Integrated Rapid Development for Unmanned Systems

Adam S. Watkins, Jason A. Stipes, Richard L. Hooks, Jonathan C. Castelli, John C. Humphreys, and Adam W. Cushman

ABSTRACT

Unmanned systems are transforming how the military performs its missions. Many existing program-of-record platforms were designed as multimission general-purpose systems—which can drive up costs and limit aspects of performance. High vehicle cost, slow payload integration cycles, and limitations on vehicle dynamics preclude the use of these platforms for many missions. Advances in 3-D printing and COTS hardware—along with open-source software—offer a solution to deliver cost-effective systems that meet performance requirements through an integrated rapid development process. This process uses the entire vehicle design trade space (cost, range, payload, endurance, autonomy, etc.) to develop customizable systems for specific missions. Current commercial systems offer single static point solutions ill equipped to perform many challenging missions. These advances have the potential to significantly open the aperture of what is possible with unmanned systems—and do it in a way that rapidly turns operator requirements into deployed capabilities.

INTRODUCTION

Limitations of Current Unmanned Systems

Currently fielded program-of-record unmanned systems are designed to perform a variety of missions and carry payloads designed to fit only the target platform. While many missions are well served by the current model, this approach has several limitations that inhibit the use of unmanned systems for numerous mission sets. Typically, program-of-record unmanned systems are teleoperated, possessing no real autonomy beyond following planned waypoints. They require at least one operator per vehicle, and in many cases an entire support team is necessary to use the system effectively. Permitted flight envelopes are very conservative, and operation with aggressive maneuvers, in dangerous locations, or in proximity to the ground is avoided to reduce risk to the unmanned asset. One might assume that unmanned systems are routinely used in high-risk missions because a pilot's life is not at risk, but in reality, operators of program-of-record unmanned systems are very averse to risk because of the expense of the asset and the adverse impact on perceived system reliability that may result from the loss of a vehicle.

A plethora of missions could be accomplished by low-cost, essentially disposable, unmanned systems with autonomous capability, but the current paradigm of unmanned system development and use does not support these kinds of systems. This article presents a new set of capabilities and a development philosophy that reinvents the creation and use of unmanned systems. By rapidly customizing unmanned systems around a specific mission or payload at low cost, a new range of mission sets becomes possible. In addition, marinization of unmanned aerial systems (UASs) enables their operation in the ocean and in littoral areas, increasing the breadth of mission scenarios that can be supported. With large numbers of low-cost, highly capable unmanned vehicles, an asymmetric effect that may be used as a component of the Third Offset Strategy is within reach.[1] The following section provides an overview of the enabling technologies and concepts used to reinvent unmanned system development.

Reinventing Unmanned Systems

The last 10 years have seen the emergence of several technologies that have caused an explosion in the real-world capabilities of unmanned systems, enabling the rapid customization of low-cost unmanned vehicles with complex, fully autonomous behaviors. Specifically, the following list outlines the pillars of the integrated rapid development process for unmanned systems, as presented in this article:

- Using additive manufacturing (also known as 3-D printing) including a novel soluble tooling process
- Leveraging COTS technology, including computing, flight hardware, autopilots, software, sensors, communications, structures, and coatings
- Understanding and automating the performance trade space of COTS technology
- Modifying open-source technologies (e.g., autopilot control software) to rapidly enhance standard capabilities
- Developing algorithms for real-time sensor processing that leverage advancements in low-power, highly parallelized computing hardware targeted for mobile devices
- Developing advanced, robust, deployable autonomous behaviors
- Conducting modeling and simulation (M&S) for rapid design and validation of autonomy and control

The following sections delve into more detail on each of these concepts, describing how capabilities now possible in each of these areas enable rapid fielding of robust customized autonomous unmanned systems.

INTEGRATED RAPID DEVELOPMENT OF UNMANNED SYSTEMS

Additive Manufacturing for Rapid Prototyping

Additive manufacturing, often referred to as 3-D printing, is a collection of technologies that has revolutionized the way engineers look at fabrication of mechanical systems. It has been said that “complexity is free” when using additive manufacturing techniques. Although that may not be entirely true, the spirit behind the statement is accurate in that additive manufacturing provides the ability to make very complicated parts very quickly and at low cost. In many cases, geometries that would not be possible using other subtractive fabrication techniques—or would require multistep fabrication processes—may be produced in a single-step additive manufacturing process. The ability to rapidly fabricate components by using both thermoplastics and metal allows for functional, fieldable parts to be produced quickly, diminishing the reliance on economies of scale for production.

In an extension of additive manufacturing, the Johns Hopkins University Applied Physics Laboratory (APL) has developed a means to construct molded and composite structures quickly and at low cost by using a novel soluble tooling process. This process uses additive manufacturing to create a form made of soluble plastic that can be wrapped in carbon fiber, fiberglass, or other composite fabric and then dissolved out to leave behind a lightweight hollow structure formed to the exact dimensions of the soluble tooling. The form may be made completely of soluble plastic or may contain a combination of soluble and nonsoluble structures. After dissolution, the nonsoluble structures remain, accurately positioned within the composite structure. Figure 1 shows the stages of this process for the fabrication of a small unmanned aircraft system designed to be submersible.

In addition to enabling the fabrication of complex shapes at low cost, the use of additive manufacturing techniques enables a very short development cycle for engineering new platforms with novel capabilities. In one case, using this process, engineers were able to model, fabricate, assemble, and fly a completely new UAS in less time than it takes to order and assemble a COTS system. With shortened development cycles, the development of more complicated flight platforms, such as ones that are submersible or able to fly both as a fixed-wing airplane and a multirotor helicopter, can be shortened from months to weeks. Case studies showing real-world results of this process are presented for platforms that were realized using additive manufacturing techniques, shortened development cycles, and the novel soluble tooling process.

