Methodologies for Engineering with Plug-and-Learn Components: Synthesis and Analysis Across Abstraction Layers

John C. Gallagher
john.gallagher@wright.edu
Wright State University, Dayton, OH

Garrison W. Greenwood
greenwd@pdx.edu
Portland State University, Portland, OR

Eric T. Matson
ematson@purdue.edu
Purdue University, West Lafayette, IN

1 Introduction

Cyber-Physical Systems (CPS) that contain self-modifying *smart components* can improve and self-repair, but sometimes at the cost of impeding model-based Verification and Validation (V&V). In this work, we focus on maintaining short and long range V&V capability in a system containing self-adaptive smart components.

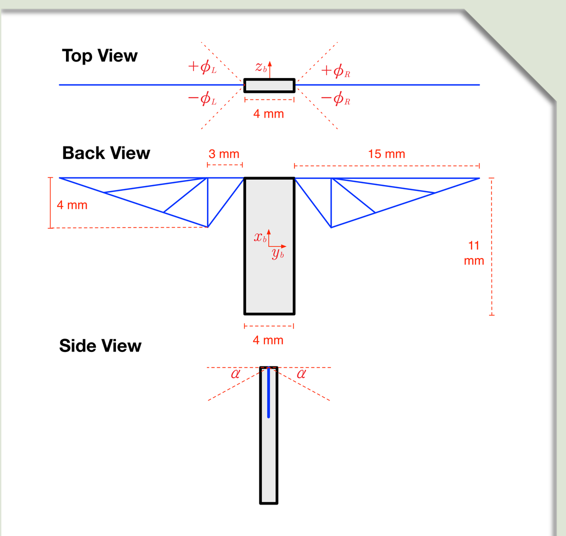

Figure One: A schematic FW-MAV based on the Harvard RoboFly. Our simulation work is based on a full 3D pendulum stable model of this vehicle. In the model, it is presumed that wing gaits and wing flapping frequencies are independently controllable.

In this work, we focus on smart component based in-flight control adaptation of damaged Flapping-Wing Micro Air Vehicles (FW-MAV). Each of our three partner institutions is making a related, but distinct, attack on the problem of encapsulating adaptation into "plug-and-learn" modules and using them to *adapt flight control in a way that enables, rather than destroys, V&V capability*. Each project partner institution is, additionally, focusing on a different level of abstraction in the system's control abstraction hierarchy.

2 Layers of Flight Control Adaptation and V&V

All partner sites use either or both of an aero-static, pendulum-stable, FW-MAV model based on the Harvard RoboFly (Figure One) or a physical flapping-wing device that is floated on water or an air cushion to emulate fine maneuvering at a set hover altitude (Figure Two). A conceptual control model, based on work at AFRL is given in Figure Three. In this model, a high-level path planner (dark orange element in Figure 3) decides *where* the vehicle should be *relative to its current position* and produces desired altitude (body **x** axis, see Figure 1), forward (body **z** axis, see Figure 1) and roll angle (angle around body **x** axis, see Figure 1). Each of these values is communicated to one of the three independent proportional differential axis controller that compute desired body **x** and body **z** translational forces and an **x** axis roll torque. Those desired forces and torque are ran through an inverted model of the vehicle to compute shape (wing gait). Those wingbeat shape parameters are ran through an allocator to combine what may be contradictory commands, and the final shape parameters are communicated to hardware wingbeat oscillators (light orange component of Figure 3) to actuate the wings in the desired manner.

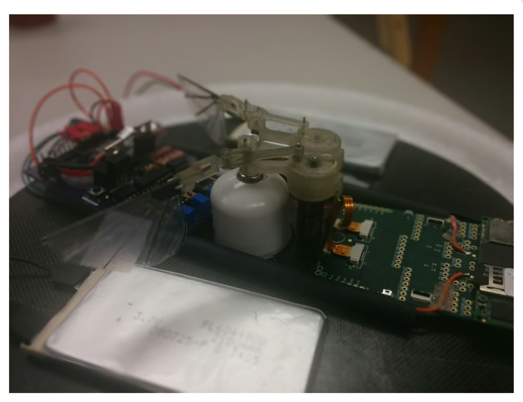

Figure Two: A FW-MAV test vehicle. This vehicle is attached to a puck that is floated on a cushion of air or in a tank of water. It propels itself using wing generated aero forces along the surface of the table. The wing gaits (wingtip trajectory shapes) and flapping frequencies are independently controllable via onboard commutation electronics. The vehicle can receive higher-level control actions via a built-in WIFI interface.

Naturally, however, there are many loci of failure in such a system. Even minor damage to wings and/or other components can render the internal inversion models inaccurate and affect both short term flight accuracy and long-term flight control stability. Full system identification of a newly damaged vehicle could restore correct models, but is not likely practical to accomplish in flight. In our method, we use adaptive oscillators that learn new wing gaits that restore precise maneuvering after wing damage. We also, during local adaptation, extract damage models that can be used to re-enable longer term V&V.

