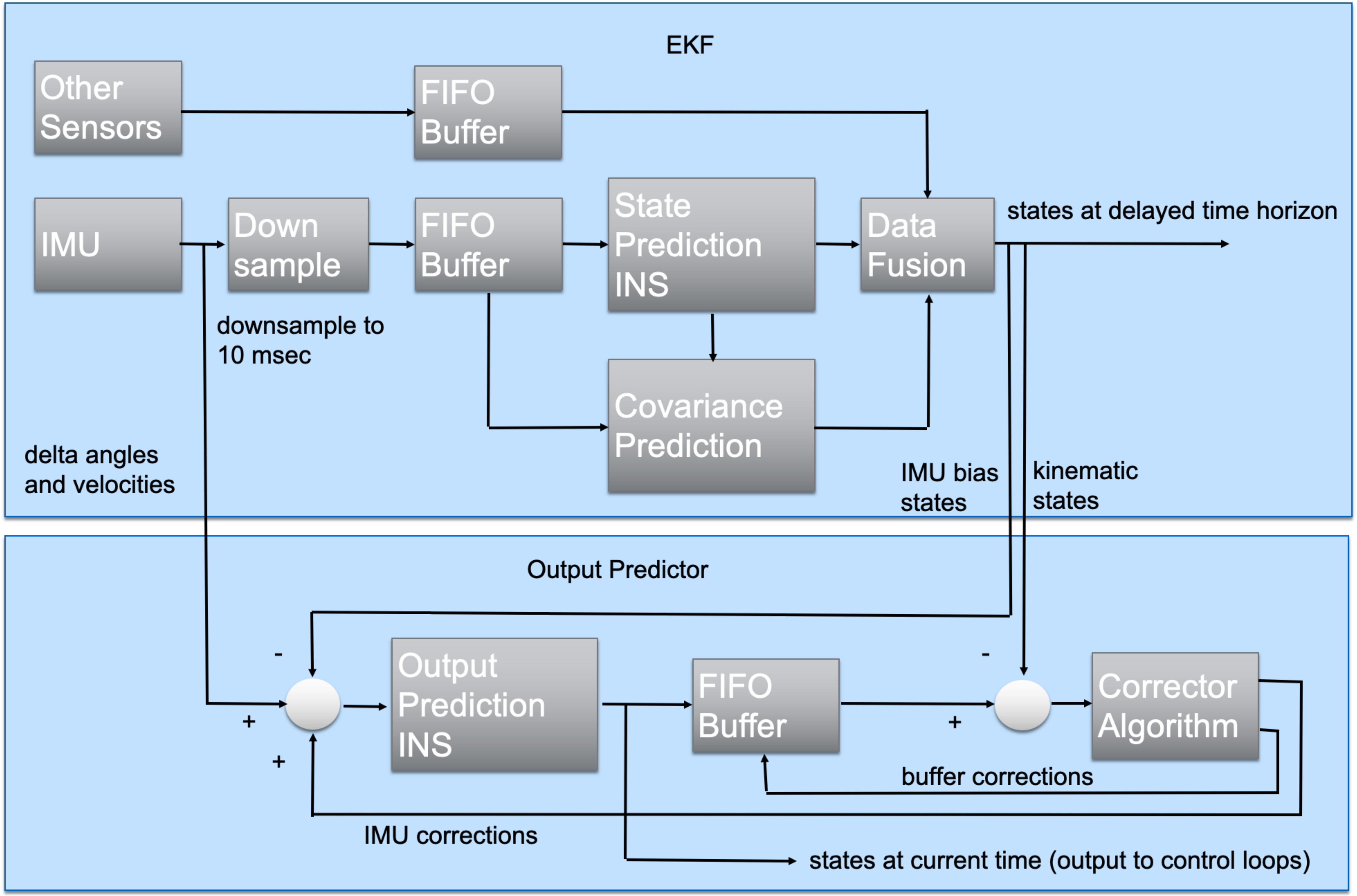
‘PX4 State Estimation’
PX4 Developer Summit
Zurich 2020

Paul Riseborough

Overview

- Vehicle state estimation is performed by the PX4 ekf2 module. It uses an Extended Kalman Filter (EKF) algorithm from the PX4/ecl EKF library
  - 24 States
    - Angular Orientation (4 quaternion states)
    - Velocity (3 states)
    - Position (3 states)
    - IMU and Magnetometer Bias (9 states)
    - Earth Magnetic Field (3 states)
    - Wind Velocity (2 states)
  - ~8000 SLOC (Executable)
  - 43 contributors

