This issue of the Johns Hopkins APL Technical Digest is devoted to science and technologies that are being pursued across APL in the field of autonomous systems. Such technologies have applications in manned and unmanned space missions as well as special operations, explosive ordnance disposal, and conventional warfare and may be deployed in a variety of means to include air, surface, and subsurface platforms.
Autonomous Systems: Guest Editor’s Introduction
G. Peter Nanos

A Brief History of Early Unmanned Aircraft
John F. Keane and Stephen S. Carr

Unmanned Aircraft System Airspace Integration in the National Airspace Using a Ground-Based Sense and Avoid System
Thomas P. Spriesterbach, Kelly A. Bruns, Lauren I. Baron, and Jason E. Sohike

Recent Enhancements to Mobile Bimanual Robotic Teleoperation with Insight Toward Improving Operator Control
Edward W. Tunstel Jr., Kevin C. Wolfe, Michael D. M. Kutzer, Matthew S. Johannes, Christopher Y. Brown, Kapil D. Katyal, Matthew P. Pena, and Michael J. Zeher

Integration of Advanced Explosive Ordnance Disposal Robotic Systems Within a Modular Open Systems Architecture
Mark A. Hinton, James M. Burck, Kristine R. Collins, Matthew S. Johannes, Edward W. Tunstel Jr., and Michael J. Zeher

Buckybot: Preliminary Control and Mapping Algorithms for a Robot Geometrically Based on a Truncated Icosahedron
Robert C. Grande, Michael D. M. Kutzer, Christopher Y. Brown, and Mehran Armand

Morphable Limbs for Multi-Legged Climbing on Convex Structures
Michael D. M. Kutzer, Edward W. Tunstel Jr., Christopher Y. Brown, Danoosh Vahdat, Jonathan E. Clark, Noah J. Cowan, and Mehran Armand


Autonomous Geolocation of RF Emitters Using Small, Unmanned Platforms
Robert J. Bamberger, Jay G. Moore, Ravi P. Goonasekeram, and David H. Scheidt
In the early 20th century, the key elements that determined success in war at sea were the caliber, range, and accuracy of guns; the speed of ships; and the thickness of armor belts. By mid-century, the key elements had become the speed and maneuverability of aircraft, the capacity and number of aircraft carriers, the quality of electronics, and the introduction of nuclear weapons, which, through deterrence, limited the scale of warfare. At the end of the century, the importance of surveillance systems, space systems, precision weapons, and networks had been established. Had the question been asked about land warfare, the progression would have been very similar, with tremendous advances in warfighting capability based on the most modern technologies.

Looking back on the amazing 20th century, it is easy to focus on the details of modern weapons systems, highlighting the microelectronics, lasers, and aerospace technology details and not concentrating enough on the important ideas that are central to our present and future capabilities. It is only by giving these capabilities the credit they deserve that we understand how military need shapes not only the employment of technology but also the prioritization of resources that lead to its development. As pointed out by Jack Keane and Steve Carr in their excellent chronology of the development of unmanned systems, the need for unmanned sensors and weapons was understood very early in the century, when heavily armored dreadnoughts ruled the seas. In fact, the idea of putting machines, rather than people, at risk for highly dangerous missions was so compelling that it resurfaced at periodic intervals whenever new technology gave promise that this fundamental operational dream could finally be realized. Tomahawk and other modern missile systems find their genesis in the first attempts to build and field aerial torpedoes in World War I. Similarly, the recasting of target drones for reconnaissance in the 1950s was the harbinger of the modern Predators that have been employed so successfully over hostile territory for both reconnaissance and strike. These were by no means rapid developments. They spanned 50 to 100 years and were full of false starts, failed programs, and disappointment. It was the compelling nature of the military need that kept the ideas, if not the specific programs, alive until the maturation of technology allowed success.

This issue of the *Johns Hopkins APL Technical Digest* offers a diverse collection of articles that highlight APL contributions to a number of issues being addressed in the modern world of unmanned systems. With the sparse air traffic of the day, few in World War I could have foreseen that ubiquitous use of unmanned systems, coupled with the need to safeguard civil aviation, would require sense and avoid systems as described by Spriesterbach et al. The wars in Iraq and Afghanistan have intensified the development of unmanned ground vehicles for a number of tasks, but none are as urgent as those that bring improved explosive ordnance disposal (EOD) capability to deployed forces. Two papers explore this area: Tunstel et al. discuss their work on improving the ability of human operators to teleoperate robots in remote hazardous locations. Hinton et al. take on the
more general topic of system integration by discussing the systems architecture of a family of EOD vehicles.

Of a more exploratory nature are papers dealing with unique modes of locomotion: Grande et al. discuss their work on mapping and control algorithms for a Buckybot. Kutzer et al. explore a mechanism that allows an insect-like vehicle to climb, and then the authors evaluate the vehicle's stability. Finally, there are two papers that deal with unmanned systems with the addition of autonomy: McGee et al. report on the successful result of 8 years of effort to develop and test an autonomous lander for exploration of airless bodies such as the moon or asteroids. Bamberger et al. explore the geolocation of RF emitters using small autonomous air platforms.

It is hard not to be impressed by the breadth and depth of these articles and what they say about the complexity of the field of unmanned systems. It takes a laboratory of the breadth and depth of APL to make meaningful contributions either to the underlying technology or to the fielding of these systems. Judging by these articles, APL is doing both. As a final note, APL has a burgeoning group of staff members that have become interested in the contributions that autonomy can make to military systems, and they are making important contributions to the field. It should be noted that autonomous systems are on the list of technology priorities for DoD. It is also hard to believe that autonomy, with its promise of cost savings and enhanced capability, will not stay near the top of DoD's list of research and development priorities for a long time. Being able to operate platforms without trained pilots and with simple interfaces to allow intuitive tasking by the operational user provides huge advances in utility but, more importantly, huge reductions in cost by obviating the need for a large number of pilots and other trained operators. That many at APL are stepping up to the critical challenges in this field says a lot about the Laboratory's ability to go where needed. It also positions APL to take a leadership role in what can become the next important idea that drives technology development.

G. Peter (“Pete”) Nanos Jr. is a retired U.S. Navy Vice Admiral and a 1967 graduate of the U.S. Naval Academy. Dr. Nanos is the Managing Executive of the Force Projection Sector. Since returning to the Laboratory from the Defense Threat Reduction Agency (DTRA), Fort Belvoir, Virginia, where in 4.5 years he founded the billion-dollar Research and Development Enterprise and served as Associate Director for Operations, Dr. Nanos has served as an acting department head in both the National Security Analysis and Global Engagement Departments. Prior to coming to APL and DTRA, he served as the Director of Los Alamos National Laboratory, New Mexico, from 2003 to 2005. His e-mail address is peter.nanos@jhuapl.edu.

The Author

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.
A Brief History of Early Unmanned Aircraft

John F. Keane and Stephen S. Carr

Current developments in unmanned aerial vehicles (UAVs) trace their beginnings to World War I. Efforts during the Interwar Period, World War II, and afterward ultimately led to the development of cruise missiles such as Harpoon and Tomahawk, aerial targets, and the current family of UAVs. UAVs have the ability to transmit to the battlefield commander real-time intelligence, surveillance, and reconnaissance information from hostile areas. They can also act as communication relays, designate targets for neutralization by other assets, or attack the targets themselves with onboard munitions and then loiter while streaming real-time battle damage information back to friendly forces—all without risking the lives of an aircrew. This article provides a historical survey on the early development of select UAVs in the U.S. military and their military applications. The development of cruise missiles and UAVs is intertwined. As the reader will see, many of the technologies experimented with in cruise missiles made their way to UAVs, and vice versa. Although making mention occasionally of cruise missiles, this article will attempt to focus on selected UAV development and employment through the Persian Gulf War.

BACKGROUND

Unmanned air systems trace their modern origins back to the development of aerial torpedoes almost 95 years ago. Efforts continued through the Korean War, during which the military services experimented with missions, sensors, and munitions in attempts to provide strike and reconnaissance services to battlefield commanders. In the 1950s, both the Navy and Air Force bifurcated their efforts to concentrate on cruise missile and unmanned aerial vehicle (UAV) development via separate means. In this article, the term cruise missile
refers to a one-way lethal munition designed to strike specific targets, while UAV refers to a reusable aircraft that has the ability to perform a variety of missions. There are three classes of UAVs: (i) pilotless target aircraft that are used for training purposes (such as target drones); (ii) nonlethal aircraft designed to gather intelligence, surveillance, and reconnaissance (ISR) data; and (iii) unmanned combat air vehicles (UCAVs) that are designed to provide lethal ISR services.

UAVs have been around much longer than most people realize. During World War I (WWI), both the Navy and the Army experimented with aerial torpedoes and flying bombs. Some of the most brilliant minds of the day were called on to develop systems to be used against U-boat bases and to break the stalemate caused by nearly 4 years of trench warfare. Efforts consisted of combining wood and fabric airframes with either gyroscope or propeller revolution counters to carry payloads of almost 200 pounds of explosives a distance of approximately 40 miles. Hostilities ceased before either could be fielded.1 These WWI UAVs highlighted two operational problems: crews had difficulty launching and recovering the UAVs, and they had difficulty stabilizing them during flight. During the Interwar Period, radio and improved aircraft engineering allowed UAV developers to enhance their technologies, but most efforts failed. Despite failures, limited development continued, and after UAVs successfully performed as target drones in naval exercises, efforts were renewed in radio-controlled weapons delivery platforms. World War II (WWII) saw the continued use of target drones for anti-air gunnery practice. Additionally, radio-controlled drones were used by both the Allied and Axis powers as weapons delivery platforms and radio-controlled flying/gliding bombs.