3 Control and Adaptation

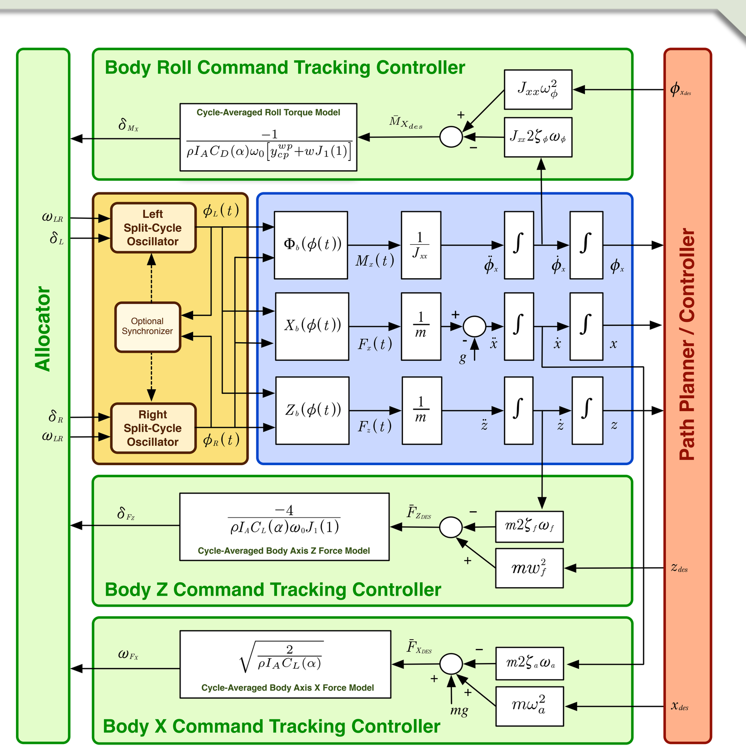

Figure Three: A conceptual control scheme for a pendulum-stable FW-MAV. Force and torque models inside each tracking controller would have their physical parameters tuned to the requirements of the specific vehicle being controlled.

After the wings are damaged, the vehicle models in the PD controllers (green boxes) become inaccurate. This results in an immediate loss of control precision and, potentially, long term controller safety issues (the vehicle flies outside of acceptable boundaries around a desired trajectory). A more traditional approach might attempt system identification to restore the accuracy of the control inversion models. This is unlikely practical during normal system operation. Instead we used *smart component* oscillators that adjusted the base wing gait patterns to restore accurate flight (Figure 4). This method has been demonstrated effective in simulation and is less computationally intensive than *in situ* system identification. Unfortunately, leaving system models unadapted leaves us unable to conduct long term V&V to ascertain if the new wing gaits are safe in the long term.

4 Evolutionary Model Consistency Checking

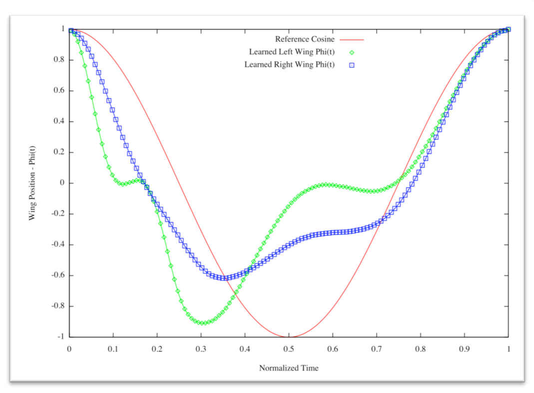

Figure Four: EMCC learned wing beat gaits that correct for damaged wings and, when run on the vehicle, provide diagnoses of the nature of the faults.

Evolutionary Model Consistency Checking (EMCC). EMCC addresses the above concern by modifying the learning algorithms inside the *plug-and-learn* adaptive oscillators so that in addition to finding locally determined wing gaits that restore nominal global behavior, they also diagnose the nature of the wing faults, in terms of loss of drag and lift forces without incurring any significant cost in learning times. The core operational idea is to construct meta-heuristic objective functions that guide search toward local wing gaits that restore correct flight behavior, create multiple wing gait pairs that enable solution of a system of equations that produce cycle-averaged estimates of the losses in left and right wing drag and lift force production. Equations (1) through (4) in the next column are the solution to a set of equations that allow for accurate estimates of wing drag force deficits that are obtained by learning two sets of wing gait pairs that enable the vehicle to fly fixed altitude straight path with a forward pitch. Equations can be similarly derived for lift force faults or potentially other types of damage. In our work, we were able over tens of thousands of random trials of a simulated vehicle with different randomly generated wing fault deficits use EMCC to determine what those faults were within 1% of their true values. This was achieved even when we were using noise-degraded measurements of altitude that one would expect in real situations. The damage estimates (denoted as D in the equations) can be produced for any model that relates wing force and drag to wing gait and forces produced. We are currently using an aerostatic model, but this can be upgraded without invalidating the method.