Architecture



Status

Since the last conference we have:

- Merged 122 pull requests
  - 40 functionality & performance enhancements
  - 30 bug fixes
  - 4 ‘potential bug’ fixes
  - 43 non-functional ‘house keeping’
  - 6 test framework enhancements
- Closed 17 issues

At the close of June 2020 we had

- 5 Open pull requests
  - 5 enhancements
- 17 Open issues
  - 7 enhancement requests
  - 3 potential bugs
  - 5 bugs
  - 2 queries

Overview of Enhancements

- Yaw alignment using only IMU and GPS velocity, #770, #831, #826, #789, #766,
  - Automatically correction for yaw error induced post takeoff loss of navigation.
  - Operation without a magnetometer or external yaw sensor if EKF2_MAG_TYPE = 5
- Compensation for accelerometer clipping effects, #663
  - EKF acceleration process noise is increased when clipping events are reported in the IMU data. Enabled by changes to IMU sampling.
- Improved rejection of bad magnetometer data #638, #662,
  - Enable use of synthetic Z axis magnetometer data
  - Magnetometer fusion is inhibited when the norm of the magnetic field significantly exceeds the expected strength from the World magnetic Model table.

Overview of Enhancements

- Improved support for external vision based navigation sources #634, #642, #668, #680, #708, #709, #721, #731
  - Improved diagnostics logging
  - Improved fallback behavior and operation with GPS
  - Use of velocity data when available for improved accuracy
  - Use of covariance data when available for improved accuracy
- Range Finder and Optical Flow usage improvements, #639, #733
  - Improves operation when operating with a range finder that returns invalid data when on and close to ground

Overview of Test Enhancements

- Support for unit testing, #657, #685, #689, #715
  - Uses Google GTest library
  - Over 50 tests implemented to date
- Automatic reversion testing of pull requests using sensor replay data, #717, #838
  - Sensor data generated using simulation or flight test is used as a reference data set.
  - All EKF states and their corresponding variances are compared before and after the change. Exact equivalence is required for this test to pass.
  - Developers are encouraged to split pull requests into functional vs non-functional changes to maximize benefit of reversion testing.