With the start of the Cold War, UAVs began to be used as ISR systems, with limited success as weapons delivery platforms. Development continued throughout the Vietnam War, but interest soon waned once hostilities ceased. The 1991 Gulf War renewed interest in UAVs, and by the time the Balkans Conflict began, military intelligence personnel were regularly incorporating UAV ISR information into their analyses. Currently, UAVs effectively provide users with real-time ISR information. Additionally, if the ISR information can be quickly understood and locations geo-registered, UCAVs can be used to strike time-sensitive targets with air-to-surface weapons.

Like many weapon systems, UAVs thrive when the need is apparent; when there is no need, they fall into disfavor. Numerous obstacles have hindered UAV development. Oftentimes, technologies simply were not mature enough for the UAVs to become operational. Other times, lack of service cooperation led to failure. For example, the U.S. Army Air Corps funded Project Aphrodite (using B-17s as flying bombs) in WWII, while the Navy’s WWII Project Anvil was very similar but used PB4Ys (the Navy’s designation for the B-24). If the services had coordinated efforts, perhaps the overall effort would have been successful. Additionally, competing weapons systems made it difficult for UAVs to get funding. And of course, it was sometimes difficult to sell pilotless aircraft to senior service leaders, who were often pilots.

Many obstacles still stand in the way of continued UAV development. These include mostly non-technical issues, such as lack of service enthusiasm, overall cost effectiveness, and competition with other weapons systems (e.g., manned aircraft, missiles, or space-based assets).

**WWI: Efforts in the Development of the Aerial Torpedo and Kettering Bug**

In 1911, just 8 years after the advent of manned flight, Elmer Sperry, inventor of the gyroscope, became intrigued with the application of radio control to aircraft. Sperry succeeded in obtaining Navy financial support and assistance and, between 31 August and 4 October 1913, oversaw 58 flight tests conducted by Lieutenant P. N. L. Bellinger at Hammondsport, New York, in which the application of the gyroscope to stabilized flight proved successful.2 In 1915, Sperry and Dr. Peter Cooper Hewitt (best known for his work in radio and contributions to the development of the vacuum tube) were appointed members to the Aeronautical Committee of the Naval Consulting Board, established by Secretary of the Navy Josephus Daniels on 7 October 1915 and led by Thomas A. Edison to advise Daniels on scientific and technical matters.3–5

By this time, Europe was embroiled in war, and the utility of unmanned aircraft was becoming apparent. Conditions on the European battlefields were ideal for such a system: enemy anti-aircraft weapons were heavily concentrated, Germany had air superiority in certain sectors, and there was an extremely static battlefield situation over 470 miles of front. Heavy British air losses led to a research program at the Ordnance College of Woolwich, United Kingdom, in remotely controlled pilotless aircraft designed to glide into the target and explode on impact. A parallel program was begun by the Royal Aircraft Factory that included aircraft manufacturers such as the Sopwith Aviation Company and de Havilland. None saw action during the war.6

By 1916, Carl Norden (developer of the famed Norden bombsight of WWII) joined the Sperry/Hewitt team and developed the concept of an aerial torpedo. After America’s entry into WWI on 6 April 1917, they convinced the Navy to fund their research. Eight days later, the Naval Consulting Board recommended to Secretary Daniels that $50,000 be granted to Sperry’s team to carry out experimental work on aerial torpedoes.7
The fact that the Western Electric Company was working on radio devices encouraged Sperry to investigate the use of radio control in the aerial torpedo problem. However, after several tests, it was determined that the radio technology was too immature, and follow-on tests concentrated on maintaining course and measuring distance to the target.

On 10 November 1917, Glenn Curtiss, inventor of the flying boat, delivered an airframe designed to carry 1,000 pounds of ordnance a distance of 50 miles at 90 mph to the Sperry Flying Field at Copiague, Long Island, New York. A successful demonstration of the Navy's aerial torpedo was conducted 11 days later. Meanwhile, Rear admiral Ralph Earle, U.S. Navy Chief of the Bureau of Ordnance, had submitted his ideas on how the aerial torpedo might best be used to win the war and suggested that it might be most effective in defeating the U-boat menace. Earle suggested that aerial torpedoes could be carried on ships stationed off shore from the German submarine bases in Wilhelmshaven, Cuxhaven, and Flanders and used to destroy submarines, shipyard facilities, etc. Earle directed Lieutenant T. S. Wilkinson, U.S. Navy, to proceed to the Sperry Flying Field to observe and report on tests being conducted there. On 6 March 1918, the Curtiss-Sperry aerial torpedo made its longest successful flight, flying a distance of 1000 yards. Experiments with pilotless flight continued, and on 17 October 1918, a pilotless N-9 aircraft was successfully launched; it flew its prescribed course but failed to land at a preset range of 14,500 yards and crashed at sea. More than 100 tests were conducted by the Navy before the Armistice was signed on 11 November 1918 and, thus, like its British counterparts, the aerial torpedo never saw wartime service.

The Army was not to be left behind. After witnessing the Navy's aerial torpedo test on 21 November 1917, Major General George O. Squier, Chief Signal Officer of the Army, determined that a parallel effort by the Army should be undertaken at McCook Field, Dayton, Ohio. The U.S. Army aircraft board asked Charles Kettering to design an unmanned “flying bomb” that could hit a target at a range of 50 miles. Kettering's design eventually acquired the name “Kettering Bug” and had Orville Wright as an airframe consultant and Childe H. Wills of the Ford Motor Company as engine consultant on the project.

Launching the 530-pound “Bug” was accomplished using a dolly-and-track system, similar to the method
used by the Wright Brothers when they made their first powered flights. Once launched, a small onboard gyroscope guided the aircraft to its destination at an airspeed of about 120 mph. Control was achieved through a pneumatic/vacuum system, electric system, and an aneroid barometer/altimeter.9

To ensure that the Bug hit its target, a mechanical system was devised that would track the distance the aircraft flew. Before takeoff, technicians plotted the plane’s intended trajectory and forecasted the en route winds. Using this information, technicians also predicted the number of engine revolutions needed for the Bug to reach its destination. As the aircraft neared the end of its flight and the estimated number of revolutions had elapsed, the engine was mechanically turned off and the wings jettisoned. The Bug then began a ballistic trajectory into the target, with the impact detonating the explosive payload. The prototype Bug was completed near the end of WWI, and the Army ordered 25 Bugs on 25 January 1918.1,10 Flight tests began in September 1918, with the first successful flight on 22 October 1918.11 Unfortunately, the Bug failed in its testing, having made only eight successful test flights of 36, yielding a 22% success rate. In a fate like those of its Navy and British counterparts, the war ended before the “Bug” could enter combat.1 If the Army (Kettering Bug) and Navy (aerial torpedo) had worked jointly on these two concurrent efforts, perhaps an operational system could have been fielded before the Armistice.

**Interwar Years: Development of Target and Assault Drones**

During the Interwar Period, the development of unmanned aircraft was impacted by several advancements in the aviation world. These advancements included the rapid growth of the aviation industry, specifically in the air transport sector. This growth hampered testing and operating unmanned systems and continues to do so today. Advancements in radio furthered development and fielding of radio-controlled aircraft. And, finally, the successful demonstrations of aircraft against capital ships off the Virginia Capes in June 1921 stressed the need for development of radio-controlled target drones for use in fleet training exercises.

Economic competition and necessity forced the American government to develop an air management system during the 1920s and 1930s, a period of rapid growth in the airline industry. This system of airways, navigation aids, airdromes, weather stations, and control centers was developed to make long-distance overland flight safer and more regular. Radios were used to simplify traffic deconfliction. Today, this system, administered by the Federal Aviation Administration (FAA), continues to provide for the safety of air travel, particularly through the enforcement of visual and instrument flight rules under which the pilot is to maintain safe separation from obstacles such as terrain, buildings, and other aircraft. Such rules remain obstacles to UAV use in the national airspace.

Interest in unmanned flight waned with military budgets as hostilities ceased. However, this did not prevent enthusiasts from continuing to develop radio-controlled aircraft. A year after the war ended, the Army conducted 14 tests on the Bug, the most successful being a flight of 16 miles that ended in engine failure.2 The Navy sponsored similar projects, primarily using radio-control technology developed at the Naval Research Laboratory and tested on aircraft at the Lower Station, Naval Proving Ground, Dahlgren, Virginia. The final aerial torpedo test flight was conducted on 25 April 1921 and ended in failure. Rear Admiral Earle’s interest in aerial torpedo technology came to an end, and he recommended to Admiral Robert E. Coontz, the Chief of Naval Operations (CNO), that further tests be discontinued.3

On 11 May 1922, the Bureau of Ordnance directed that the Proving Ground at Dahlgren acquire one of the original N-9 aircraft used in the aerial torpedo experiments and fit it with a Norden radio-control system.4 The Naval Research Laboratory, in April 1923, announced...
controlled aircraft was flown remotely through all phases of flight—takeoff, maneuver, and landing. Tests continued over the next 14 months, but after an unsuccessful test on 11 December 1925, interest once again waned and, although the project was not canceled, it remained dormant until 1936.4,8

After WWI, both the Army and the Navy began disputing the future use of aircraft against surface ships. In March 1919, while on the staff of General Charles T. Menoher, Chief of the Air Service, General Billy Mitchell, a self-proclaimed advocate of air power, proposed a test to determine the outcome. Mitchell hypothesized that the days of the battleship were over, and he wanted to prove it in an actual test. He aggravated his seniors in both services, most notably Rear Admiral William A. Moffett, the Navy’s first Chief of the Bureau of Aeronautics.

