Platooning II - This is a temporary title

***Abstract*—[ref to platooning paper] proposed a formulation for platooning UAVs to make reachability analysis on UAVs in traffic infrastructures tractable. In [ref to platooning paper], the theory and scenario studies, through virtual simulation, showcase the potential of the formulation for ensuring liveness and safety guarantees on vehicles in transit. To show the efficacy of platooning on real systems we implemented the formulation on real quadrotors. Due to the curse of dimensionality, HJ reachability-based control is rarely implemented for quadrotors. In this paper we present a novel hardware implementation architecture for platooning with both real UAV's and real information patterns. We demonstrate the efficacy of our implementation by showcasing the results of running the system through three scenario studies.**

I. Introduction

** Similar to the lab's usual UTM intro, but talk about hardware emphasize both the importance of a scalable airspace architecture and the importance of hardware implementation **

II. Background

** Lots of referencing of Platooning I and specifying where this paper expands on Platooning I **

A. UTM Hybrid System

** Walk through hybrid system and talk about both dynamics and control of hybrid system **

- Vehicles listen to either safety or liveness controls broadcasted by the TFM node and liveness node, respectively, where guards are triggered by TFM.
- TFM computes safety within platoons and for platoon to platoon. It sets boolean of isSafe for each.
- Liveness listens to state of vehicle, mode switches, and broadcasts appropriate controls for vehicles to reach their targets through the air highway system.
- Illustrated in Fig 1

B. Platooning

1) Full Quadrotor Dynamics: ** Quick section on full 12 state model of quadrotor dynamics and near-hover linearization **

2) Lower-Dimensional Vehicle Dynamics: ** The vehicle dynamics from Platooning I that are used to compute reachable sets for this implementation, reference later section that goes over LQR framework**

3) Platoon Structure: ** Platoon formulation and structure **

4) Liveness: ** Mention of the meaning of liveness and what behaviours it entails**

5) Safety: ** Talk about what is done by the centralized traffic flow manager **

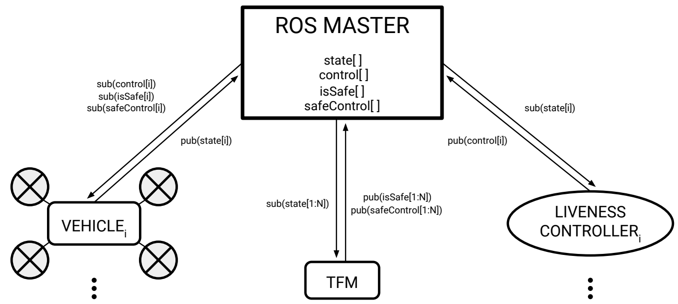


Fig. 2. ROS Network governing the platooning implementation.

III. Implementation

A. Hardware Implementation

1) Hardware Environment: ** Hardware that is being used and their real world analogs. Expand on how the real world would more likely work? **

- Vehicle: Crazyflie 2.0.
- Localization: VICON. Acts in place of a sensor fusion between GPS, accelerometer, vision and lidar. **Expand on why so many things fused**
- Controller: Liveness ROS node running on laptop connected to LAN.
- Centralized air traffic control: TFM ROS node running on laptop connected to LAN.

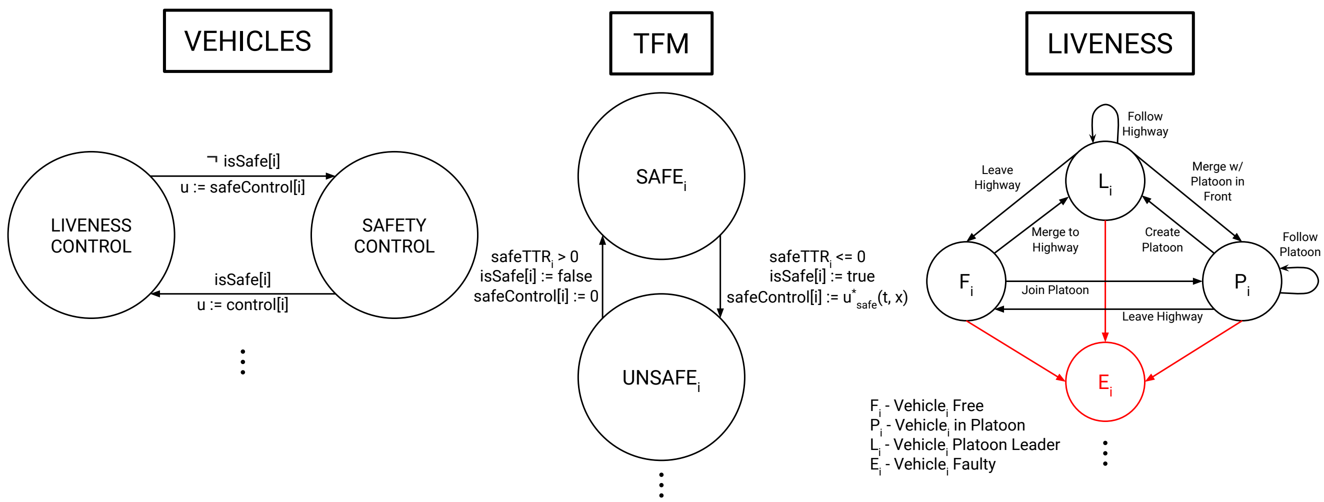
2) Communications Infrastructure: ** The ROS communications layer and why it is structured the way it is ** [ref to wolfgang's work]

- Crazyflie ROS Node (Subscribes to controls and isSafe, publishes state estimation)
- Server ROS Node (Hosts all the Crazyflie ROS Nodes)
- Liveness ROS Node (Subscribes to state estimation, published liveness controls)
- TFM ROS Node (Subscribes to state estimation, publishes isSafe and safe controls)

B. Reachability on real systems

** Talk about the numerical challenges and solutions of implementing reachability on real dynamical systems **

- Work with windows/intervals instead of single real values/floats
- Use minimum real information to mitigate noise (eg. only using y-coordinate and assumption that vehicle being followed is already on highway)
- Convex blending for reach set edges (prevent bang-bang situations)



Left: Model of hybrid control of the UAVs; Center: Finite automatons managed by the third-party TFM; Right: Hybrid system that computes the liveness controls for the UAVs.

Fig. 1. Finite automaton model of the entire platooning hybrid system

C. Controller

1) LQR Framework: [ref to CDC system ID paper] ** Detail the LQR framework being used for this implementation and the time-invariant near hover K matrix being used **

2) Enforcing lower-dimensional reachability: ** How guarantees on a lower-dimensional system are being enforced in the full higher-dimensional state space of quadrotor dynamics **

- Roll-pitch-yaw already assumed to be zero for near hover assumption
- Only control second order behaviour of position and first order behaviour of velocity in XY-plane.

IV. Case-Studies

A. Forming a platoon

** Forming three vehicle platoon **

B. Intruder on platoon

** Platoon reacting to a single intruder **

C. Changing platoons

** One vehicle leaving a three vehicle platoon to join a two-vehicle platoon to form new three vehicle platoon **

V. Conclusion

References

[1] ref to platooning I
[2] ref to cfnnet
[3] ref to wolfgang's repo
[4] ref to ROS