Yaw Alignment Using IMU and GPS

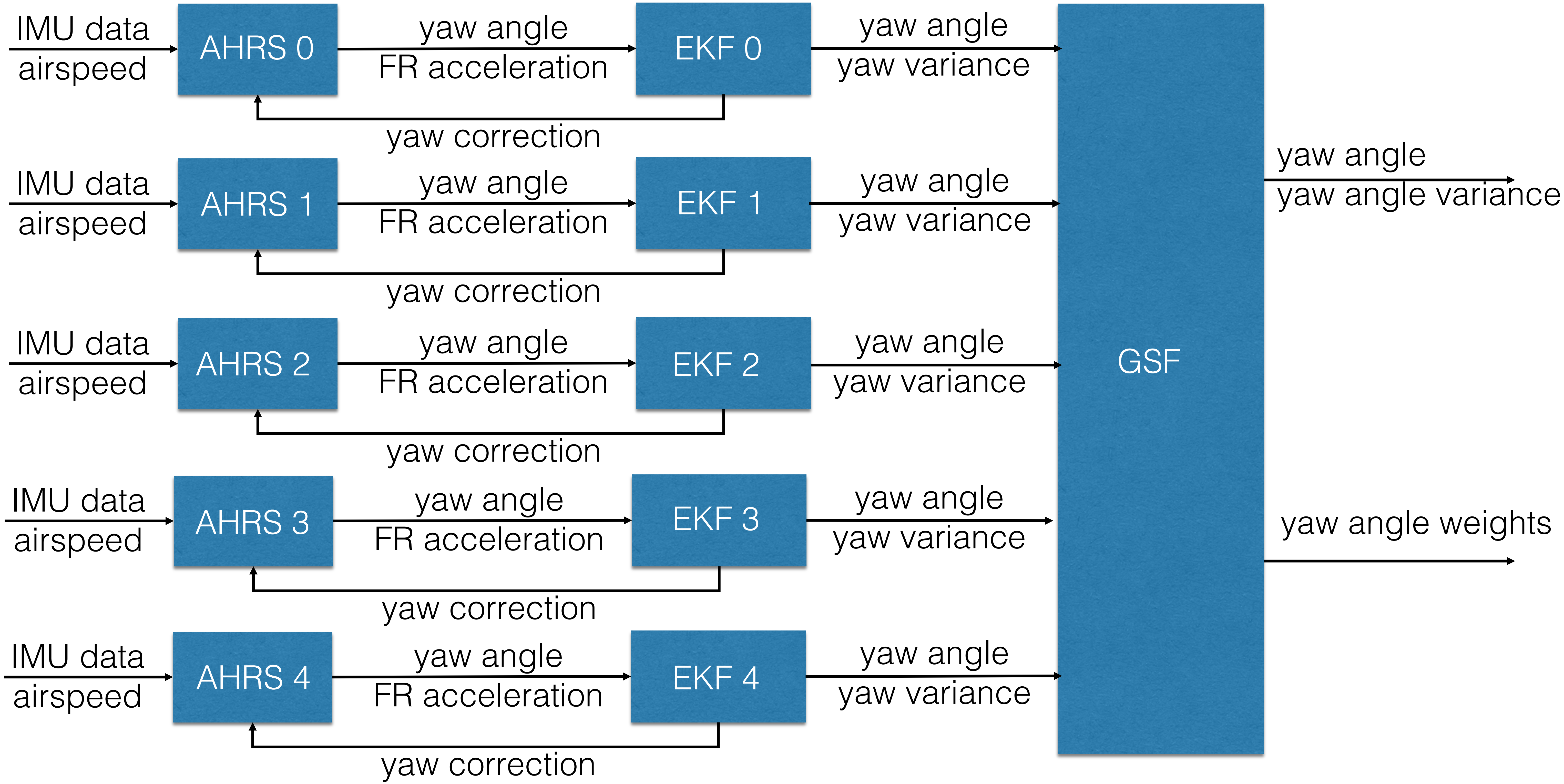
- Preliminary algorithm work performed in 2018 to determine if it was possible to rapidly determine the yaw angle using only IMU and GPS Velocity data without relying on assumptions about the vehicle dynamics.
- Investigation used simulated data and compared performance of different 3-state filter designs:
  - EKF (Extended Kalman Filter)
  - UKF (Unscented Kalman Filter)
  - EKF-GSF (Gaussian Sum Filter using states from multiple EKF’s)
  - UKF-GSF (Gaussian Sum Filter using states from multiple UKF’s)
  - EKF-IMM (Interacting Multiple Model Filter using states from multiple EKF’s)
  - UKF-IMM (Interacting Multiple Model Filter using states from multiple UKF’s)
  - Particle Filter
- The GSF-EKF provided the best performance/computational cost trade-off for our application
- *Acknowledgements: Most of the matlab scripting for the preliminary study was contributed by Rudaba Khan*

EKF-GSF Yaw Estimator Description

The GSF-EKF algorithm consists of the following:

- A bank of five AHRS solutions using a complementary filter
  - These calculate a predicted yaw angle and a forward, right acceleration.
  - Airspeed (measured or estimated) is used for centripetal acceleration correction during fixed wing flight.
- A bank of five 3-state Extended Kalman Filters
  - States are North, East velocity and yaw angle
  - Yaw angle estimates start equally spaced
  - GPS North, East velocity is used as the observation
- A Gaussian Sum Filter
  - Calculates a weighting for each EKF based on the normalized NE GPS velocity innovation levels.
  - Outputs a yaw angle estimate which is the weighted average of the individual EKF estimates.