Leveraging COTS Technology

The use of low-cost COTS components enables rapid customization of unmanned systems for specific mis-

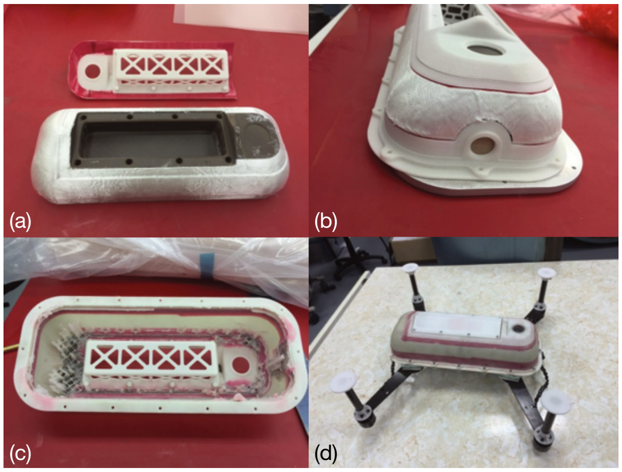


Figure 1. Example of APL's novel additive manufacturing process using soluble tooling depicting (a) the soluble tool wrapped in fiberglass, (b) frames added to the tool, (c) the structure after tool dissolution, and (d) the assembled vehicle frame.

sions. The current boom in the consumer drone market has had the effect of drastically reducing the cost of both small UASs and their components, including motors, batteries, speed controllers, propellers, sensors, cameras, gimbals, autopilots, radios, computing hardware, materials, and coatings. Companies that produce low-cost drones have helped reduce the cost of ready-to-fly UASs by an order of magnitude from where it was only a few years ago. The growing marketplace has resulted in continual improvements in capability and lowered cost of key technologies. APL has leveraged this rapid advancement of COTS hardware to create highly capable autonomous platforms at low cost.

In recent years, proprietary autopilots have been supplanted by equally capable open-source autopilots that are available at a fraction of the cost. Open-source autopilots currently on the market are capable of aircraft stabilization, Global Positioning System (GPS) waypoint navigation, and autonomous execution of preprogrammed missions. Such systems are continually integrated with the latest sensors, including laser altimeters and imagers. One of the primary drivers for the reduction in cost of COTS autopilots derives from the smartphone industry. These autopilots use the same gyroscopes and accelerometers found in today's smartphones, leveraging the performance advances and economies of scale derived from mass production. In addition to highly extensible COTS autopilots, a myriad of COTS flight-related hardware can be leveraged to quickly customize the design of a platform to optimize performance for a given mission's requirements. By understanding the trade space of these components and their impact on flight time, speed, payload capacity, acoustic signature, and other parameters, a wide array of system capabilities may be realized with COTS hardware. More details on understanding this complex trade space appear later in this article.

One extremely important COTS technology leveraged in APL's integrated approach is high-performance computing solutions in packages with low size, weight, and power. Efficiently processing sensor data in real time, running complex control algorithms and autonomy, and managing communications and telemetry are all essential tasks for computing systems placed on board small unmanned vehicles. Weight and power are critical in these applications as both have a direct impact on flight endurance. The recent emergence of computing systems leveraging general-purpose graphical processing units (GPGPUs) and their integration into efficient electronics packages has revolutionized the ability to embed complex real-time image-processing and autonomy algorithms into small unmanned systems. A COTS GPGPU board enables real-time video processing and autonomy while occupying a footprint of only 2 × 2 in. and consuming less than 10 W of power.

COTS industrial coatings are also leveraged when necessary. One example of this an application that required UAS motors to operate when submerged in salt water. Conformal polymer coatings were applied to the motors to improve their corrosion resistance. An untreated motor and a treated motor were placed in a saltwater bath and run every day for 2 months. After 2 months, the motors, shown in Fig. 2, demonstrate the effectiveness of the coating, with the untreated motor exhibiting substantial corrosion.

Flight Performance Trade Space

The widespread availability of a variety of motors, propellers, speed controllers, batteries, and sensors provides for a large trade space of possible UAS performance parameters. Selecting the appropriate components is critical to maximizing system performance. Trades between size, power, controllability, and maximum lift directly impact endurance, range, speed, maximum payload, and aircraft detectability. Understanding this trade space and the impact that different COTS components have on system performance is essential to optimize solutions for

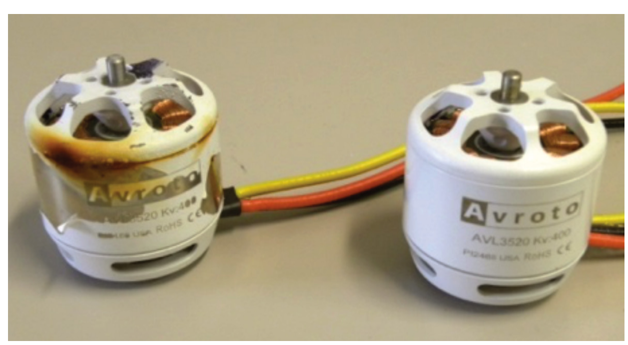


Figure 2. Motor treated with corrosion-resistant coatings (right) and control (left, untreated) after both were immersed in salt water for 8 weeks.

a particular mission set. To this end, maintenance of a database of COTS hardware and performance data is a critical component to our integrated approach to rapid unmanned system development. Understanding and cataloging the available COTS options is an ongoing process as new components reach the market on a daily basis.

The need to understand available COTS hardware extends to payloads that may be used on unmanned systems. As an example, low-cost camera options are widely available in the marketplace, stimulated by demand related to their use in cell phones and standalone sports cameras. Understanding system specifications for resolution, field of view, and overall image quality is important for selecting the appropriate payload.