Leveraging WWI technology used by the Germans in their unmanned torpedo boats, the U.S. Navy renamed the ex-USS Iowa (BB-4) to “Coast Battleship No. 4” and converted it to a radio-controlled target ship for gunnery training.5 Controlled by the USS Ohio (BB-10), in June 1921 it was used as a target vessel during tests off the Virginia Capes during which equipment for radio control of an F-5L aircraft had been demonstrated up to a range of 10 miles and that it believed that radio control of an aircraft during landing and takeoff was feasible.7 Tests continued, and on 15 September 1924, two test flights were made in which both the automatic stabilization and radio-control systems functioned flawlessly. A third flight was conducted that same day and, for the first time in history, a radio-controlled aircraft was flown remotely through all phases of flight—takeoff, maneuver, and landing. Tests continued over the next 14 months, but after an unsuccessful test on 11 December 1925, interest once again waned and, although the project was not canceled, it remained dormant until 1936.4,8

After WWI, both the Army and the Navy began disputing the future use of aircraft against surface ships. In March 1919, while on the staff of General Charles T. Menoher, Chief of the Air Service, General Billy Mitchell, a self-proclaimed advocate of air power, proposed a test to determine the outcome. Mitchell hypothesized that the days of the battleship were over, and he wanted to prove it in an actual test. He aggravated his seniors in both services, most notably Rear Admiral William A. Moffett, the Navy’s first Chief of the Bureau of Aeronautics.

Leveraging WWI technology used by the Germans in their unmanned torpedo boats, the U.S. Navy renamed the ex-USS Iowa (BB-4) to “Coast Battleship No. 4” and converted it to a radio-controlled target ship for gunnery training.5 Controlled by the USS Ohio (BB-10), in June 1921 it was used as a target vessel during tests off the Virginia Capes during which equipment for radio control of an F-5L aircraft had been demonstrated up to a range of 10 miles and that it believed that radio control of an aircraft during landing and takeoff was feasible.7 Tests continued, and on 15 September 1924, two test flights were made in which both the automatic stabilization and radio-control systems functioned flawlessly. A third flight was conducted that same day and, for the first time in history, a radio-controlled aircraft was flown remotely through all phases of flight—takeoff, maneuver, and landing. Tests continued over the next 14 months, but after an unsuccessful test on 11 December 1925, interest once again waned and, although the project was not canceled, it remained dormant until 1936.4,8

Figure 4. On 15 September 1924, for the first time in history, a radio-controlled Curtiss F-5L was flown remotely through all phases of flight.

Figure 5. The sinking of the ex-USS Alabama (BB-8) (top left) and the ex-SMS Ostfriesland (top right) off the Virginia Capes in 1923 demonstrated the vulnerability of surface ships to aircraft. However, it contributed to friction between General Mitchell (bottom center) and senior officers in the Army and Navy, most notably General Menoher (bottom left) and Rear Admiral Moffett (bottom right) and ultimately led to Mitchell’s court martial.
it sustained two hits on the forecastle, causing little damage. Ex-\textit{iowa} would continue to serve the Navy as a radio-controlled target vessel, ultimately being sunk by the USS Mississippi (BB-41) in March 1923 in the Gulf of Panama. History was made that June when the German battleship ex-SMS Ostfriesland and the U.S. pre-Dreadnought battleship ex-USS Alabama (BB-8) were sunk by aircraft. Mitchell had ordered his pilots to avoid direct hits in favor of near misses in the hopes that the explosive forces and water pressure would weaken the hull, resulting in a catastrophic failure.\textsuperscript{11} A board of naval officers who had observed the tests concluded that the “airplane is a powerful weapon of offense.”\textsuperscript{12} Moffett was forced to admit that “the bombing experiments during June and July . . . demonstrate beyond question that unhampered aircraft suitably armed can sink any type of ship.”\textsuperscript{13}

The results of the tests off the Virginia Capes sent virtually every maritime nation into crisis. Proponents of air power were convinced of the vulnerability of surface vessels of all types, while naval officers confessed that the tests were unrealistic in that the ships were not shooting in self-defense. To that end, throughout the 1920s, the Royal Air Force worked on a dual-purpose, radio-controlled unmanned aircraft that would perform as an aerial target and as an aircraft capable of weapons delivery. Efforts led to tests of a radio-controlled version of the Fairey IIIF reconnaissance float plane. Nicknamed the “Fairey Bee,” it was used successfully in exercises against the Home Fleet in the Mediterranean in January 1933. After these tests, the British went on to develop an all-wood version of the de Havilland Tiger Moth named the “Queen Bee,” which would see service through 1943.\textsuperscript{1,4,6,8}

In 1935, while attending the London Disarmament Conference, CNO Admiral William H. Standley, U.S. Navy, was provided with a demonstration of British aerial targets. On 1 May 1936, after conferring with Rear Admiral Ernest J. King, the Chief of the Bureau of Aeronautics, he directed the Bureaus of Aeronautics and Engineering, Rear Admiral Harold G. Bowen, to proceed with the development of radio-controlled targets. On 20 July 1936, Lieutenant Commander Delmar S. Fahrney was ordered as officer in charge of the Radio-Controlled Aircraft Project.\textsuperscript{4,6,14} Work commenced on the airframe at the Naval Aircraft Factory in Philadelphia, while the radio equipment was developed by the Radio Division at the Naval Research Laboratory. In his semiannual report for the last half of 1936, Fahrney introduced the term drone for aerial targets, a designation that endures to this day.\textsuperscript{1,4,7,8,15}

Tests continued through May 1938. On 1 June, all personnel and equipment were moved to San Diego and assigned to the Fleet Utility Wing. To remain clear of populated areas and the congested San Diego airspace, operations took place from Otay Mesa.\textsuperscript{14} Drones were used as aerial targets for the first time in the United States on 24 August 1938, when gun crews aboard the USS Ranger (CV-4) destroyed the target drone. The second live-fire test was held on 14 September 1938, when gunners aboard the USS Utah (AG-16) successfully destroyed a drone simulating a dive-bombing attack. Use of such drones continued over the following year until their use became routine, revealing deficiencies in fleet air defenses against a maneuvering target and accelerating improvements in fire-control systems. The Navy was now committed to funding and developing assault drones.\textsuperscript{1,4,7,8,14,15}

As a result of his work, Fahrney recommended that the aerial torpedo project of WWI be revived and that the Chief of the Bureau of Aeronautics, Rear Admiral Arthur B. Cook, investigate the use of radio control for testing new aircraft. In early 1938, the Navy commenced discussions with the Radio Corporation of America (RCA) to investigate the possibility of using television equipment to provide an operator in a trailing aircraft with information pertaining to drone instrumentation, as well as to provide the controller with a view ahead of the assault drone during the attack run. Such tests provided the Navy with data to further develop both the assault drone and guided missiles.\textsuperscript{1,4,7,8,14,15}

In March 1939, tests continued with the USS Utah at Guantanamo Bay, Cuba, in which only two drones were lost to gunfire. In a 1980 article in the U.S. Naval Institute Proceedings, Fahrney wrote, “This precipitated an agonizing reappraisal of the effectiveness of fleet antiaircraft defenses and resulted in redesign of both fire control and artillery systems.”\textsuperscript{14} The Army sent Captain George Holloman and Lieutenant Rudolph Fink of the Army Air Corps to observe and report on the Guantanamo tests. In a report to the Chief of the Air Corps, Major General Henry “Hap” Arnold, they recommended that the Army initiate its own developmental programs for radio-controlled targets and weapons.

As a result of these tests, the Army contracted with British actor Reginald Denny who, after a move to Hollywood, had started his radio-controlled aircraft

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure6.png}
\caption{The de Havilland Tiger Moth was modified to provide radio-controlled aerial target services to the Royal Navy.}
\end{figure}
Figure 7. In early 1942, Rear Admiral John Towers (left) convinced CNO Harold Stark (right) to pursue development of a radio-controlled aircraft capable of deploying ordnance.

company, Radioplane. In 1941, he began to supply the U.S. Army Air Corps with aerial targets, and over the course of WWII, he supplied them with more than 3,800 target drones. Eventually, Radioplane was acquired by what is now Northrop Grumman, which still produces unmanned aircraft.¹,⁸,⁹,¹⁵

WWII

While the United States watched as war spread across the world, the services continued their efforts to perfect radio control of aircraft, primarily as weapons delivery and guided missile platforms. Upon entry into the war, naval forces in Europe submitted an urgent operational need for a weapon that could be flown into the reinforced U-boat pens along the coast of France. In the South Pacific, the Navy searched for a weapon that could be used to suppress the Japanese defenses of Rabaul.¹⁴ Meanwhile, the Army’s Eighth Air Force attempted to develop a similar weapon that could be used to strike heavily defended strategic targets in mainland Europe, specifically facilities supporting the testing and use of Germany’s so-called “Vengeance Weapons”—the V-1 flying bomb, V-2 rocket, and V-3 cannon. However, as in WWI, these efforts were uncoordinated and mired in intra- and interservice politics, resulting in limited operational success.

