CPS Virtual Organization Challenge

CPS-VO2 (version: 13 Jan 2016)

# Summary:

We invite university students from the University of Arizona, University of California, Los Angeles, University of Pennsylvania, and Vanderbilt University to participate in the 2015 Cyber-Physical Systems Virtual Organization Challenge. The Challenge will provide engineering students the opportunity to validate their analytic studies through a real-world vehicle design and verification experience.

Student teams will modify an existing quadrotor airframe and its on-board computers, and its payload, in order to demonstrate the flying and maneuvering capabilities of an unmanned, electric radio controlled or autonomous vehicle that can best meet the specified mission profile. The goal is a balanced design possessing good demonstrated qualities with a payload and without, with reliable pickup/dropoff of payload, scalability of their solution, and high overall vehicle performance. A premium is placed on the use of simulation as part of the scoring of the competition in early phases, in order to stress analysis and simulation over in-situ experimentation, and to provide interfaces for the verification of system parameters.

# Judging

# Schedule

## Framework schedule:

# Design

## Vehicle

The Vehicle is based off of the DJI Flame Wheel F450 airframe with the Pixhawk flight management unit (FMU). The vehicle must be capable of transport of the Payload (as designed and modified by each team), since Payload retrieval, transport, and dropoff are necessary portions of the Challenge.

The Vehicle can be modified in any way by each team in terms of the addition or removal of stock sensors, replacement of batteries, and addition of payload retrieval, transport, and dropoff mechanism(s).

GPS will be available during the final competition, and teams are welcome to use existing or new sensors on board the Vehicle to improve localization. You are not allowed to modify the environment to improve localization.

## Payload (Trap)

The Payload is based off of the Project Premonition insect Trap. When attached to the Vehicle it is called the Payload, when deployed for capturing insects, it is called the Trap. The Challenge will use a 1:2 scale version of the actual trap for Project Premonition, to provide a significant simplification of the challenges. The starting point for the Trap design is:

* Weight: 600g (or less)
* Trap diameter: 100mm
* Height: 150mm
* Trap top diameter: 75mm

Changes may be made to the Trap as part of the codesign of the Payload for pickup, transport, and dropoff. The most significant change will be the coupling to the Vehicle.

Teams may modify the Payload to have additional sensors, images, beacons, batteries, computing devices, etc. The limit is only in the requirement that the Payload must be retrieved, transported, and dropped off by the Vehicle.

Payload shells must be 3D printable, nothing that (after printing) modifications may be made to the shell in terms of adding sensors, actuators, batteries and computers.

Teams may design any number of Payloads during the Challenge, but a final design must be used during the Challenge’s Final Demonstration. Teams may bring backup Payloads, in case one is damaged as part of the Final Demonstration.

# Mission Components

## Takeoff

The Vehicle takeoff may be either with Payload, or without, depending on the mission.

## Retrieval

The Retrieval of the Payload may be with a human under control, or may be fully or semi-autonomous. Teams will design their components in order to maximize their score for the retrieval process, according to each mission. Retrieval should not damage the Payload

## Transport

The Transport of the Payload should be handled in autonomous mode.

## Dropoff

The Dropoff of the Payload may be with a human under control, or may be fully or semi-autonomous. Dropoff should not damage the payload.

## Waypoint

A waypoint tuple is composed of latitude, longitude, altitude, radius, and yaw (with yaw error). This permits a waypoint to be “near” an area, and the yaw angle gives an idea for the direction of forward travel when the vehicle enters the circle near that waypoint.

## Timed Waypoint

A timed waypoint tuple is composed of time in, time “radius”, latitude, longitude, altitude, radius, and yaw (with yaw error). This permits a waypoint to be “near” an area, and the yaw angle gives an idea for the direction of forward travel when the vehicle enters the circle near that waypoint. The time radius is the +/- time of entry into the waypoint radius.

# Missions for final demonstration

## Mission Sequence (general)

The sequence of mission events is described below:

1. Mission-specific details are shared with the team early on the day of the event
2. Team has a short time (on the order of hours) to respond with updated artifacts
3. Missions performed
4. Data acquired are used to score the mission.

Missions are also decomposed into two basic types. *Vehicle* challenges perform tasks with the vehicle in the loop. *System* challenges provide answers to system questions, based on the parameters measured in Vehicle missions.

* Vehicle Challenge 1: Will judge the team’s ability to design and implement the pickup and dropoff mechanism, and its associated logic.
* Vehicle Challenge 2: Rewards teams who can accurately predict the trajectory of their vehicle while transporting the Payload. Requires tuning of the on-board vehicle controllers, according to the Payload mechanism design.
* Vehicle Challenge 3: Rewards teams who can predict the times at which their vehicles will (or will not) be in certain locations when in operation, in order to operate at scale without “sense and avoid” capabilities.