Figure 3. Pixhawk commercially available open-source autopilot. (Image from https://pixhawk.org/modules/pixhawk.)

Open-Source Autopilot Customization

The ArduPilot open-source autopilot firmware provides a reliable low-cost autopilot for unmanned vehicles. The continually evolving codebase is freely available online, and a community of developers update it with additional capabilities daily. The firmware provides autonomous flight control for helicopters, multirotor aircraft, fixed-wing aircraft, and ground vehicles. Several open-source ground control stations that communicate with the ArduPilot firmware are also freely available. This software is available under a GNU General Public License, which allows for completely free use and modification of the codebase.

The autonomous capabilities of an unmanned vehicle can be augmented by building custom changes to the already powerful ArduPilot codebase. The ArduPilot source code provides the ability to customize navigation behaviors and control parameters that dictate flight characteristics. Modifications to the autopilot can be simple parameter adjustments, modifications to existing control modes, or completely new behaviors. The low-level hardware abstraction of motor and servo actuation and sensor integration is managed by the existing ArduPilot implementation, and higher-level control is easily customizable.

The ArduPilot firmware can be run on a variety of COTS hardware devices; one well-packaged, user-friendly option is the Pixhawk autopilot (see Fig. 3). The Pixhawk has integrated sensors (inertial measurement unit, barometer, and magnetometer), an onboard processor, and multiple external interfaces to add sensors and communication streams. External sensors, such as high-accuracy altimeters and GPS modules, make the Pixhawk hardware extensible.

Autopilot Customization for Precision Optical Landing

Developers at APL have made custom modifications to the ArduPilot firmware to allow external algorithms to control the vehicle for autonomous flight and applications such as GPS-denied optical-based precision landing. For optically guided flight, position error is derived from a downward-facing camera on the aircraft by using optical tracking algorithms. A new control mode, called VisualNav, has been added to the standard Pixhawk control system so that the system can accept image-processing-derived commands for autonomously controlled landing in the absence of GPS. The new control loop uses the existing logic in the Pixhawk's Loiter con-

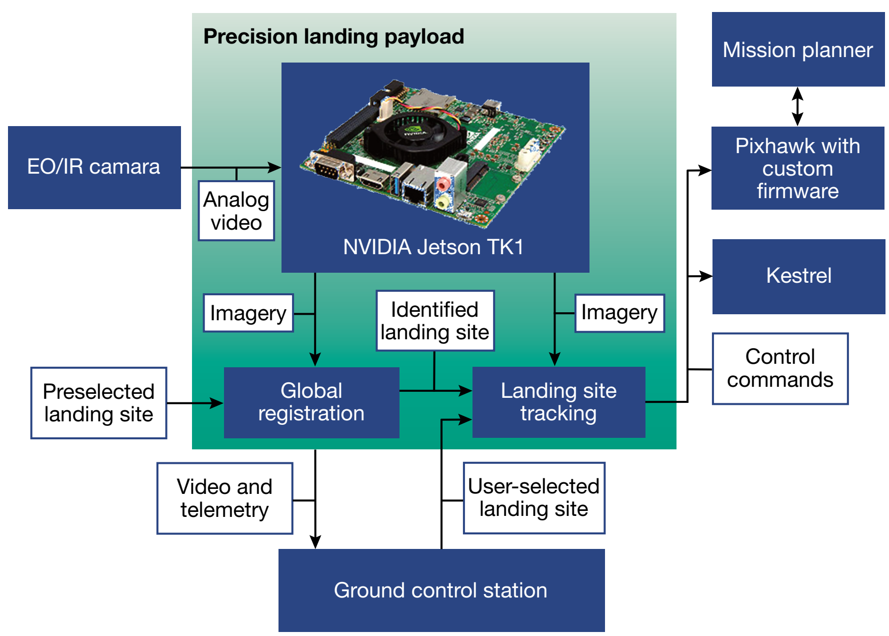


Figure 4. Optical precision landing system.

trol mode but relies on commands provided by the landing site tracking algorithm instead of GPS. The control commands are smoothed over a time window of configurable length to eliminate noise in the image-processing result. Additionally, the VisualNav control mode can operate without GPS, deriving velocity commands from the comparable position commands. The algorithms used to process video to enable optical-guided flight control are described in the next section.

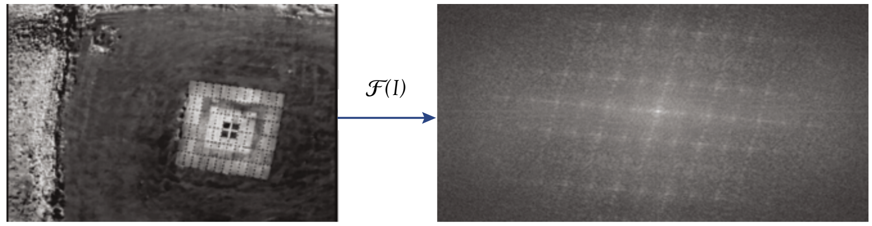


Figure 5. Example image undergoing Fourier transformation.

Precision Optical Landing Algorithm

The team developed the capability to autonomously guide a system's flight and land at a precise location based on input from a downward-looking camera. This capability was rapidly integrated to serve a variety of missions. Similar to the vehicle manufacturing portion of the integrated rapid development process, the onboard sensor processing development process leverages open-source computer vision libraries (e.g., OpenCV) and COTS computing hardware.

The optical landing system comprises two image-processing algorithms, the NVIDIA Jetson TK1 GPGPU and the custom firmware for the Pixhawk autopilot. The first image-processing algorithm performs global registration to locate a landing site by correlating live imagery to a precollected image. The second image-processing algorithm accepts a landing site location in live imagery and continuously tracks the location as the vehicle descends, generating control commands based on vehicle state information. Figure 4 is a diagram of the precision landing system. The Pixhawk autopilot with customized firmware interfaces with the NVIDIA Jetson, which runs the image-processing algorithms, streams video and telemetry to a ground station, and generates control signals to autonomous flight. The desired landing site is either designated in the precollected image or selected in real time using the live video stream. The following sections provide more detail on the global registration and landing site tracking algorithms.