Navy Efforts

The prewar target drone successes and the lack of sufficient aircraft carriers and their embarked air wings sparked a revival of the aerial torpedo concept. In January 1942, the Chief of the Bureau of Aeronautics, Rear Admiral John H. Towers, pushed to develop a radio-controlled aircraft capable of conducting offensive operations while carrying either a torpedo or depth charge (a forerunner of today’s UCAV).¹⁶ Within 3 months, he advised CNO Admiral Harold R. Stark that radar was being developed to replace television as the primary guidance system to allow operations under all conditions of visibility. During that same month, March 1942, the Navy conducted the first successful live attack with a radio-controlled aircraft armed with a dummy torpedo set against a maneuvering destroyer, USS Aaron Ward (DD-483) in Narragansett Bay. Controlled by a “mother” aircraft 20 miles away, the radio-controlled aircraft scored a direct hit on the destroyer’s target raft.¹,⁷ Further tests against the Ward were equally successful when the torpedo was deployed 300 feet from the target and successfully passed directly under the full length of the ship.⁷,¹⁴

Additional tests were conducted in April 1942 when Utility Squadron VJ-1 flew a BG-1 drone just beyond the wreck of the USS San Marcos in Tangier Sound in the Chesapeake Bay. A second test was conducted at Civil Aeronautics Administration Intermediate Field in Lively, Virginia. Using a BG-2 drone equipped with a television camera, the control plane, flying 11 miles in trail, successfully directed the drone’s crash into a raft being towed at a speed of 8 knots.⁷ Successful tests were conducted that summer during which depth charges and torpedoes were deployed from radio-controlled aircraft using television guidance systems. In all, 47 of 50 runs were satisfactorily completed with a maximum distance of 6 miles between the controlling aircraft and the drone in which a clear television picture was maintained. As a result, the Navy ordered 500 assault drones and 170 mother aircraft in preparation for WWII.¹,⁴,⁷,¹⁴

Rear Admiral Towers, impressed with the success of the tests, suggested that as many as 100 obsolete TBD Devastator aircraft be assigned to the program, and as SB2C Helldiver and SB2D Destroyer aircraft became obsolete, that they also be assigned. He was emphatic that special assault aircraft be developed in such a manner that they could be manufactured in quantities by industries not connected with the aircraft industry so that the industry would not be further burdened with a weapon unproven in combat.⁵,¹⁴,¹⁶

Figure 8. The only remaining TDR-1 on display at the National Museum of Naval Aviation in Pensacola, Florida.
In May 1942, after viewing films of successful tests conducted in the Narragansett and Chesapeake Bays, CNO Admiral Ernest King directed that a program be initiated to expedite the development and use of target drones as guided missiles in combat. Dubbed “Project Option,” under the command of Commodore Oscar Smith, drones produced by the Interstate Aircraft and Engineering Corporation were designated TDR-1 and outfitted with television guidance systems and controlled from a trailing TBF Avenger. They could be armed with either a torpedo or a 2000-pound bomb.\(^4\)\(^{14–18}\)

Meanwhile, on 29 August 1943, the Navy established Special Air Task Groups (STAG) to operate the drones. Although experiments conducted from both the training carrier USS Sable (IX-81) (one of the Navy’s two side-wheel propulsion Great Lakes training carriers) and the escort carrier USS Charger (CVE-30) determined the feasibility of deploying the TDR-1s from fleet carriers, their only combat use was to be from land. Between July and October 1944, STAG-1 deployed to the Solomon Islands. During that time frame, squadrons VK-11 and VK-12 deployed to Sunlight Field on Banika Island and executed 46 TDR-1 missions against selected Japanese targets from Stirling and Green Islands. Of the 29 missions that reached the target area, 18 were considered successful. The first successful mission on 30 July 1944 was conducted by four drones against the beached Japanese merchantman Yamazuki Maru on which anti-aircraft batteries were mounted. Other targets included bypassed Japanese units such as anti-aircraft installations, supply caves, and radar sites. Because the major conflict had moved far to the north, these strikes had little effect on the South Pacific. Additionally, Navy leadership believed that the number of available carriers in the Pacific was sufficient now that the tide had turned and that the TDR, a one-way weapon, was of limited value to the carrier battle group. The unit was disestablished on 27 October 1944, the day after its last mission against a Japanese target.\(^4\)\(^{14–19}\)

Meanwhile, on the other side of the world, the U.S. Army Air Forces (USAAF) and the U.S. Navy were busy with separate, uncoordinated efforts to attack strategic sites critical to the German war effort. On 6 July 1944, Commander, Air Force Atlantic Fleet, formed a special air unit tasked with converting PB4Y-1 Liberator bombers to assault drones. Reporting directly to Commander, Fleet Air Wing 7 (FAW-7), in Dunkeswell, United Kingdom, the special air unit was tasked with sending explosive drones into the U-boat pens in Helgoland, Germany. Dubbed “Project Anvil,” the first such mission was flown unsuccessfully on 12 August 1944. Project Anvil is blamed for the death of Joseph P. Kennedy Jr., the oldest brother of John F. Kennedy. During an Anvil flight, Kennedy’s aircraft mysteriously exploded in midair, incinerating Kennedy and his crew.\(^1\)\(^{20–22}\)

### Army Efforts

Operation Aphrodite was the code name of a secret program initiated by the USAF. The U.S. Eighth Air Force used Aphrodite both as an experimental method of destroying German V (“V” for “Vengeance”) weapon production and launch facilities and as a way to dispose of B-17 bombers that had outlived their operational usefulness—known as “war-weary” bombers.

The plan, first proposed by Major General James H. Doolittle in 1944, called for B-17 aircraft that had been taken out of operational service to be loaded with explosives and flown by remote control into bomb-resistant fortifications.\(^20\) In preparation for their final mission, several old B-17 Flying Fortress bombers were stripped of all normal combat armament and all other nonessential gear (armor, guns, bomb racks, seats, etc.), reducing each plane’s weight by about 12,000 pounds. The stripped aircraft were then equipped with a radio remote-control system, including television cameras mounted in the cockpit and at the bombadier’s station in the plexiglass nose, and loaded with up to 18,000 pounds of Torpex explosives, more than twice the normal bomb payload.\(^20\) The cameras were used to provide views of both the ground and the main instrumentation panel. These views were transmitted back to an accompanying control aircraft, allowing the craft to be flown remotely on its one-way mission.\(^1\)\(^{20,21}\)

Because the remote control did not allow for safe takeoff from a runway, each craft was taken aloft by a volunteer crew of two (a pilot and flight engineer), who were to fly the aircraft to an altitude of 2000 feet, at which point control would be transferred to the remote operators. Just before reaching the North Sea, the two-man crew would prime the explosive payload and parachute out of the cockpit. The “mothership” would then direct the pilotless aircraft to the target.

The first six Aphrodite flights, launched on 4 and 6 August 1944 against facilities for the V-1 flying bomb
at Siracourt, the V-2 rocket at Watten and Wizernes, and the V-3 cannon at Mimoyecques (all in northern France), were only moderately successful.\(^2\) The first mission was flown on 4 August 1944 against a V-1 launch site. One plane lost control after the flight engineer bailed out, and it crashed near Oxford, United Kingdom, making a huge crater and destroying more than 2 acres of the surrounding countryside and killing the pilot. The view from the nose of the other drone was obscured as it came over the target, and it missed by several hundred feet. In the mission’s next phase, one drone was shot down by flak and the other missed its target by 500 yards.\(^2\) During the second mission, flown on 6 August, more problems occurred. Although both crews were able to successfully abandon their B-17s without complications, one of the armed B-17s lost control and fell into the sea. The second also lost control but turned inland and began to circle Ipswich, United Kingdom. After several minutes, it fortunately crashed harmlessly at sea.\(^2\)

Follow-on flights were halted while Doolittle ordered a failure investigation. Concluding that Project Aphrodite was not successful against the “hard targets” for which it had been designed, U.S. Strategic Air Forces Headquarters ordered that aircraft be sent against industrial targets instead. Two more missions were flown against oil facilities in Heide, Germany, and both were failures. Bad weather and control problems caused misses. The only drone that actually hit the target did not explode, supplying the Germans with a completely intact B-17.\(^1\)

The final Aphrodite mission was flown on 30 October 1944, when two drones were launched against the submarine pens on Helgoland, Germany. One drone landed close enough to the target to cause significant damage and casualties. The second B-17 failed to respond to control signals from the mothership and continued eastward until it eventually crashed and exploded in Sweden.\(^1,2,21,23\) The USAF decided that the concept behind Project Aphrodite was unfeasible and scrapped the effort. In the course of the operation, only one drone caused any damage, and none hit its assigned targets.

Hence, Project Aphrodite, like the Navy’s Anvil, consisted of just a few flights and was canceled for the same reasons. Perhaps if the USAF and U.S. Navy had worked together as a joint team, an effective guided bomb could have been delivered to operational units.

The Cold War

The Cold War started immediately after WWII. America’s concern was to suppress the spread of Communism by maintaining a nuclear weapons advantage and developing a significant intelligence database to support strategic planning. Manned reconnaissance missions were conducted by U.S. Air Force and Navy crews against the Soviet Union, Cuba, China, and North Korea. Aircrews conducted both intentional overflights of sovereign territory as well as Peacetime Aerial Reconnaissance Program flights along unfriendly borders and coastlines. Between April 1950 and April 1969, 16 such missions encountered hostile fire, with the loss of 163 lives.\(^24,25\)

**The Korean War and the 1960s**

When North Korean forces launched a sudden all-out attack on the Republic of Korea on 25 June 1950, U.S. forces in the Pacific were unprepared. In fact, the U.S. Army had no troops on the peninsula, the Air Force had only a few air wings available in the region, and the Navy had just one cruiser, four destroyers, and a few minesweepers on station in the Sea of Japan. Within 36 hours, the United Nations called on its members to assist the South.\(^23\) The next month, in a record Pacific transit, USS Boxer (CV-21) carried badly needed Air Force and Navy aircraft and personnel to the war zone. Boxer would make three more deployments to Korea. On 5 August 1952, while engaged in combat operations, it suffered a fire on its hangar deck. Two weeks later, after repairs, it was back on station with Guided Missile Unit 90 (GMU-90) and embarked with six F6F-5K Hellcat drones. Each Hellcat carried a 1000-pound bomb under the fuselage and a television and radio repeater pod mounted on the wing.