## Vehicle Challenge 1: Time prediction for payload pickup/transport/dropoff

* Overview: Starting from any waypoint L0, pick up a payload from a defined waypoint L1, bring payload to second waypoint L2, and drop it off before going to waypoint L3 (a simulated recharge station) and landing at the recharge station.
* Three attempts are allowed
* After mission spec and after each mission run, team has 10 minutes to provide:
  + Software updates (if desired)
  + New time bound during which they think that the mission can be completed.
* Score = T\*M + k\*T\_ground – T\_auto
  + The lowest score, calculated by the product of the
    - T is the time required (in seconds), made up of
      * T\_air time the vehicle is in the air
      * T\_ground time the vehicle is on the ground
    - M is the time bound (in seconds), estimates as T +/- M
    - T\_auto is any time spent in autonomous mode during pickup and dropoff
    - k is a multiplier to penalize sitting near the final waypoint until T
  + Idea: how to multiply this if pickup and dropoff are autonomous?

## Vehicle Challenge 2: Measurement of deviation from trajectory with payload

* Overview: Waypoints (L0, L1…) provided, the vehicle (with payload already attached) takes off, flies to each of these waypoints, and lands at the last waypoint.
* Three attempts are allowed
* After mission spec and after each mission run, team has 10 minutes to provide:
  + updated model (if desired)
  + an expected trajectory, based on the simulated model flying the same waypoint set.
* Score: min( Score 1, Score 2 )
  + Score 1: sum of squares error to the nearest point of the straight lines drawn between each waypoint (within a certain error is 0, e.g., less than 1m)
  + Score 2: sum of squares error to the nearest point of the expected trajectory provided by the Team’s simulation (within a certain error is 0, e.g., less than 1m).

## Vehicle Challenge 3: Measurement of timed deviation from trajectory with payload

* Overview: Timed Waypoints (TL0, TL1…) provided, the vehicle (with payload already attached) takes off, flies to each of these waypoints and loiters (or lands) for a time centered around the required time, and arrives at the last waypoint at the required time.
* 1 attempt is allowed
* After mission spec, team has 10 minutes to provide:
  + updated model (if desired)
  + a model that explains the time range justification
  + an expected timed trajectory, based on the simulated model flying the same waypoint set, including time ranges during which they are guaranteed to be at each waypoint
* Score: min( Score 1, Score 2 )
  + Score 1: sum of squares error to the nearest point of the straight lines drawn between each waypoint (within a certain error is 0, e.g., less than 1m); the error is taken from the nearest point of the segment for which the time is valid
  + Score 2: sum of squares error to the nearest point of the expected trajectory provided by the Team’s simulation (within a certain error is 0, e.g., less than 1m); the error is taken from the nearest point of the trajectory for which time is valid.

## System Mission 1: Air-Traffic Controller

* Overview: Team must provide a timed sequence of waypoints to move the vehicles to their final locations, using values for vehicles from Vehicle Mission 3 above.
* 1 attempt (Note: scoring is done in simulation)
* Team receives an area description in which vehicles can fly (or not), a list of initial Vehicle locations, a list of final Vehicle locations, and the Vehicle Mission 3 parameters from each of the teams.
* Team provides
  + Timed waypoints sequence for each Vehicle
* Simulator
  + Executes the series of timed waypoints using (vehicle models, over approximations)
* Scoring
  + Minimum time to complete
  + Minimum time in the air for each vehicle
  + Subtract points if UAVs get within (distance 1) of one another; 0 points for the entire Mission if any UAVs get within (distance 2)

## System Mission 2: Strategic Deployment of Vehicles and Traps

* Overview: Team chooses how to allocate Traps in order to obtain reliable coverage for a disease with a specific incubation period
* 1 attempt is allowed (Note: mission is in simulation only)
* Team receives the description of an area for coverage, specified as an ordered set of Waypoints that describe a (potentially convex) polygon, one unique waypoint as the “warehouse,” a point where a collection of Traps is currently located, and models provided as part of Vehicle Mission 3.
* Team provides
  + locations (in Waypoint format) to deploy traps from the warehouse
  + locations of recharge stations (in Waypoint format)
  + initial location of UAVs to refresh the mosquito Traps (in Waypoint format)
  + plan to fly UAVs to each waypoint, avoiding other UAVs by design (not with sense-and-avoid)
* Simulator:
  + Uses the UAV models in Vehicle Mission 3
  + Executes the sequence of timed waypoints for each vehicle,
* Score (need to codify, still)
  + Takes into account number of UAVs needed
  + Uses time to pickup/dropoff based on the largest value in Vehicle Mission 1
  + Vehicle cannot run out of battery
  + Vehicle Team: receives points if their Model scores best
  + to work in one or two areas only