Global Registration

The global registration algorithm coregisters two images, one collected before the mission and one collected by the onboard camera, by using a combination of phase correlation and the log-polar transform (LPT) as described by Zitova and Flusser.[2] The approach was selected because it offers speed, ease of implementation on the GPU, and the ability to use edge images, allowing for the possibility of registering images collected by different sensor modalities (e.g., color and infrared). The selected registration approach falls into the category of Fourier methods for image registration because the registration occurs in the frequency domain by first applying a two-dimensional Fourier transform to the input imagery. Figure 5 shows an example image transformed into the Fourier domain.

Phase correlation exploits the Fourier shift theorem by recognizing that two images that differ by a translation have the same Fourier magnitude, but they differ in phase by an amount proportional to the translation. The phase difference between two images, I_1 and I_2, is equal to the cross-power spectrum of the images as given by:

$$C(I_1,I_2) = \frac{\mathcal{F}(I_1)\mathcal{F}(I_2)^*}{|\mathcal{F}(I_1)\mathcal{F}(I_2)^*|}$$

where $\mathcal{F}$ denotes a Fourier transform and $\mathcal{F}^*$ is the complex conjugate of the transform. Finally, an inverse Fourier transform, $\mathcal{F}^{-1}$, of the cross-power spectrum will give a function that is near zero except for an impulse response at a point defining the translation distance between the two images.

Phase correlation can find the relative translation between two images, but it cannot find the relative rotation and scale difference between images that would commonly be part of the general transformation between a precollected image and a live image collected by an UAS. The LPT, however, can be combined with phase correlation to compute relative rotation and scale. The LPT is a mapping of Cartesian coordinates to log-polar coordinates as shown in Fig. 6.

The same cross-power spectrum computation from phase correlation can be applied to images after a LPT,

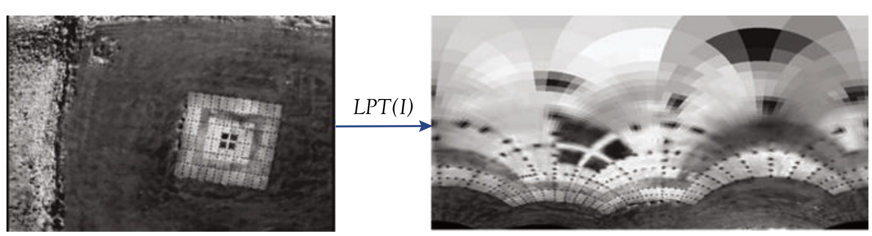


Figure 6. Example image undergoing log-polar transformation.

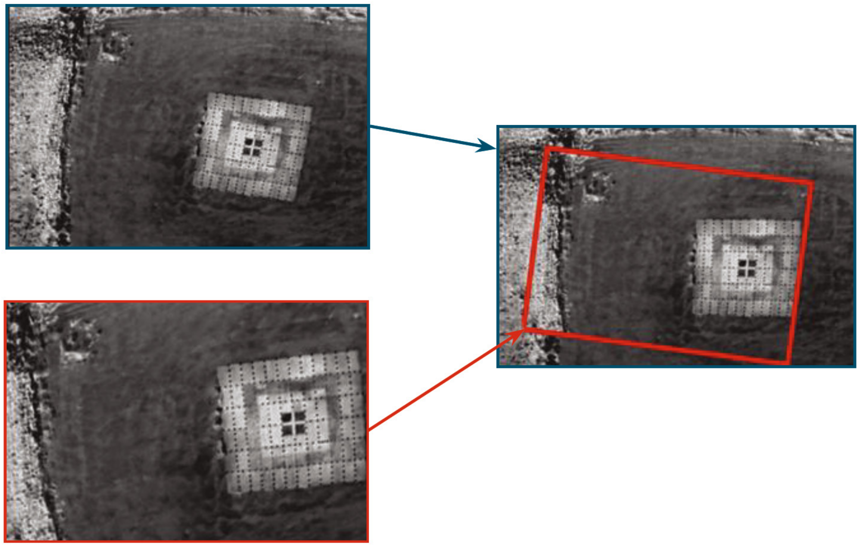


Figure 7. Example global registration result.

onboard computing hardware. The standard CMT algorithm is a keypoint-based approach to deformable object tracking where tracked pixel regions, or keypoints, vote for the object center. The voting process begins with estimating the change in rotation and scale of the object by comparing the locations of tracked keypoints in the current object model to the tracked keypoints in absolute image coordinates. The vote for each tracked keypoint is the scaled and rotated vector from the keypoint to the object center. A hierarchical clustering scheme is then used to prune outliers.

The standard CMT algorithm was altered to enable tracking of a landing site in the context of the entire camera image. Although the algorithm was originally intended to track a single object comprising a small subset of the overall image, the desired behavior for precision landing is to track a single point robustly through large-scale changes during descent and quick changes in camera attitude where the object may intermittently leave the camera's field of view. To achieve this, keypoints are extracted from the entire image instead of a small subregion. This full-scene tracking provides robustness to significant changes in camera attitude because it is likely that a sufficient number of keypoints will remain in the field of view if they are initially uniformly distributed.

at which point the resulting inverse Fourier transform gives an impulse response corresponding to the relative rotation and scale of the two images. The resulting rotation, however, is ambiguous by a factor of π radians. The complete registration algorithm addresses this ambiguity by transforming the original image by the computed rotation and the rotation plus π. Then the relative translation is computed for both rotated images, and the result that produces the largest impulse response peak is selected. Figure 7 shows an example registration result.