$$K_{pre} = \left(mg - \frac{\cos(\theta)\left(D_{LLW} F_L(\vec{\sigma}_{LWB}, \omega_B) + D_{LRW} F_L(\vec{\sigma}_{RWB}, \omega_B)\right)}{\sin(\theta)} \right)$$

$$K_{post} = \left(mg - \frac{\cos(\theta)\left(D_{LLW} F_L(\vec{\sigma}_{LWA}, \omega_A) + D_{LRW} F_L(\vec{\sigma}_{RWA}, \omega_A)\right)}{\sin(\theta)} \right)$$

$$D_{DRW} = \frac{K_{pre} - \dfrac{K_{post} F_D(\vec{\sigma}_{LWB}, \omega_B)}{F_D(\vec{\sigma}_{LWA}, \omega_A)}}{F_D(\vec{\sigma}_{RWB}, \omega_B) - \dfrac{F_D(\vec{\sigma}_{RWA}, \omega_A) F_D(\vec{\sigma}_{LWB}, \omega_B)}{F_D(\vec{\sigma}_{LWA}, \omega_A)}}$$

$$D_{DLW} = \frac{K_{pre} - \dfrac{K_{post} F_D(\vec{\sigma}_{RWB}, \omega_B)}{F_D(\vec{\sigma}_{RWA}, \omega_A)}}{F_D(\vec{\sigma}_{LWB}, \omega_B) - \dfrac{F_D(\vec{\sigma}_{LWA}, \omega_A) F_D(\vec{\sigma}_{RWB}, \omega_B)}{F_D(\vec{\sigma}_{RWA}, \omega_A)}}$$

5 Whole-Vehicle Model Checking

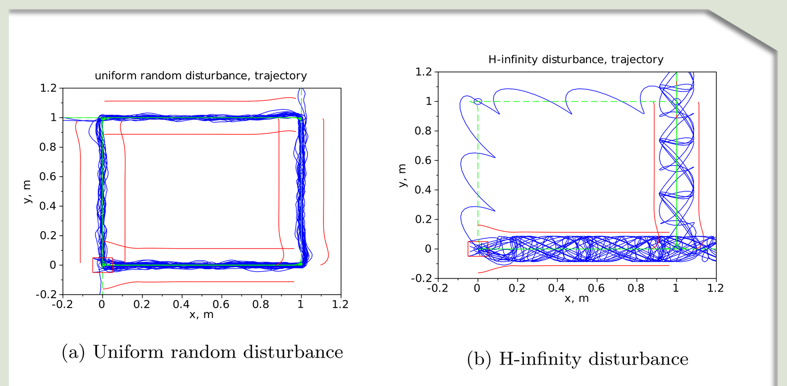

Figure Four: Monte-Carlo simulations of FW-MAVs unable to hold long-term trajectories using naive trajectory trackers with radius based waypoint transition guards.

We have also extended existing algorithms for V&V of Polyhedral Invariant Hybrid Automatons (PIHAs) to account for bounded disturbances in linear hybrid systems using the H-infinity norm. The H-infinity norm of the system can be computed efficiently and only requires updating when the linear system model changes. Coupled with the efficient reachable set computations for linear systems, this makes it possible for us to combine damage estimate updated models of vehicle behavior to determine if the vehicle could maintain trajectories under disturbances with several different control logics and different wing gaits. We are now in the process of integrating EMCC and extended PIHA model checking to restore long-term V&V capability and to further condition oscillator learning to make long-term stable solutions more likely.

6 References

Gallagher, J.C., Sam, M., Boddhu, S., Matson, E., Greenwood, G. (2016). Drag force fault extension to evolutionary model consistency checking for a flapping-wing micro air vehicle. in the 2016 IEEE Congress on Evolutionary Computation.

Gallagher, J.C., Matson, E., and Greenwood, G. (2014). Improvements to evolutionary model consistency checking for a flapping-wing micro air vehicle. in the 2014 IEEE International Conference on Evolvable Systems (ICES) (ICES2014)

Goppert, J., Gallagher, J.C., Hwang, I., and Matson, E., and Greenwood, G. (2013). Model checking of a flapping-wing micro-air-vehicle trajectory tracking controller subject to disturbances. in The 2nd International Conference on Robot Intelligence Technology and Applications (RiTA 2013). Dec 18-20, Denver, CO (*Best in Conference Paper Award)*

Gallagher, J.C. and Oppenheimer, M. (2012). An Improved Evolvable Oscillator and Basis Function Set for Control of an Insect-Scale Flapping-Wing Micro Air Vehicle. Journal of Computer Science and Technology, Vol. 27, No. 5, pp. 966-978. Springer.

Gallagher, J.C., Doman, D., Oppenheimer, M. (2012). The technology of the gaps: an evolvable hardware synthesized oscillator for the control of a flapping-wing micro air vehicle. *IEEE Transactions on Evolutionary Computation*. Vol 16, No. 6.

This material is based upon work supported by the National Science Foundation under grant numbers CNS-1239196, CNS-1239171, and CNS-1239229 Opinions expressed in this poster are those of the authors and do not necessarily reflect those of the National Science Foundation.