EKF-GSF Yaw Estimator Diagram

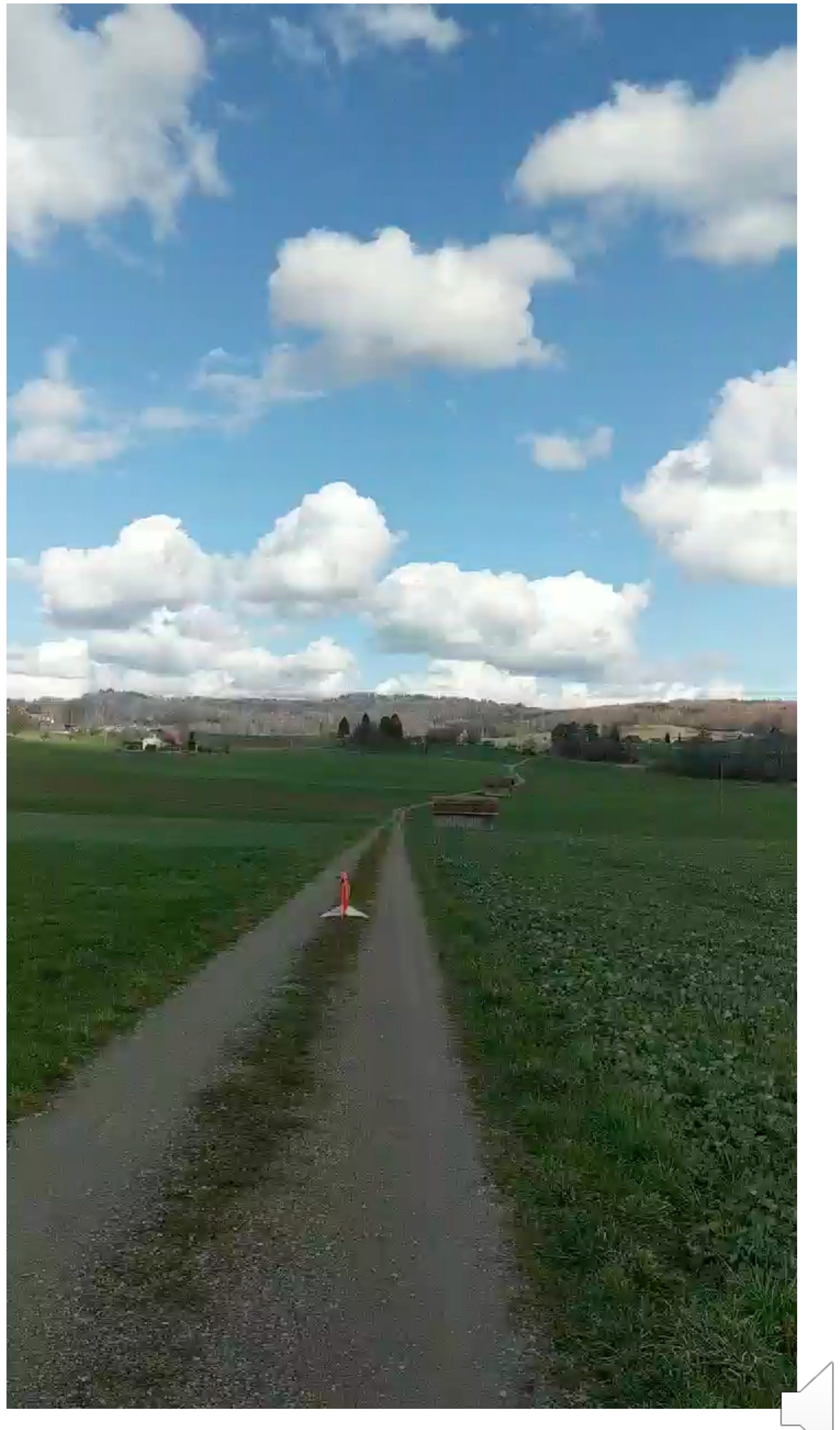


EKF-GSF Yaw Estimator Status

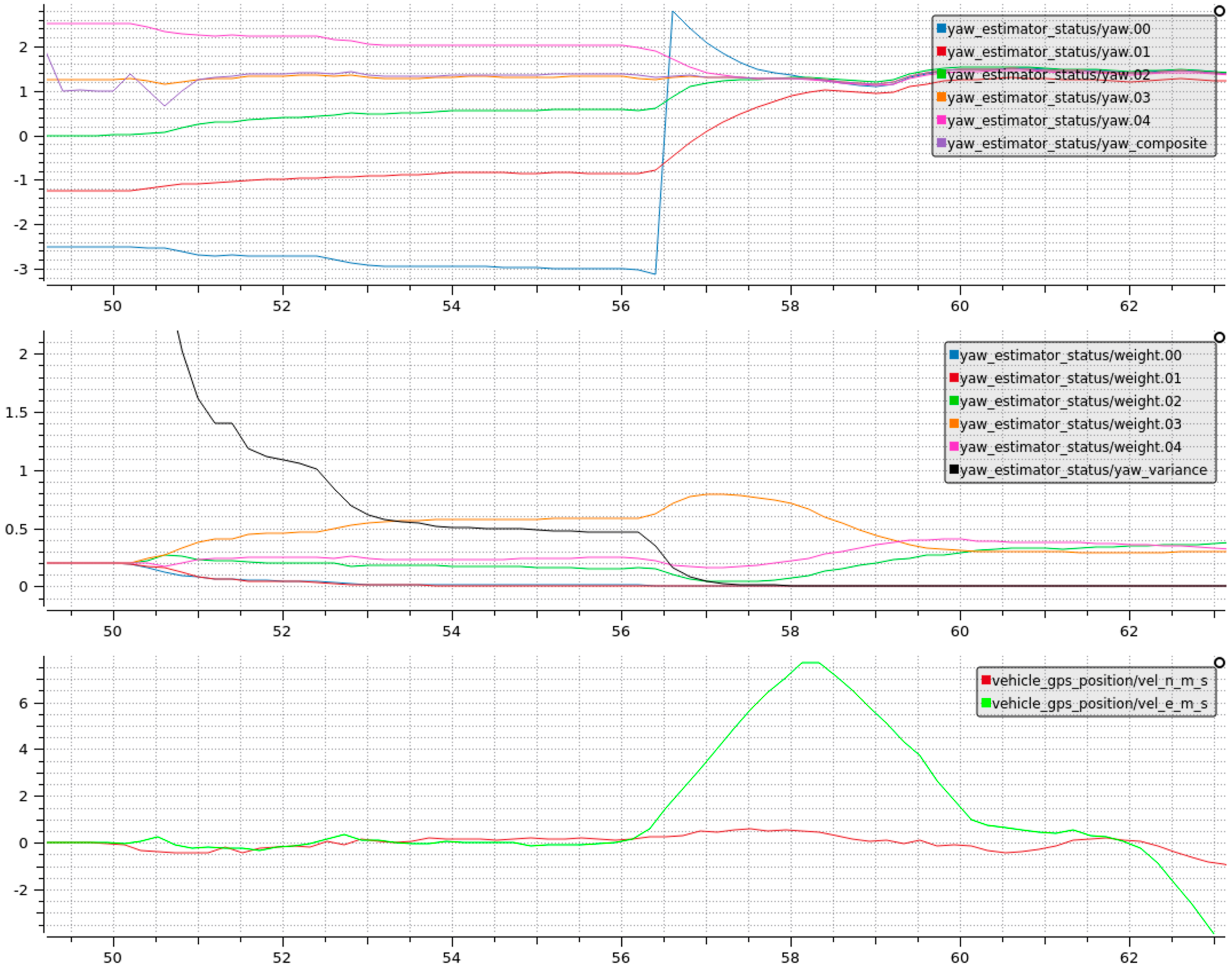
- Support from multiple sponsors (Wingtra, Auterion) enabled previous algorithm development work to be taken to a 6DoF prototype in Matlab followed by coding for the PX4/ecl library.
- The feature is now available in PX4/Firmware master and will be part of the 1.11 release. It adds:
  - Rapid automatic recovery from bad magnetic yaw induced post take-off navigation failure.
  - Operation without a magnetometer or use of RTK GPS Yaw if EKF2_MAG_TYPE parameter is set to 5. Take-off and initial horizontal movement must be in a mode that doesn’t require a position fix, eg Altitude. The EKF will automatically align yaw and commence GPS use, enabling use of position fix modes.

EKF-GSF Yaw Estimator Testing

- Flight testing to date has demonstrated:
  - Multicopter takeoff in PosCtrl and Mission mode with declination set to a value up to 180 degrees from truth. The navigation recovered and the flyaway stopped before the Commander failsafe activated.
  - Multicopter take-off in Altitude control mode with compass use inhibited. Navigation started and PosCtrl could be selected a few seconds after takeoff with less than 4m of horizontal movement required.
  - Flights with a Wingtra TailSitter (PX4/ecl) set up with 180 degree compass error. Navigation recovered and the flyaway stopped before the failsafe activated.
- CPU load increase measured on a STM32-F4 is <1%



EKF-GSF Yaw Estimator Test Results



The GSF weights converge before the individual EKF filters.

Horizontal velocity change greater than GPS velocity uncertainty is required.

Convergence is faster with a larger velocity change.

Large velocity changes due to loss of navigation result in convergence within 1 second.

Future Estimation Work

- Continue to improve tolerance to sensor faults
- Continue to expand handling of non-GPS navigation sensors
- Continue to increase unit test coverage.