To address the difficult problem of multimodal image registration, the algorithm can first compute an edge image using the well-known Canny edge detector. This operator converts multimodal images into common space because gradient information is often invariant across collection modalities.

Once the two images have been registered, the computed transformation is used to find the pixel that identifies the landing size in the live image stream by simply applying the transformation to the pixel selected in the historical imagery.

Landing Site Tracking

The result of the global registration algorithm is a pixel location in the live imagery of the desired landing site. The pixel location can also be selected by a human operator through the ground control station. Once the location has been selected, the landing site tracking algorithm will track the point in the live camera stream throughout the vehicle's descent.

The landing site tracking algorithm is based on the consensus-based matching and tracking (CMT) algorithm presented by Nebehay and Pflugfelder.[3] This algorithm was selected because of its overall robustness with respect to geometric deformations in the scene and ability to be at the frame rate of the camera given the

The algorithm was also customized to provide an adaptive approach to keypoint identification in which the threshold defining what is a keypoint is adaptively updated to maintain a desired number of feature points throughout the tracking process. This approach addresses changes in image characteristics because the overall scene illumination varies with the time of day (for color cameras) or ambient temperature (for infrared). Additionally, image characteristics fluctuate with the large-scale changes inherent in imagery collected by a descending vehicle.

The final significant change to the CMT enables close management of the distribution of tracked keypoints to enforce a uniform distribution of keypoints across the image through careful replacement of redundant keypoints. A real-time result of the algorithm during the descent of a UAS is shown in Fig. 8.

The output of the landing site algorithm is an offset in image space coordinates of the landing site from the center of the image. The pixel offset is converted into an offset in east–north–up coordinates using the platform's state information including the measured height above ground.

Autonomous Vehicle Control

Another key component in the integrated approach to rapidly fielding capable autonomous vehicles is the leveraging of a robust framework for implementation of autonomous multivehicle control algorithms. This framework, called the Autonomy Toolkit (ATK), was developed by APL to provide an architecture for creating autonomous systems that can robustly achieve mission objectives. Although single-vehicle systems can be developed with ATK, the primary focus has been on swarming autonomy in which numerous unmanned agents coordinate and/or cooperate to complete complex tasks. The advantages of swarming autonomy include system robustness with respect to vehicle attrition, scalability to large numbers of agents, low computational requirements, lack of reliance on high-bandwidth communications, and the ability to demonstrate complex behaviors through simple interactions with other agents. Over the past decade, numerous autonomous systems have been developed and demonstrated using ATK.[4–6]

The core components of ATK that enable swarming autonomy include a behavior-based robotic control architecture, a communications management system for limiting bandwidth, a knowledge (i.e., belief) management system for constructing a distributed world model, and a hardware abstraction layer for ensuring autonomous solutions are hardware agnostic. The software components of ATK are assembled into an agent, shown in Fig. 9, which can ingest sensor information, reason about the external environment, communicate with other swarm agents, and compute control commands.

The general framework provided by ATK enables the rapid creation of autonomous swarms. The behavior-based robotic architecture is modeled as a finite-state machine in which each node can contain multiple parallel behaviors whose output is combined to provide a single control command. Typical behaviors include area search, track, agent deconfliction, obstacle avoidance, and communications chain, which are then combined to produce aggregate behaviors (e.g., search and avoid). The aggregate behaviors can then be interconnected in the finite-state machine to create a system capable of executing complex missions in which aggregate behaviors are selected according to various external stimuli typically associated with mission objectives.