On 28 August, they took off under radio control from Boxer and, under the radio control of Douglas AD-4N Skyraiders from Composite Squadron 35 (VC-35), were guided against selected targets. In all, six such missions were conducted between 28 August and 2 September 1952 against power plants, rail tunnels, and bridges in North Korea, but with an operational success rate of less than 50 percent, the program was dropped. The Hellcats continued to be used by the Navy as targets at China Lake, California, into the 1960s.\(^7,26,27\)
In the early 1950s, Ryan Aeronautical Company developed and built 32 jet-propelled, subsonic UAVs known as Ryan “Firebees.” The Firebee design lives to this day and has dominated UAV history. The Firebee UAV (originally designated Q-2A/B) dates to 1951 and was used initially as a target drone. The political fallout of Francis Gary Powers being shot down over the USSR in his U-2 in 1960 led many in the DoD to start thinking about unmanned reconnaissance of the USSR. Ryan Aeronautical modified some of its standard Firebee training targets into reconnaissance UAVs (recon-UAVs) and designated them the 147A “Firefly.” The Firefly, renamed the “Lightning Bug,” was modified considerably to change it from a target drone to a recon-UAV. In the early 1960s, Ryan Aeronautical Company designed and developed more than 20 versions of its famous Lightning Bug unmanned subsonic target drone. One model, the AQM-34N, had wet wings, meaning that it carried fuel in its wings, giving it a range of approximately 2500 miles.

In the mid- to late 1950s, the United States needed a way to counter the threat of a rapidly growing Soviet submarine force. The Navy’s DASH (QH-50) was the first operational unmanned helicopter designed for a combat role. In 1960, a QH-50 (powered by a Gyrodyne-Porsche engine) made its maiden flight at Patuxent River Naval Air Station in Maryland. During the 1960s, almost 800 QH-50s were built. The DASH unmanned helicopters were flown remotely from a destroyer’s deck and carried Mk-44 homing torpedoes or Mark 17 nuclear depth charges. They could also be controlled from manned aircraft or ground vehicles and could drop sonobuoys and flares, perform rescues, transport cargo, illuminate targets, deploy smoke screens, perform surveillance, and target spot for naval fire support. QH-50s subsequently carried a mini-gun and even dropped an assortment of bomblets in various covert missions over Vietnam in the late 1960s. In 1970, the DASH program was canceled, and remaining QH-50s were used as target drones.

The Bikini program was a 7-year U.S. Marine Corps research and development program, beginning in 1959, that looked at methods of providing organic, real-time reconnaissance for a battalion commander. It was designed to consist of a two-man drone team with a jeep-mounted launcher and an airborne drone with a 70-mm camera. However, the U.S. Marine Corps determined that technology at that time was inadequate and the system was therefore not fielded. However, the 7-year program did lead to development and employment ideas for UAVs three decades later.

On 27 October 1962, President Kennedy demanded that the Soviet Union dismantle its missile bases and remove its nuclear warheads from Cuba. That same day, Soviet SA-2 missiles in Cuba shot down a U-2, killing its pilot Major Rudolph Anderson Jr. With the Cuban Missile Crisis at its peak, the United States needed photographic confirmation that the Soviets had either removed their missiles or refused to do so. Only two U-2s were immediately available to continue the Cuban overflights. Because only two of the Ryan Lightning Bugs had been built and their operational testing had not yet been completed, RF-8A Crusader aircraft were used to image Cuba. The Cuban Missile Crisis demonstrated a need for a concerted UAV development effort by the U.S. military. By the August 1964 Tonkin Gulf incident, UAVs were finally accepted for wartime service.

In 1964, while deployed to Kadena Air Base, Okinawa, the Strategic Air Command’s 100th Strategic Reconnaissance Wing launched Teledyne-Ryan AQM-34 drones from DC-130 Hercules aircraft flying along the coast of mainland China. These UAVs penetrated Chinese airspace and obtained high-quality photographic imagery of military facilities and troop movements and

---

**Figure 11.** A QH-50C DASH drone is recovered aboard USS Hazelwood (DD-531).

**Figure 12.** Still used today, two Teledyne-Ryan AQM-34 drones are carried on the wings of a DC-130 Hercules aircraft.
heavier payloads. These UA Vs could now perform signals intelligence missions in addition to their IMINT roles. Late in the Vietnam War, UAVs also performed psychological operations via leaflet drops.

In 1965, the U.S. Air Force established a requirement for a long-range recon-UAV. Ryan developed the model 154, Compass Arrow, designed to fly at 78,000 feet; it was also designed with minimal heat and radar signature, thus becoming the first UAV to use stealth technologies. Like its cousin the Lightning Bug, Compass Arrow was launched from a DC-130, was recovered via MARS, and had electronic countermeasures to improve its survivability. The program failed to move forward because of various political, financial, and technical problems. So while the Lightning Bug was an enormous success, both as a drone and a recon-UAV, Compass Arrow was a failure and possibly led to the lack of UAV acceptance at that time by many in the aviation business.

But with the success of the Lightning Bug, the modern UCAV was born. After 4 years of research and development, Ryan Aeronautical took its Lightning Bug design and showed that it could strike and destroy a ship from a distance of about 100 miles. In 1971, the Lightning Bug (model BQM/SSM) flew a perfect demonstration, slamming into the side of the ex-USS John C. Butler (DE-339). But the BQM/SSM was competing against the more versatile Harpoon weapon system, which was all-weather and could be employed from a variety of platforms. Hence, the Navy chose Harpoon and canceled the BQM/SSM effort.

Like the BQM/SSM, the BGM-34A was developed because of hostilities. Israel was concerned about Soviet-made anti-aircraft artillery emplacements along the Suez Canal. In 1971, Teledyne-Ryan Aeronautical (TRA) developed a UCAV that could deliver air-to-surface munitions. TRA again used the Lightning Bug as the basic frame and then used pieces from other UAVs to develop the final BGM-34A product. In less than a year, TRA had developed a UCAV that was used to fire a powered, guided air-to-surface missile against a simulated target. American military thinkers had the idea of using these UCAVs on the first wave to soften a target then to finish off the target with manned aircraft. The Israelis agreed and used the BGM-34A against Egyptian missile sites and armored vehicles in the October 1973 Yom Kippur War and again in 1982 against Syrian missile emplacements in the Bekaa Valley. These Israeli UCAVs certainly saved the lives of Israeli pilots. Americans never used this UCAV in Vietnam because it could not perform better than manned technology. After the Vietnam conflict, a few improvements were made to the BGM (such as models 34B and C), but generally speaking, interest in UAVs in general waned and further expenditures on recon-UAVs were put on hold. Additionally, UAVs had to compete with new high-speed missile systems, long-range bombers, and cruise missiles.
So, with drastic budget cuts, UAV development basically ceased for about a decade.

In the late 1970s, the U.S. Army began a major UAV acquisition effort known as Aquila. It was originally estimated to cost $123 million for a 4-year development cycle, followed by $440 million for the production of 780 vehicles. Its original mission was to be a small propeller-driven, man-portable UAV that provided ground commanders with real-time battlefield intelligence. As development continued, requirements grew and the UAV’s small size could no longer handle the avionics and payload items the Army wanted, such as autopilot, sensors to locate the enemy in all conditions, laser designators for artillery projectiles, and abilities to survive against Soviet anti-aircraft artillery. The Army abandoned the program in 1987 because of cost, schedule, and technical difficulties (and after $1 billion in expenditures).

The First Persian Gulf War

Israeli successes in 1973 and 1982 led the United States to finally procure a new UAV of its own, primarily to conduct battle damage assessment for the U.S. Navy. This Israeli Aircraft Industries UAV, Pioneer, has been used by U.S. forces since the late 1980s. Pioneer was procured starting in 1985 as an interim UAV capability to provide IMINT for tactical commanders on land and at sea. Pioneer skipped the traditional U.S. development phase of the acquisition process, and nine systems, each with eight air vehicles, were procured beginning in 1986, at an estimated cost of $87.7 million. Similar to Aquila, Pioneer is a small propeller-driven aircraft. The Pioneer encountered unanticipated problems almost immediately after delivery. Recoveries aboard ship and electromagnetic interference from other ship systems were serious problems that led to a significant number of crashes. The Pioneer system also suffered from numerous other shortcomings. The Navy undertook a $50-million research and development effort to bring the nine Pioneer systems up to a level it described as a “minimum essential capability.”

However, the Pioneer flew 300+ combat reconnaissance missions during Persian Gulf operations in 1990–1991. The system received extensive acclaim for outstanding performance in Operations Desert Shield and Desert Storm. Army, Navy, and Marine Corps commanders lauded the Pioneer for its effectiveness. During the Persian Gulf War, all the UAV units at various times had individuals or groups attempt to signal the Pioneer, indicating their willingness to surrender. The most famous incident occurred when the USS Missouri (BB-63), using its Pioneer to aim its accurate 16-inch gunfire, devastated the defenses of Faylaka Island off the coast near Kuwait City. Shortly thereafter, while still over the horizon and invisible to the defenders, the USS Wisconsin (BB-64) sent its Pioneer over Faylaka Island at low altitude. When the Pioneer came over the island, the defenders recognized that they were about to be targeted, so using handkerchiefs, undershirts, and bedsheets, they signaled their desire to surrender. Since the Persian Gulf War, Pioneer has flown operationally in Bosnia, Haiti, and Somalia, and, of course, it has become one of the primary weapons of choice in the Second Persian Gulf War and the Global War on Terror.

CONCLUSIONS

In this review article, we have presented the reader with a brief history of early unmanned aircraft, focusing on WWI through the First Persian Gulf War. Nowadays, unmanned aircraft such as the Predator are armed with laser designators and Hellfire missiles so they can perform attack orchestration and target termination, not just ISR. Other unmanned aircraft, such as Global Hawk, operate almost completely autonomously, remotely piloted by operators thousands of miles away—this type of vehicle uses GPS and transmits a live video feed back to its operations center. In addition, other unmanned aircraft are so small that they can be hand launched and have become useful in street fighting or other types of close-in engagements, where they can assist the operator in discovering imminent ambushes.

Unmanned aircraft have come a long way over the past century. Just as Brigadier General Billy Mitchell crusaded against traditional military thinking when it came to the use of airpower, current UAV crews are striving for greater recognition of their aircraft and the operations they perform, from ISR to force protection to precision strike.

ACKNOWLEDGMENTS: The authors thank the following sources for the images included in this article.

Figure 14. A Pioneer UAV is recovered aboard an Iowa-class battleship.

