POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Aerospaziale

Tesi Magistrale

Processor in the loop simulations and Code generation for Unmanned Aerial systems

Relatore:
Prof.ssa Elisa Capello

Co-Relatore:
Dott. Davide Carminati

Autore:
Luigi Sante

Dicembre 2020

Sommario

TODO : Parlare di cosa tratta la tesi

Ringraziamenti

TODO : Scrivere i ringraziamenti

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Capitolo 2

Letteratura

Capitolo 3

Sistema Quadrirotore

3.1 Descrizione del drone

3.2 PX4 Autopilot

Il firmware utilizzato nelle simulazioni di software in the loop e processor in the loop è il PX4 Autopilot.

Questo software mette a disposizione diverse funzionalità per avere un sistema di gestione e controllo robusto e affidabile, implementato in diversi tipi di sistemi. L'implementazione non è quindi specifica solo a mezzi aerei di qualsiasi configurazione, ma anche a velicoli di terra, marini e razzi. Il software è open-source e vanta del contributo di parecchi sviluppatori, dagli esperti del settore a contributi di livello accademico. Lo sviluppo open-source permette quindi di aggiungere o modificare le funzionalità messe a disposizione in modo da soddisfare le proprie esigenze e arricchire il progetto generico di nuove funzionalità utili ad altri. Il sistema operativo sulla quale viene eseguito materialmente il codice può essere Nuttx o Linux/MacOS la cui distinzione principale in questa applicazione è solo nella gestione di task e thread.

Il sistema operativo Nuttx è un sistema RTOS (Real-Time Operating System) è svilupato appositamente per implementazioni embedeed. Essendo sviluppato per un contesto specifico ha tutte le caratteristiche necessarie per essere eseguito in sistemi che devono avere prestazioni migliori con poche risorse disponibili. Vengono utilizzati gli standard POSIX e ANSI. Inoltre, sono implementate funzionalità di programmazione concorrenziale per l'esecuzione di processi in parallelo. Le funzionalità del firmware vengono eseguite in questo sistema come task separati e ogni task può eseguire diversi thread. Nell'implementazione su sistemi Linux/MacOS invece i moduli sono eseguiti come thread del processo principale, non c'è quindi una distinzione tra threads e tasks, oltre a non essere ottimizzato per sistemi embedeed.

3.2.1 Architettura del software

Il firmware è principalmente suddiviso in due categorie di moduli:

- **Flight stack** : composta dalla parte che stima lo stato del sistema e il relativo controllo

- **Middleware** : composta dalle interfacce che collegano i vari moduli interni di PX4 tra di loro e verso l'esterno, con la possibilità di integrare gli hardware utilizzati.

Il sistema quindi separa le varie funzionalità in moduli separati, eseguiti in modo indipendente che scambiano i dati e comandi tra di loro e con l'esterno attraverso messaggi asincroni. Nella figura 3.1 è riportato lo schema di alto livello del software di PX4 e la sua modularità.

Flight stack

Il flight stack, mostrato in figura 3.2 è l'insieme di moduli che si occupano della stima dello stato del sistema e di tutti le funzionalità per il controllo,la guida e la navigazione. Esiste anche un modulo per interfacciarsi con il volo manuale attraverso radiocomando.

Estimator L'estimator è il modulo che prendendo i dati da uno o più sensori determina lo stato attuale del velivolo. E' possibile selezionare diversi tipi di estimato, quello selezionato in questo caso è uno stimatore con filtro di Kalman.

Controller Si occupa di prendere in input i vari punti della pianificazione e confrontarli con lo stato attuale determinato dell'estimator. In questo modo vengono determinati i segnali di comando di output che saranno poi elaborati dal mixer. Questa funzionalità è implementata da più moduli, suddividendo la dinamica lenta e la dinamica veloce del drone. Verrà disabilitata la sua funzionalità per sostituirla con l'applicazione sviluppata su Simulink.

Mixer Il mixer si occupa di tradurre i segnali del comandi normalizzati di rollio, imbardata, beccheggio e manetta, da consegnare all'hardware che genera gli impulsi pwm utilizzati per il controllo del motore.