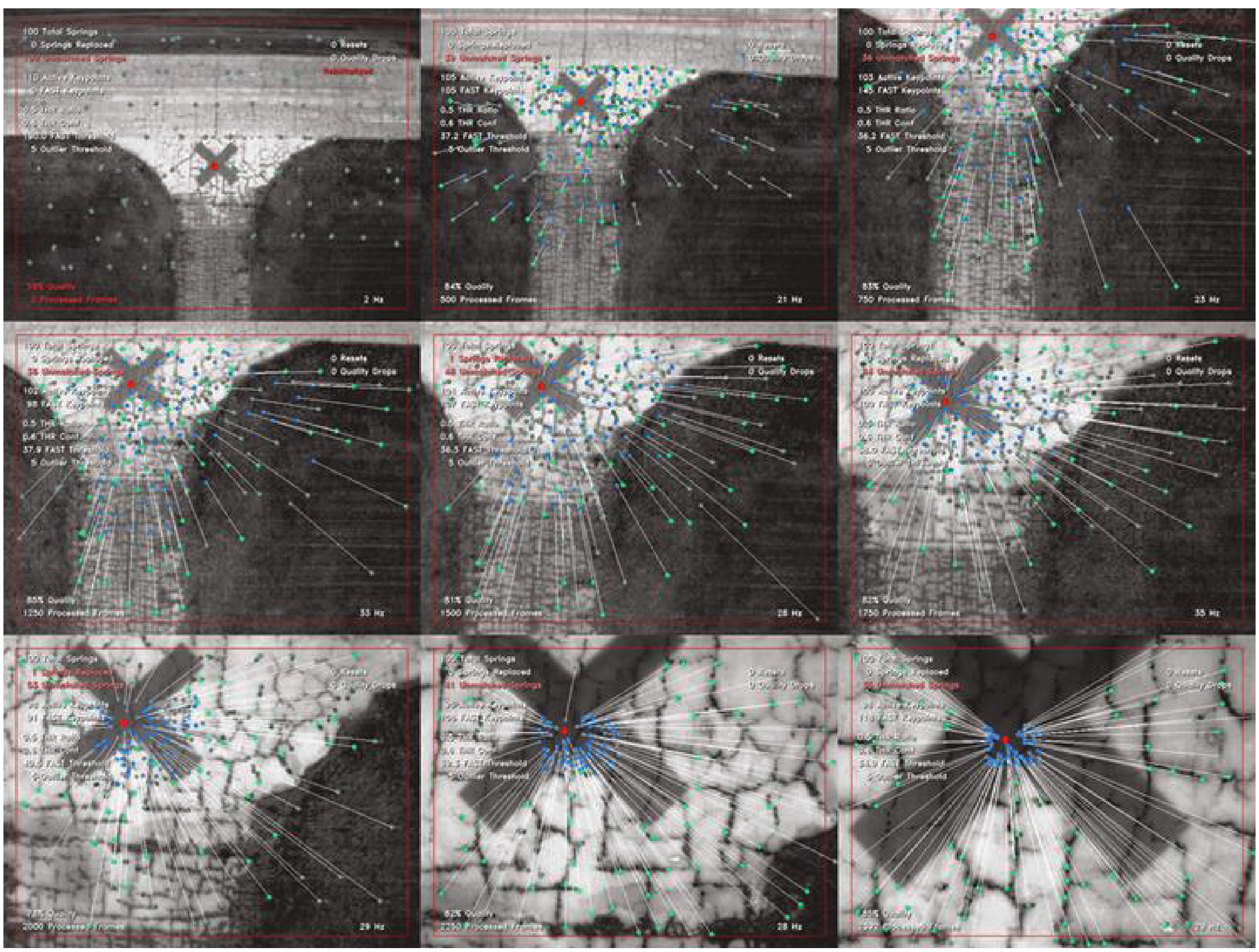


Figure 8. Example image sequence of the landing site tracking algorithm.

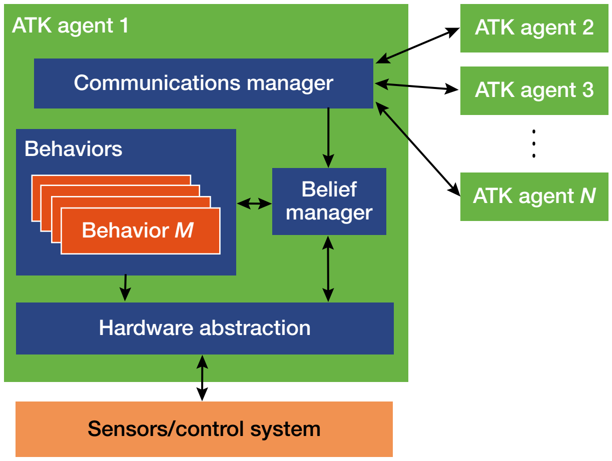


Figure 9. ATK agent architecture.

Modeling and Simulation

While developing a new unmanned system as part of an integrated development process, understanding and tuning the performance and stability of the autonomy and autopilot control algorithms is critical. M&S techniques allow refinement of the control strategy before hardware testing is conducted. Rapid prototyping of unmanned systems requires the ability to develop and tune vehicle-specific autopilots very quickly. M&S can be used to help speed this development process and is particularly important when developing complicated nonlinear control systems. An example of the use of M&S is shown in Fig. 10, which illustrates how optimization techniques have been used to tune and gain-schedule an autopilot. In Fig. 10a, the results of several optimization runs are shown for a step response for a UAS control loop. This optimization approach balances time domain (e.g., rise time, settling time, overshoot) and frequency domain (e.g., gain margin, phase margin, vector margin) criteria, resulting in reasonably tuned autopilots in a very short time line.

M&S can also be used to help develop and test the higher-level autonomy algorithms. In addition to simulating and visualizing the vehicle flight characteristics and autopilot performance, the modeling of the environment and mission parameters can be used to better develop the autonomy algorithms and to conduct Monte Carlo simulations to validate the performance of the system before test flights. This enables the developer to simulate numerous scenarios when it is not possible or practical to execute using the system during actual flight tests. This provides a higher level of confidence in system autonomy and reduces the potential for failures in the field during development. Figure 10b shows a screenshot from a full 3-D rendered simulation of a swarm of UASs autonomously accomplishing a mission in a realistic environment with modeled environmental effects such as wind. The fidelity shown in this model and simulation can be leveraged to assist in the integrated rapid development of unmanned systems and provides a useful means to communicate and present the capabilities of the autonomous behaviors developed for a given mission.

CASE STUDIES

The following case studies illustrate examples in which concepts from APL's integrated rapid development process were applied to develop and fabricate unmanned aerial vehicles that are customized to achieve unique capabilities. These case studies highlight the novel applications and short development times enabled by our unique development process.

Four-Day UAS Development Cycle

The first case study involved evaluation of a customized UAS airframe concept designed specifically to reduce the weight of a typical multi-rotor airframe when contrasted with existing construction techniques. For