REFERENCES

29 Memorandum from Commanding Officer, USS Boxer (CV-12), to Chief of Naval Operations, 8 Sep 1952 (in possession of authors).
33 Memorandum from Commanding Officer, USS Boxer (CV-12), to Chief of Naval Operations, 8 Sep 1952 (in possession of authors).

The Authors

John F. “Jack” Keane is a member of APL’s Principal Professional Staff. Since joining APL in 1997, he has worked on analyses for missile defense and precision engagement systems. Mr. Keane has been the supervisor of the Precision Engagement Systems Branch in the Force Projection Sector since 2007. Steve Carr is a member of APL’s Principal Professional Staff. After retiring from the U.S. Air Force in 2001, Steve joined the Space Department, where he worked on space environmental issues and their concomitant effects on military systems. He became the Command and Control Group Supervisor in the former Global Engagement Department in 2005, and he became the Sensors and Communication Systems Branch Manager in the Air and Missile Defense Sector in 2012. For further information on the work reported here, contact Jack Keane or Steve Carr. Their e-mail addresses are jack.keane@jhuapl.edu and steve.carr@jhuapl.edu, respectively.

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.
The concept of integrating unmanned aircraft systems (UASs) into the National Airspace System (NAS) is being developed by multiple governmental and nongovernmental organizations and spans multiple system development efforts. The ability for an unmanned aircraft to “see and avoid” is the primary technical challenge. When UASs were first introduced into the NAS, agreements with the Federal Aviation Administration required either visual ground observers or manned aircraft following the UASs and restricted operations to daytime only. These conditions significantly reduce the quality and quantity of DoD UAS training in the United States. This article covers the DoD Ground-Based Sense and Avoid (GBSAA) technology initiatives to reduce the burden of visual observers, as well as APL’s role and contributions to GBSAA. The first initiative described is the Army’s initial GBSAA system, which implemented a safe-state concept for the terminal-area operations access profile. The second initiative described is the Army’s current follow-on GBSAA system, which allows for greater flexibility in flight operations while providing information for maneuver decisions for terminal-area operations and lateral-transit access profiles. The final initiative discussed is the Marine Corps system, which uses a safe-state concept to support the lateral-transit access profile. In 2013, the Federal Aviation Administration issued a Certificate of Waiver or Authorization for the Marine Corps GBSAA system, a major step toward UAS airspace integration.

INTRODUCTION

One can imagine that in the not-too-distant future, remotely piloted civilian aircraft will be flying everything from cargo to passengers or providing services from traffic spotting to supporting fire and police forces. Dramatic changes to aviation begin with modest starts but have long-lasting impacts. The development of
Currently, there is little airspace in the NAS where the military services can develop, test, train, and operate UASs. The continued yearly increase in the number and types of UASs puts a further strain on the limited restricted airspace. Gaining access to civil U.S. airspace, as well as to international airspace, to perform these functions is critical for training and integrating future capabilities.

Today, there are two primary means by which the U.S. military has the capability to fly UASs in the NAS. The first method is to fly only in active restricted or warning area airspace. The U.S. military controls restricted areas, and it assumes responsibility for the safety of any UAS flights within the restricted airspace. The second means allows for flights outside of the restricted areas through the Certificate of Waiver or Authorization (CoA) process with the Federal Aviation Administration (FAA). The CoA process is an agreement for special provisions that the FAA levies on the UAS operator to provide safe operation in the NAS. These provisions in the past have been through the use of visual observers either on the ground (remaining in sight of the UA) or in a chase aircraft. For civil operators, special airworthiness certificates are available for experimental purposes only. Each of these methods comes with its own constraints and limitations. Temporary flight restrictions are another short-term means to fly UASs in the U.S. airspace but are mainly limited to emergency response or national security considerations.

In 2006, APL began working with the DoD to develop methods to ease access to the NAS for UASs. The DoD established a tri-service joint integrated product team (IPT) in which each service provided leadership and resources to solve the “see and avoid” problem. The joint IPT was superseded by the Office of the Secretary of Defense (OSD) Airspace Integration (AI) IPT in order to more closely coordinate the DoD efforts for UAS access to the NAS. This new tri-service AI IPT has a specific subgroup to deal with GBSAA issues and challenges.

### GBSAA OVERVIEW

Ground-Based Sense and Avoid (GBSAA), using electronic sensors to provide a safe method for unmanned aircraft (UA) to remain clear of other aircraft while safely flying within the National Airspace System (NAS), is one of these modest beginnings to a historic change in aviation.

The proliferation of unmanned aircraft systems (UASs) in both the civilian and military worlds has necessitated the need for UA to safely integrate with manned aircraft in the United States’ NAS. As shown in Fig. 1, the DoD has significantly increased its use of UASs in the past 15 years. As the U.S. military continues to support overseas operations, the contributions of UASs, in terms of hours and expanded roles, continue to increase, and the number and variety of UASs are expected to increase significantly in the next decade. The U.S. military has deployed more than 20 different UASs that are flown and operated overseas by the U.S. Marine Corps, Navy, Air Force, Army, and Army National Guard. The role of the UAS, which once included reconnaissance only, now includes strike, force protection, and signals collection.

The Challenge

A significant challenge that could hinder the growth and further incorporation of UASs into the U.S. military is the ability to operate in the NAS and airspace worldwide. The Federal Aviation Regulation part 91.113 states that “vigilance shall be maintained by each person operating an aircraft so as to see and avoid other aircraft.” Because there is no pilot onboard a UA to perform see-and-avoid responsibilities, special provisions must be made to achieve a target level of safety that is consistent with that of onboard piloted aircraft when the UA is operating outside of designated safe states. A safe state refers to an area within the NAS that is fully controlled, such as a restricted area, warning area, and Class A airspace area.
airspace. The access profiles include visual line of sight operations, terminal-area operations, operating area operations, lateral-transit operations, vertical-transit operations, and dynamic operations. The near-term focus of GBSAA is on enabling terminal-area operations, lateral-transit operations, and vertical-transit operations with a desired end state of dynamic operations.

Terminal-area operations are focused on a fixed volume of airspace, typically near a small airport or Class D airspace. This access profile enables UAS operators to practice takeoffs and landings and to fly in a region of airspace that is generally clear of surrounding aircraft. Figure 2 shows how a UA will fly in an operational volume that is within the ground sensor coverage area (i.e., surveillance volume).

Lateral-transit operations are focused on enabling UASs to transit from one region of airspace to an adjacent region of airspace through a lateral corridor. For example, as shown in Fig. 3, lateral-transit operations enable a UA to transit from a Class D airspace to an adjacent restricted area. Once in the restricted area, and assuming the restricted area was reserved for UA operations, the UAS operator can fly the UA within the designated portions of the restricted area to conduct testing or training. This profile can be used by all classes of UAs.

Figure 2. Terminal-area operations access profile.\textsuperscript{2} ATS, air traffic services; GCS, ground control station; IFR, instrument flight rules; VFR, visual flight rules.

The OSD AI IPT developed an Unmanned Aircraft System Airspace Integration Plan,\textsuperscript{2} which describes six access profiles to enable UAS access to nonsegregated airspace. The access profiles include visual line of sight operations, terminal-area operations, operating area operations, lateral-transit operations, vertical-transit operations, and dynamic operations. The near-term focus of GBSAA is on enabling terminal-area operations, lateral-transit operations, and vertical-transit operations with a desired end state of dynamic operations.

Terminal-area operations are focused on a fixed volume of airspace, typically near a small airport or Class D airspace. This access profile enables UAS operators to practice takeoffs and landings and to fly in a region of airspace that is generally clear of surrounding aircraft. Figure 2 shows how a UA will fly in an operational volume that is within the ground sensor coverage area (i.e., surveillance volume).

Lateral-transit operations are focused on enabling UASs to transit from one region of airspace to an adjacent region of airspace through a lateral corridor. For example, as shown in Fig. 3, lateral-transit operations enable a UA to transit from a Class D airspace to an adjacent restricted area. Once in the restricted area, and assuming the restricted area was reserved for UA operations, the UAS operator can fly the UA within the designated portions of the restricted area to conduct testing or training. This profile can be used by all classes of UAs.

\textsuperscript{2}ATS, air traffic services; GCS, ground control station; IFR, instrument flight rules; VFR, visual flight rules.
Figure 3. Lateral-transit operations access profile.²

Figure 4. Vertical-transit operations access profile.²
Potential regulatory changes may someday provide an alternative means of complying with this regulation, but for now the DoD has been using the CoA process to gain access to the nonsegregated NAS for military UAS operations. Under the CoA process, the FAA conducts a comprehensive operational and technical review for the applied method of UAS access. The FAA levies limitations on the UAS operations to ensure the safety of others within the NAS. Until GBSAA, these strict limitations have limited the areas of operation and the time of day for operations and have required the use of visual observers or chase aircraft. Although these restrictions have provided the DoD with limited access to airspace, it has proven logistically difficult, expensive, and cumbersome to conduct testing and training. For example, for an operator to fly a UAS within the nonsegregated NAS, the FAA would require a manned chase aircraft with a visual observer onboard to fly within 1 mile of the UA at all times, while maintaining voice communications back to the ground control station for the UA, effectively doubling the costs and significantly increasing the manpower for one UA flight.

GBSAA was identified not as the “golden key” to unlock all airspace within the NAS for UA activity but as a first step to lessen the impact of FAA limitations.

Vertical-transit operations are similar to lateral-transit operations but enable UAs to vertically transit through a corridor to airspace above the airfield. Figure 4 shows how the UA could fly from a terminal area through a vertical corridor to Class A airspace. This profile is most commonly used by larger UAs that are approved for flight in IFR airspace and that are able to operate at higher altitudes.