Middleware

Questo insieme di moduli si occupa invece di tutte le comunicazioni interne tra processi e tra PX4 e il mondo esterno. É composta principalmente dai driver per i sensori, i canali di comunicazione verso l'esterno e il bus di trasmissione di messaggi attraverso µORB e MAVLink. In questo contesto ricade anche la connessione con un simulatore per testare il codice generato nelle varie fasi di validazione.

3.2.2 Strumenti per lo sviluppo del codice

L'intero codice del firmware PX4 viene messo a disposizione attraverso la piattaforma github. Il progetto contiene all'interno le toolchain necessarie per compilare il sistema nei vari sistemi operativi. Agendo sulle varie possibili configurazioni di compilazione è possibile modificare e aggiunge delle funzionalità. Modificando le impostazioni di compilazione si può generare il programma finale da caricare ed eseguire sull'autopilota. Sono presenti anche delle configurazioni per effettuare l'analisi e la verifica del codice generato attraverso l'utilizzo di un ambiente simulato. I simulatori che presentano una configurazione di base sono : Gazebo, jMAVSim , AirSim, Xplane. Per quanto riguarda questa tesi, verrà utilizzato il software Gazebo sfruttando parzialmente il codice già presente e adottando alcune modifiche.

Nulla vieta comunque di poter collegare un simulatore diverso attraverso la creazione di un interfaccia dati con il firmware. Infatti, la connessione viene effettuata attraverso UDP o seriale, utilizzando su essa il protocollo MAVLink.