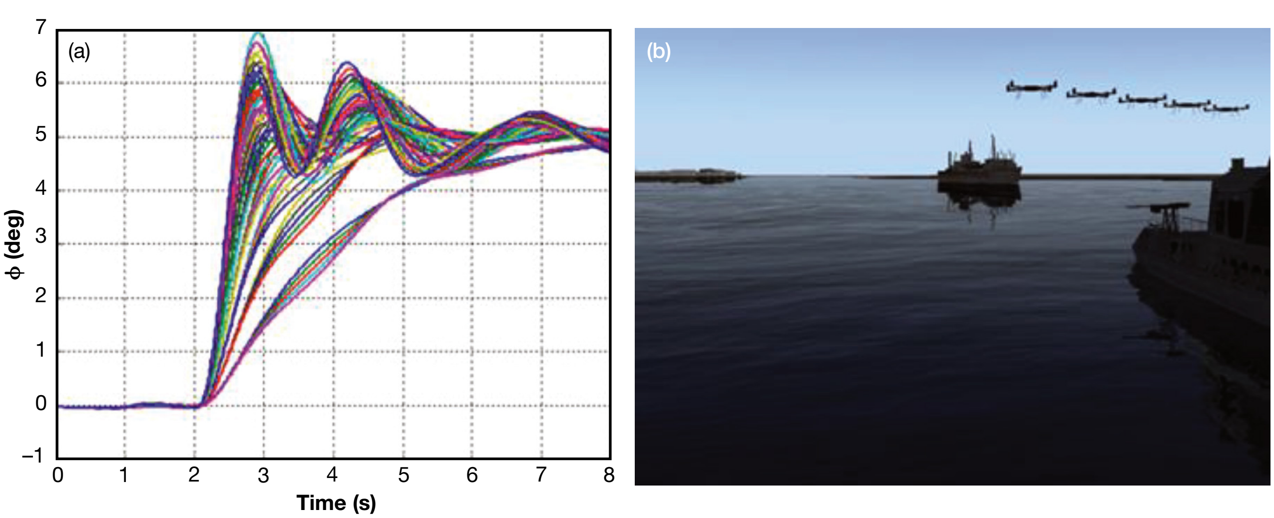


Figure 10. Simulation output used to optimize control gains (a) and 3-D visualization of simulation (b).

Figure 11. Example airframe constructed using novel additive manufacturing techniques.

example, COTS airframes are usually made of aluminum or composite materials. Three toroidal multi-rotor airframes were designed and built to take advantage of the natural structure of a circular frame with a tubular cross section. The final design is shown in Fig. 11. The airframes themselves were constructed by leveraging the soluble tooling process. The result of the experiment was an extremely strong structure with a significant weight savings compared to conventional designs. The smallest airframe design saved 18% of weight, the medium airframe saved 40%, and the large airframe saved 50% when compared to COTS copters of comparable size and capability. Perhaps the most interesting observation from this experiment was the speed with which custom designs could be realized. The smallest airframe was designed using computer-aided design, printed, fabricated, assembled, and flown within 4 days.

CRACUNS

The second case study is the Corrosion Resistant Aerial Covert Unmanned Nautical System (CRACUNS), a proof-of-concept system developed to demonstrate the ability to create an amphibious UAS by using the integrated rapid development approach. The system was designed to be submersible to a depth of hundreds of feet and able to withstand the corrosive maritime environment. The prototype system is shown in Fig. 12.

Development of CRACUNS expanded the possible operating environments of rapidly developed unmanned systems into the littoral and underwater areas. Additionally, the system demonstrated the ability to create pressure vessels and sealable O-ring surfaces by using additive manufacturing techniques. The rapid development approach APL pioneered compressed the development time line for this platform to 4 months.

The lessons learned in the CRACUNS system development were leveraged to develop Mini-CRACUNS, which can be folded to fit inside an unmanned underwater vehicle. Like the CRACUNS, it is a pressure-sealed UAS capable of being submersed to depths up to 50 ft and released to float to the surface where it can autonomously take off to perform its mission. The purpose of the design was to extend the underwater release concept to include deployment from an unmanned underwater vehicle. Specifically, it was designed to fit within a 12-in.-diameter and 14-in.-long payload cylinder. From concept through design and testing, the Mini-CRACUNS prototype, shown in Fig. 13, was produced and flight-tested in a 2-month period.

Unlike CRACUNS, which was constructed primarily from additive manufactured parts, Mini-CRACUNS was built from a combination of machined and additive manufactured parts. The top is a machined piece of fiberglass, and the bottom hull is made from printed parts and a fiberglass wrap using the soluble tooling

Figure 12. CRACUNS submersible UAS.

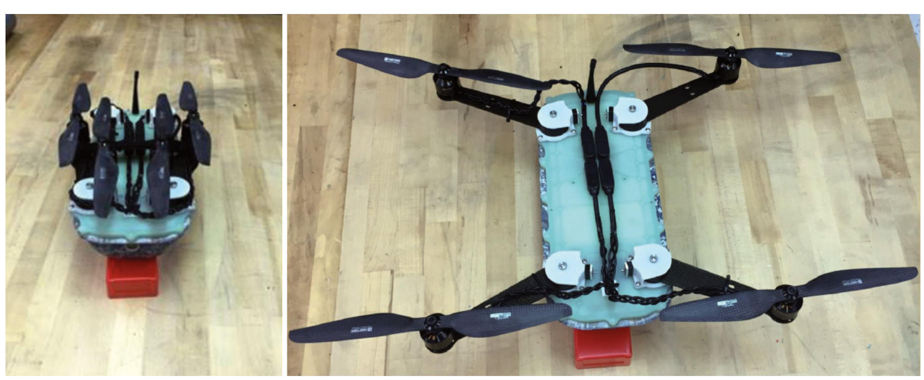


Figure 13. Folding CRACUNS platform model showing folding arms for stowage inside an unmanned underwater vehicle.

method described earlier. Fiberglass was selected for its strength and transparency to radio frequencies allowing the use of internal antennas. Similarly, the arm mechanisms were built from a combination of machined and additive manufactured parts. The system weight is approximately 6 lb, and it measures 21.3 in. diagonally from motor to motor and is approximately 14 × 12 × 6 in. when folded. This platform is an excellent example of how highly capable customized systems can be implemented rapidly by using the new development paradigm presented in this article.

CONCLUSION

The examples in this article show that recent changes in manufacturing technologies, algorithms, computing components, sensor systems, UAS hardware, and mission requirements have coalesced to change how small unmanned systems are developed and how dynamic operational needs are met. An asymmetric effect is achievable through rapid fielding of large numbers of highly capable, autonomous, low-cost unmanned systems in ways not previously possible—putting within reach an asymmetric effect that can be used as a component of the Third Offset Strategy.[1] An integrated rapid-development process that takes into account the whole trade space (cost, range, payload, endurance, autonomy, etc.) and allows the rapid fielding of low-cost, extremely customizable platforms, whereas current commercial systems merely offer static points in the same trade space. This new development paradigm has the potential to significantly open the aperture of what is possible with unmanned systems—and to do it in a way that rapidly turns operator requirements to deployed capabilities.