Finally, the dynamic operations access profile enables UA access to the airspace, just like any other manned aircraft. Unlike terminal-area operations in which the operational volume is generally clear of other aircraft, the dynamic operations access profile allows a UA to fly among other aircraft using a GBSAA solution. Figure 5 shows how the UA would interact in the airspace under dynamic operations.

**PATHWAY TO AUTHORIZATION**

As previously stated, currently there are limited ways UAs can gain access to nonsegregated airspace within the United States. UAs cannot comply with Federal Aviation Regulation part 91.113, the need for the pilot onboard an aircraft to see and avoid other aircraft.
placed on UA operations because of the UAS’s inability to see and avoid other aircraft. Each implementation of a GBSAA system requires a rigorous review by the particular military service’s safety organization, followed by additional scrutiny by the FAA. The FAA reviews all aspects of the proposed solution, including, but not limited to, software implementation, the engineering design of the methodology, implementation of the logic algorithms, and the operations and procedures to be followed by the UA and its crew. Additionally, the FAA requires a comparative safety assessment to ensure that this “new” system can be considered as safe, if not safer, than previously approved CoAs that use visual observers.

The following sections describe GBSAA methods used by the Army and the Navy/Marine Corps to gain limited access to nonsegregated airspace for UASs. Each GBSAA system handles the mitigations differently through the use of effectively used electronic means and operations and procedures to remove the need for visual observers and chase aircraft to be used during UA flights in the NAS.

GBSAA ARMY ACTIVITIES

Background

As the GBSAA lead designated by the OSD AI IPT, the Army’s Program Manager for Unmanned Aircraft Systems Unmanned Systems Airspace Integration Concepts Product Directorate has led the way on a variety of GBSAA activities, including an initial GBSAA system at El Mirage; a follow-on GBSAA system currently under development at Dugway Proving Grounds (Utah), which will ultimately be used at multiple sites; and other process improvement and standardization activities.

One of the restrictions of using visual observers is that UA operations can take place only during the day. However, to train under conditions similar to those of actual operations, there was a need to also perform night operations. To enable 24-h-a-day Gray Eagle operations at El Mirage, California, the Army developed an initial GBSAA system. This system allowed UA to use the terminal-area operations access profile so that the USA operators could practice takeoffs, landings, and terminal-area operations. The system was designed such that a GO works in close coordination with the UAS operator. The GO is stationed at a GBSAA system that uses a simple red light and green light display. If the light is green, then it is safe for the UA to conduct operations within the operational volume. If the light is red, then the UA must land as soon as possible. After completing a safety case review with the FAA, the Army was granted a CoA for Gray Eagle operations at El Mirage using the GBSAA system. On 27 April 2011, it conducted the first flight of a UA in the NAS using a GBSAA system.

The Army is currently developing a follow-on GBSAA system to enable airspace access for additional locations to support both the terminal-area operations access profile and the lateral-transit access profile. A version of this software is currently being tested at Dugway Proving Grounds.

GBSAA Architecture and Description

The initial system installed at El Mirage, California, used ground-based radars along with a stand-alone GBSAA system to monitor the local airspace. This system implemented a concept referred to as safe states. A safe state is a volume of airspace in which the UA is safe from an intruder aircraft. Examples of safe-state operations include landing at the airfield or operating in a restricted area. For the initial El Mirage system, the only safe-state operation was landing the UA at the El Mirage airfield because the concept implemented the terminal-area operations access profile. For this system, the GBSAA algorithm assesses whether an aircraft, under the worst-case assumptions, can reach the operational volume before the UA can land back at the airfield. This included the assumptions that the surrounding aircraft was traveling at 250 knots, co-altitude with the UA, and was headed directly for the UA. On the basis of the radar locations and associated sensor coverage, a surveillance volume was defined that provided enough time for the UA to land before the intruder aircraft could reach the operational volume on the basis of worst-case assumptions. Within the surveillance volume, an operational volume was also defined in which the UA could operate. If the GBSAA system determines that there are no detected intruders or system issues, then a green light is displayed. If there is a detected aircraft in the surveillance volume or there is a system issue, then a red light is displayed along with an audio alert. This system is composed of three subsystems, one for sensing and tracking, one for alerting, and one for health and integrity monitoring.

The system currently under development also uses ground-based radars to monitor the local airspace; however, this system uses complex algorithms to fuse the data from the ground-based radars, assess which aircraft may pose a threat to the UA, and provide airspace information to the GO. This system is composed of multiple subsystems to perform the sensing, fusing, classifying, assessing, informing, and monitoring functions. This concept is designed to be robust enough to enable different types of operations and access profiles at different sites, including terminal-area operations and lateral-transit operations. The current system under development also uses the concept of a surveillance volume and operational volume. However, the operational volume is significantly larger than the previous system’s operational volume, on the basis of different underly-
ing assumptions. The current system not only uses the aircraft's current position, but it also takes into account the altitudes, headings, and speeds of the UA and the intruder aircraft. There are still assumptions about aircraft maneuverability and acceleration; however, using fewer assumptions about the intruder aircraft and taking into consideration information on the UA results in more operational time in the NAS. The other major difference is that this system does not require the UA to return to a safe state once an intruder aircraft has been detected. Rather, the GO provides information to the UAS operator so that the latter can make an informed decision on what action to take.

**Airspace Architecture**

The most common airspace architecture is a military airfield adjacent to a restricted area. For UA to transit to the adjacent restricted area, they must transit through a segment of nonsegregated airspace. These transits are on the order of minutes and are critical to support UAS operator training. GBSAA operations are well suited for these types of operations because of the short transits and the availability of existing ground-based radars. The current GBSAA system plans to incorporate existing sensors as well as GBSAA-specific sensors. Existing sensors, referred to as inorganic sensors, are typically owned by the FAA and support only range and azimuth detections and tracking. GBSAA-specific sensors, referred to as organic sensors, support range, azimuth, and altitude detections and tracking.

To test airspace integration concepts, the Army has installed a test bed at Dugway Proving Ground in Utah that allows UA operations entirely within a restricted area. The GBSAA system installed at Dugway enables the exploration of different algorithms and geometries to test the robustness of the sense-and-avoid algorithms.

**Roles and Responsibilities**

The GBSAA system is monitored by a dedicated person, referred to as the GO. The GO works closely with the UAS operator to ensure safe operation of the UA in nonsegregated airspace. For both the initial El Mirage system and the current system under development, the GO monitors the GBSAA system and alerts the UAS operator when an action is required. One of the major underlying assumptions for both systems is that the GBSAA operations will not have an impact on current air traffic control (ATC) operations. That is, the GO interacts only with the UAS operator, and the UAS operator interacts with ATC with or without GBSAA.

**APL’s Role**

APL provided critical contributions to the Army’s initial system at El Mirage and continues to provide major systems engineering contributions to the system currently under development. Initially, APL led the development of architectures for both current and future operations for El Mirage. This was important to understanding how future operations using GBSAA would affect the current operations with visual observers. APL also developed and maintained the system-level requirements associated with the initial GBSAA system.

For the current system development, APL continues to play a critical role in the systems engineering activities. This includes documenting the concept of operations, developing the system-level requirements, maintaining traceability to detailed requirements developed by other contractors, developing operational and system architectures, and leading the planning and execution of system-level integration and verification. Thorough systems engineering and traceability are imperative for this safety-critical system to convince the certifying organizations of both the Army and the FAA that the system is sufficient as an alternative means of complying with see-and-avoid regulations.

Because the Army is the GBSAA lead, it also oversees other cross-service activities, to which APL is the prime contributor. One major initiative is the development of a common set of GBSAA requirements across the services to aid in future development and procurement activities. To develop a standardized set of requirements, APL has led a requirements workshop with cross-service representation and several follow-on meetings. Another initiative is to standardize certification activities. The Army and APL co-led a safety workshop to share lessons learned from the El Mirage safety case development and a software certification workshop to discuss methodologies, standards, and criteria.

**GBSAA NAVY ACTIVITIES**

**Background**

The Navy’s implementation of GBSAA, led by PMA-262, is for Marine Unmanned Aerial Vehicle Squadron 2 (VMU-2) at Marine Corps Air Station (MCAS) Cherry Point in North Carolina. VMU-2 operates the RQ-7B Shadow and provides aerial surveillance for the II Marine Expeditionary Force.

VMU-2 continues to be deployed overseas to Afghanistan, and use of the Shadow UAS is a daily occurrence. In the past, the Marines have been forced to gain most of their training flight hours overseas because of logistical struggles to train in the NAS before being deployed. This lack of state-side training and preparation often results in inexperienced flight crews learning on the job in often high-stress and time-critical environments. There are two main ways that VMU-2 conducts UAS training flights in the United States: (i) launch and recovery from an airfield embedded in the restricted
area and (ii) through the use of visual observers, which observe the UA and surrounding air traffic. Each of these approaches has logistical issues.

Near Cherry Point, there is a large restricted area, R-5306A, that resides over a small airfield, Atlantic Field Marine Corps Outlying Field. Because R-5306A is continuously in effect from the surface to, but not including, flight level 180, VMU-2 is able to launch its Shadow UAS from Atlantic Field and conduct training missions completely within the restricted airspace, without having to enter nonsegregated airspace. This approach involves VMU-2 packing up all equipment (including multiple cargo trucks) and “deploying” to Atlantic Field for multiple days at a time; the drive to Atlantic Field is longer than 2 h. For squadron members, this mini-deployment means additional time away from their families, increased burden of planning and packing, potential for delayed and/or canceled flight time if the weather does not cooperate, and significant costs for the Marine Corps to deploy near their home base. Because of these logistical burdens, VMU-2 is not able to gain sufficient training flight hours using this method.

