbb{I},\ \det(R) = 1\}.$$

For a rotation from coordinate frame {A} to frame {B}, we have that

$${}^A_B R = \begin{bmatrix} {}^A\mathbf{x}_B & {}^A\mathbf{y}_B & {}^A\mathbf{z}_B \end{bmatrix} = \begin{bmatrix} {}^B\mathbf{x}_A & {}^B\mathbf{y}_A & {}^B\mathbf{z}_A \end{bmatrix}^T,$$

where for example ${}^A\mathbf{y}_B$ is the column vector corresponding to the y-axis of frame {B}, expressed in the coordinates of frame {A}.

Definition: Quaternions

The set of all **quaternions** $\mathbb{H}$, and the subset $\mathbb{Q}$ of pure rotations, are given by

$$\mathbb{H} = \{q = (q_0, \mathbf{q}) = (w, x, y, z) \in \mathbb{R}^4\}, \quad \mathbb{Q} = \{q \in \mathbb{H} : |q| = 1\}.$$

Rotation of a vector by a quaternion is most efficiently computed as

$$L_q(\mathbf{v}) = q\mathbf{v}q^* = \mathbf{v} + q_0\mathbf{t} + \mathbf{q} \times \mathbf{t}, \quad \text{where } \mathbf{t} = 2(\mathbf{q} \times \mathbf{v}).$$

Definition: Parallel and antiparallel

Two linearly dependent vectors are **parallel** if they are a positive multiple of each other, and **antiparallel** if they are a negative multiple of each other.

Identity: Conversions between quaternions and rotation matrices

Given a quaternion q = (w, x, y, z), the **equivalent rotation matrix** is given by

$$R = \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & 1-2(x^2+z^2) & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & 1-2(x^2+y^2) \end{bmatrix}.$$

Given a rotation matrix R, the **equivalent quaternion** is given by

If $R_{11} + R_{22} + R_{33} \geq 0$,

$$r = \sqrt{1 + R_{11} + R_{22} + R_{33}}$$
$$q = \left(\tfrac{1}{2}r, \tfrac{1}{2r}(R_{32} - R_{23}), \tfrac{1}{2r}(R_{13} - R_{31}), \tfrac{1}{2r}(R_{21} - R_{12})\right)$$

Else if $R_{33} \geq R_{22}$ and $R_{33} \geq R_{11}$,

$$r = \sqrt{1 - R_{11} - R_{22} + R_{33}}$$
$$q = \left(\tfrac{1}{2r}(R_{21} - R_{12}), \tfrac{1}{2r}(R_{13} + R_{31}), \tfrac{1}{2r}(R_{32} + R_{23}), \tfrac{1}{2}r\right)$$

Else if $R_{22} \geq R_{11}$,

$$r = \sqrt{1 - R_{11} + R_{22} - R_{33}}$$
$$q = \left(\tfrac{1}{2r}(R_{13} - R_{31}), \tfrac{1}{2r}(R_{21} + R_{12}), \tfrac{1}{2}r, \tfrac{1}{2r}(R_{32} + R_{23})\right)$$

And otherwise,

$$r = \sqrt{1 + R_{11} - R_{22} - R_{33}}$$
$$q = \left(\tfrac{1}{2r}(R_{32} - R_{23}), \tfrac{1}{2}r, \tfrac{1}{2r}(R_{21} + R_{12}), \tfrac{1}{2r}(R_{13} + R_{31})\right).$$