REFERENCES

[1]U.S. Department of Defense, *The Third U.S. Offset Strategy and its Implications for Partners and Allies*, Deputy Secretary of Defense Speech, As Delivered by Deputy Secretary of Defense Bob Work, Willard Hotel, Washington, DC, 28 Jan 2015, http://www.defense.gov/News/Speeches/Speech-View/Article/606641/the-third-us-offset-strategy-and-its-implications-for-partners-and-allies.

[2]Zitova, B., and Flusser, J., "Image Registration Methods: A Survey," *Image Vision Comput.* **21**(11), 977–1000 (2003).

[3]Nebehay, G., and Pflugfelder, R., "Consensus-Based Matching and Tracking of Keypoints for Object Tracking," in *Proc. 2014 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Steamboat Springs, CO, pp. 862–869 (2014).

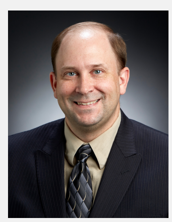
[4]Scheidt, D., Stipes, J., and Neighhoff, T., "Cooperating Unmanned Vehicles," in *Proc. IEEE International Conf. on Networking Sensing and Control*, Tucson, AZ, pp. 326–331 (2005).

[5]Stipes, J., Hawthorne, R., Scheidt, D., and Pacifico, D., "Cooperative Localization and Mapping," in *Proc. 2006 IEEE International Conf. on Networking, Sensing and Control*, Ft. Lauderdale, FL, pp. 596–601 (2006).

[6]Funk, K. B., Castelli, J. C., Watkins, A. S., McCubbin, C. B., Marshall, S. J., et al., "JEFX 10 Demonstration of Cooperative Hunter Killer UAS and Upstream Data Fusion," in *Proc. SPIE Unmanned Systems Technology XIII*, Orlando, FL, paper 80450E (2011).

Adam S. Watkins, Force Projection Sector, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Adam Watkins is the Section Supervisor of the Intelligent Sensor Technologies Section in the Ocean Systems and Engineering Group in APL's Force Projection Sector. He provided leadership in the conceptual development of the Range Adversarial Planning Tool (RAPT). Adam holds a Ph.D. in aerospace engineering from the University of Florida. His research is focused on autonomous system control and image processing. His e-mail address is adam.watkins@jhuapl.edu.



Jason A. Stipes, Space Exploration Sector, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

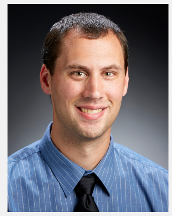
Jason Stipes is a member of the Principal Professional Staff in APL's Space Exploration Sector, where he specializes in embedded systems and unmanned system autonomy. He received a B.S. in mechanical engineering from Purdue University in 1995, an M.S. in electrical engineering from Johns Hopkins University in 2003, and an M.B.A. from Loyola University in 2005. As Principal Investigator for EMAPS (Enhanced Mapping and Positioning System), he developed a backpack system capable of generating interactive annotated maps of GPS-denied areas that display sensor readings and omnidirectional imagery in real time. As Lead for the WACS (WMD Aerial Collection System), he developed and demonstrated hardware and software to allow an RQ-7B Shadow unmanned aerial system to autonomously detect, interrogate, and track an explosive plume to sense and collect weapons of mass destruction materials. He was also Project Lead for the Advanced SEAL Delivery System (ASDS) Training Simulator and has developed control software for a neurally controlled prosthetic arm as part of DARPA's Revolutionizing Prosthetics 2009 program. His e-mail address is jason.stipes@jhuapl.edu.

Richard L. Hooks, Force Projection Sector, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Richard Hooks is an aerospace/mechanical engineer and a Senior Professional Staff member in the Ocean Systems and Engineering Group of APL's Force Projection Sector. He received an M.S. in mechanical engineering from Johns Hopkins University and a B.S. in aerospace engineering from the University of Maryland at College Park. He has extensive experience in design and fabrication of hardware for various applications ranging from underwater to spaceflight, and his research interests include hardware development related to unmanned aerial systems, unmanned underwater vehicles, and autonomous systems. His e-mail address is richard.hooks@jhuapl.edu.

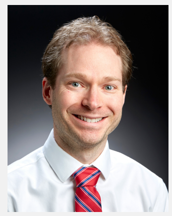
Jonathan C. Castelli, Affiliation unknown

While at APL, Jonathan C. Castelli was the Manager of the Rapid Guidance, Navigation, and Control Prototyping and Air Vehicle Integration Laboratory and an Associate Staff member in the Guidance and Test Section of the Weapon and Targeting Systems Group. He provided engineering and UAS operational expertise to the independent research and development demonstrations and was the lead systems engineer on the JEFX 10 project.



John C. Humphreys, Force Projection Sector, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

John Humphreys is an autonomous systems developer in APL's Force Projection Sector. He studied mechanical engineering at Virginia Tech, with additional focuses in robotics and software development. He has extensive experience developing software for and integrating unmanned systems, including autonomous control algorithms, sensor interfaces, and operator control stations. His e-mail address is john.humphreys@jhuapl.edu.



Adam W. Cushman, Force Projection Sector, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Adam W. Cushman is a member of APL's Senior Professional Staff and is Assistant Group Supervisor of the Special Concepts and Engineering Development Group. He has served as project or technical lead on a wide variety of projects, and his areas of interest are physics-based modeling and simulation; autonomous systems; navigation, guidance, and control; and software development. His e-mail address is adam.cushman@jhuapl.edu.