Strapdown-rs - A simple strapdown INS implementation

HTML: JOSS Submitted

Markdown: JOSS Submitted

crates.io v0.4.0 docs passing license MIT

Strapdown-rs is a straightforward strapdown inertial navigation system (INS) implementation in Rust. It is designed to be simple and easy to understand, making it a great starting point for those interested in learning about or implementing strapdown INS algorithms. It is currently under active development.

The primary contributions of this project are:

1. A source library of strapdown INS algorithms and utilities (`strapdown-core`: /core).
2. A program for simulating INS performance in various GNSS conditions (`strapdown-sim`: /sim).
3. A dataset of smartphone-based MEMS IMU and GNSS data (`strapdown-data`: /data).

Additionally experimental research is being conducted on improving the accuracy and robustness of strapdown INS algorithms under degraded and denied GNSS conditions by providing an alternative PNT solution in the form of geophysical anomaly data (`strapdown-geonav`: /geonav). This is an area of active research.

Installation

To use `strapdown-rs`, you can add it as a dependency in your `Cargo.toml` file: `cargo add strapdown-rs`. You can install the whole package or just the core library. Similarly you can install the simulation binary `cargo install strapdown-sim`.

Summary

`strapdown-rs` is a Rust-based software library for implementing strapdown inertial navigation systems (INS). It provides core functionality for processing inertial measurement unit (IMU) data to estimate position, velocity, and orientation using a strapdown mechanization model that is typical of modern systems particularly in the low size, weight, and power (low SWaP) domain (cell phones, drones, robotics, UAVs, UUVs, etc.). Additionally, it provides some basic simulation capabilities for simulating INS scenarios (e.g. dead reckoning, closed-loop INS, intermitent GPS, GPS degradation, etc.).

`strapdown-rs` prioritizes correctness, numerical stability, and performance. It is built with extensibility in mind, allowing researchers and engineers to implement additional filtering, sensor fusion, or aiding algorithms on top of the base INS framework. This library is not intended to be a full-featured INS solution, notably it does not have code for processing raw IMU or GPS signals and only implements a loosely-couple INS.

The toolbox is designed for research, teaching, and development purposes and aims to serve the broader robotics, aerospace, and autonomous systems communities. The intent is to provide a high-performance, memory-safe, and cross-platform implementation of strapdown INS algorithms that can be easily integrated into existing systems. The simulation is intended to be used for testing and verifying the correctness of the INS algorithms, by providing a simple simulation that allows users to generate a "ground truth" trajectory.

Functionality

`strapdown-rs` is intended to be both a source code library included into your INS software and simulation environment as well as very light-weight INS simulator. The library provides a set of modules modeling the WGS84 Earth ellipsoid, a common 9-state strapdown forward mechanization, and a set of navigation filters for estimating position, velocity, and orientation from inertial measurement unit (IMU) data.

The simulation program provides a simple command line interface for running various configurations of the INS. In can run in open-loop (dead reckoning) mode or closed-loop (full state loosely couple UKF) mode. It can simulate various scenarios such as intermittent GPS, GPS degradation, and more. The simulation is designed to be easy to use and provides a simple API for generating datsets for further navigation processing or research.