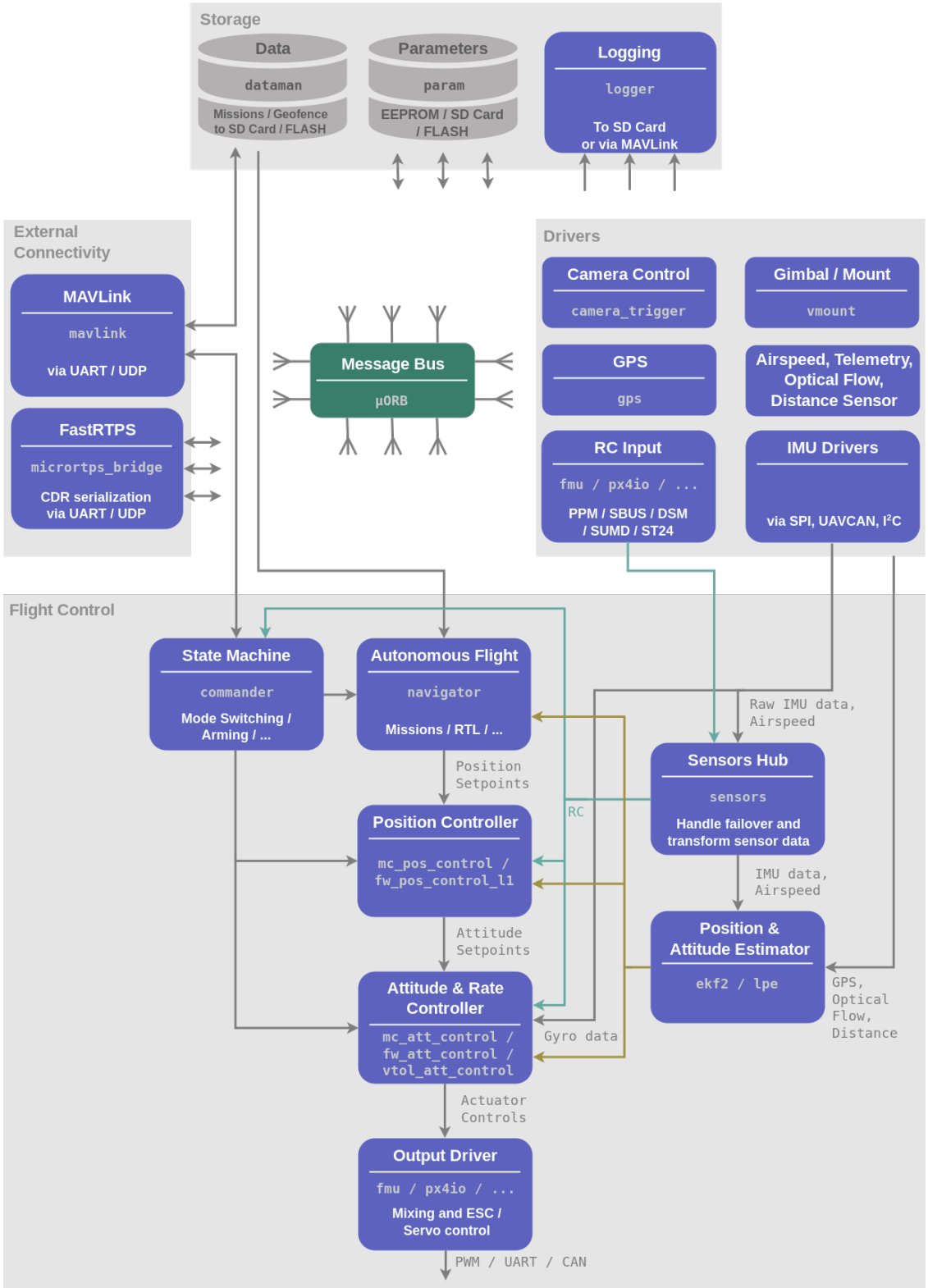

Figura 3.1. Architettura del codice di PX4 Autopilot

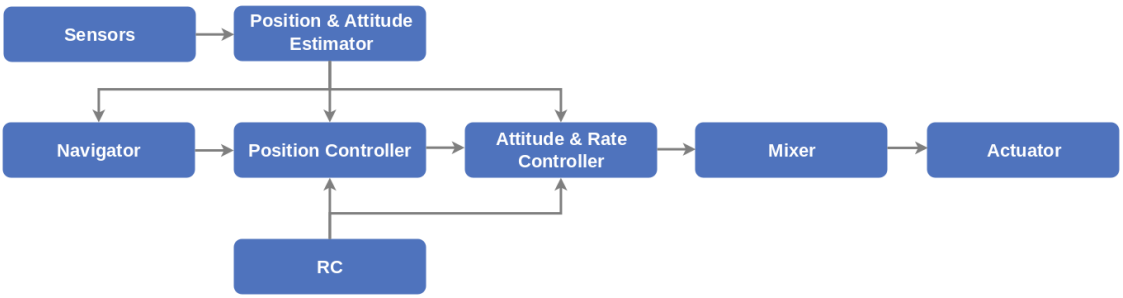

Figura 3.2. Architettura del flight stack di PX4

3.3 Modello matematico

3.4 Leggi di controllo

3.5 Algoritmo di guida

Capitolo 4

Simulazioni

4.1 Simulazione SITL

4.2 Conclusioni

Bibliografia

[1] Davide Carminati. «Design and Testing of Indoor UAS Control Techniques». Politecnico di Torino, 2019.

[2] *MAVLink Developer Guide*. Dronecode Project. 2020. URL: https://mavlink.io/en/ (visitato il 30/03/2020).

[3] *NuttX*. Wikipedia. 2020. URL: https://it.wikipedia.org/wiki/NuttX (visitato il 30/03/2020).

[4] *PX4 Autopilot User Guide (1.8.2)*. PX4 Dev Team. 2020. URL: https://docs.px4.io/v1.8.0/en/#px4-autopilot-user-guide--180 (visitato il 30/03/2020).

[5] *PX4 Development Guide (v1.8.0)*. PX4 Dev Team. 2020. URL: https://dev.px4.io/v1.8.0/en/index.html#px4-development-guide-v180 (visitato il 30/03/2020).

[6] Inc. The MathWorks. *PX4 Autopilots Support from Embedded Coder*. 2020. URL: https://it.mathworks.com/hardware-support/px4-autopilots.html (visitato il 30/03/2020).