Alternative methods for the UAS to gain access to the restricted areas around MCAS Cherry Point include using either ground-based observers or an observer in a chase aircraft to monitor the UAS flight and provide information to separate the UA from other aircraft with which it may be in conflict. MCAS Cherry Point currently uses ground observers. The primary duties of the ground-based observer are to report observed segment status; report UAS observation, weather, and traffic updates; and acknowledge when transit is complete. At Cherry Point, ground-based visual observers covering the Class E airspace are employed in teams of two Marines. One individual visually scans and acquires the UAS, maintaining constant visual contact with the air vehicle at all times. The second observer communicates with the UA commander (UAC)/pilot in command and assists the first member in acquiring the UAS and scanning for traffic. Visual observer teams need to be deployed at 1-nautical-mile intervals along the entire planned flight path of the UAS and must remain in constant visual contact with the UAS at all times. This solution requires extensive planning and manpower. The Marines typically use a team of 12 Marines to provide enough visual observers for the Shadow to make a typical transit from Cherry Point to the restricted area.

Neither of the two previously described solutions increase training flight hours, nor would they be considered optimal. To maximize training opportunities and minimize cost, Shadow UAS operations at MCAS Cherry Point dictate that the UA must transit between the Cherry Point Class D surface area (CDSA) and the restricted areas R-5306A and R-5306C through the NAS. A GBSAA solution was chosen as risk mitigation to the see-and-avoid requirement while flying within the NAS.

### GBSAA Architecture and Description

In lieu of ground observers or chase aircraft, the GBSAA system at Cherry Point provides the necessary see-and-avoid capability through the use of technology and procedures. The GBSAA system analyzes track data from the MCAS Cherry Point surveillance radar, the ASR-11, and displays noncooperative air hazards, in addition to cooperative air traffic, within the local operating area.

An underpinning of the GBSAA concept classifies aircraft into two categories: cooperative aircraft and noncooperative aircraft. Cooperative aircraft are defined as those that are squawking a discrete Mode 3/C beacon code. These aircraft are assumed to be under the control of ATC and are therefore not considered a safety risk to the UA because ATC personnel are required to separate controlled aircraft from all other aircraft. Noncooperative aircraft are defined as those transmitting a Mode 3 code ending in “00” or not transmitting any electronic identification. It is assumed that these aircraft are not under the control of ATC, and they are therefore monitored and scored by the GBSAA system. Both types of aircraft are displayed to the GO, but they are displayed and handled differently because of their varying levels of risk potential.

The GBSAA system operates as a stand-alone console with nominal initial employment in an ATC facility radar room; the console is referred to as the GBSAA operator console (GOCon). The console’s primary indicator, referred to as the alerting system, is a three-color operational decision (“stoplight”) display that will show red, yellow, or green lights depending on the noncooperative intruder’s ability to interfere with the UA’s transit.

The GOCon software supports the reception and tracking of data from the ASR-11 and displays the resultant tracks. The software is built on a government off-the-shelf system currently used by Navy range facilities, the Stand Alone Radar Console (STARCON) system. The primary function of the GOCon’s display is to alert the GO to the highest threat level determined by GBSAA system. The GO uses the alerting system, which consists of a three-color (red, yellow, and green) stoplight display, to continually monitor the overall threat level. Additional display functionality provides the GO with operational awareness of the UA’s state (i.e., location, speed, and orientation), as well as noncooperative intruders’ and cooperative traffic’s geographical positions, speeds, and headings. This display allows the operator to provide information on individual intruders. The GOCon will alert the GO when a system or subsystem failure is detected. When failures are detected, the system will display a red overall threat level, and a failure description will be displayed to the GO.

The GOCon will display the following: a unique indicator for the UA, all aircraft within the surveillance volume (i.e., all cooperative and noncooperative
aircraft, including the respective threat levels of noncooperatives); an overall threat-level status indicating either a red, yellow, or green alert status; the surveillance volume; the active operational transit volume (OTV); the CDSA and restricted areas; and any preset exclusion zones.

Each noncooperative intruder is scored on the basis of its ability to interfere with the UA's transit through the active OTV. This scoring yields an intruder's individual threat level. Noncooperative intruders are displayed on the GOCon display as either a red, yellow, or green target on the basis of that individual intruder's scored threat level. All cooperative aircraft are displayed as blue targets and are not scored or assigned a threat level because they are under the control of ATC.

The overall threat level, displayed as a three-color stoplight on the GOCon display, as shown in Fig. 6, reflects the highest threat level for a noncooperative intruder currently being tracked by the system; this highest threat level is also referred to as the rolled-up threat level. A rolled-up threat level status of red indicates that one or more noncooperative aircraft being tracked could reach the active OTV at their current speeds. A rolled-up threat level status of red can also indicate that there is a critical subsystem failure or that connectivity to the radar is lost. A rolled-up threat level status of yellow indicates that one or more noncooperative aircraft being tracked could reach the active OTV if the aircraft accelerate to 250 knots at an assumed maximum rate of 2 knots per second. A rolled-up threat level status of green indicates that there are no noncooperative aircraft being tracked that can reach the active OTV, even if an intruder aircraft were to accelerate to 250 knots at an assumed maximum acceleration of 2 knots per second.

The GO reports the rolled-up threat level status to the UAC in order for the UAC to make an informed decision on whether to begin transit through the OTV. The UA can begin transit through the OTV only if the rolled-up threat level is green. If the UAC determines that additional airspace information would be important for his or her decision, he or she can ask the GO to provide that information.

### Airspace Architecture

The GBSAA concept at MCAS Cherry Point uses the lateral-transit access profile. Shadow UAS training flights depart from and arrive at MCAS Cherry Point CDSA and fly training missions in either of two adjacent restricted areas, namely R-5306A to the northeast.
The surveillance volume is defined as a volume of airspace in which aircraft are detected and tracked by the surveillance radar and in which the GBSAA system will monitor and display the tracked aircraft. Within the surveillance volume, all noncooperative aircraft are scored using the GBSAA algorithm. Any cooperative or noncooperative aircraft detected within the surveillance volume are defined as intruder aircraft. The GBSAA system monitors traffic within the surveillance volume by displaying data from the surveillance radar system (ASR-11 at MCAS Cherry Point). This volume extends out to 40 NM from the Cherry Point radar, with upper and lower limits that correspond to the typical radar envelope provided by the ASR-11.

An exclusion zone is defined as any area where detected aircraft are not evaluated for threat level. Exclusion zones are identified so as to limit false-positive radar detections within airspace regions that are known to be of no consequence. For Cherry Point, the exclusion zones are not configurable by the GO and include R-5306A, R-5306D, and an area within a 2-NM radius from the center of the Cherry Point CDSA.

Roles and Responsibilities

The GO is a suitably qualified staff member of the ATC facility or VMU-2 and serves as an integral member of the UAS crew. The GO directly reports to the UAC for the duration of the mission. The GO operates the
GOCon any time the UA is preparing for flight or is in flight; the GO's main duties include monitoring aircraft (mainly noncooperative aircraft, which may pose a threat to the UA while it is in the NAS and using the GOCon), ensuring the proper coordination and communications with approach control, and ensuring timely and accurate airspace updates to the UAC. Figure 8 illustrates the organization and communications necessary for the GBSAA system.

The UAC is the lead member of the VMU-2 aircrew. The UAC operates or manages the operator of the UA and is in constant communication with the GO in order to fully understand the airspace. The UAC's main duties include overall responsibility for the operation and safety of the UAS mission, sole authority and responsibility for transit decisions, and ensuring the proper coordination and communications with the GO, tower, approach control, and range control.

ATC handles several functions within the Cherry Point airspace. Approach control provides separation services for the aircraft within the Cherry Point area; it is also located in close proximity to the GO in the radar room. The close proximity of approach control to the GO aids in their coordination during UA operations. The approach controller is aware of the UA operations within his or her airspace but assumes no additional responsibility for the GBSAA and UA activity; the UA is treated the same as any manned aircraft. Approach control's main duties are unchanged from everyday operations and include maintaining situational awareness of UA operations within the assigned airspace, providing radar monitoring of the UA, and providing advisories and safety alerts regarding UA operations to other manned aircraft as prescribed per FAA Order JO 7110.65.

**GBSAA Concept at Cherry Point**

When UA operations are scheduled at Cherry Point, the GO is required to perform preflight checks of the GOCon and prepare for UA operations at least 1 h before launch. When the mission time approaches, the UAC contacts the GO and relays any pertinent launch information. The UAC also contacts the necessary entities within ATC to ensure that they are aware of the UA operations.

Once the UA is launched within the CDSA, the UA proceeds to a holding point and loiters in a pattern until the GO relays a green rolled-up threat level indicating a sanitized airspace. If the rolled-up threat level is red or yellow, there is an aircraft in the area that could reach the UA during its transit through the OTV or there is a system malfunction, and the UA should not transit. The GO will continue to update the UAC on the threat level status until a green status is gained and the airspace is clear of all imposing traffic.

After receiving a green rolled-up threat level status, the UAC makes the decision to begin transiting through the OTV. The UAC has sole authority for the decision to transit and will use all available information and resources to aid in the decision-making process to ensure that the airspace is sufficiently sanitized and that the UA will be able to cross the OTV safely. Once the UA exits the CDSA and enters the OTV, procedures dictate that it should continue on to the restricted area; the UA should not turn around within the OTV. In the event of the rolled-up threat level changing to yellow or red or the system malfunctioning, the UA should still have sufficient time to cross the OTV safely on the basis of the concept's time and buffer allocations. Once the UA exits the CDSA on the rolled-up threat level status of green, the airspace is sufficiently sanitized for 5.2 min. Once it reaches the restricted area, the UA performs its training mission without the aid of the GBSAA system. When the UA is ready to transit back to the CDSA, the GBSAA system is used in the same manner as it was used to get to the restricted area.

**APL's Role**

APL was involved with the Navy/Marine Corps' GBSAA effort from the beginning, before the MCAS Cherry Point was selected as the first GBSAA site. The Navy/
Marine Corps GBSAA team remained relatively small throughout the development phases, and APL played a significant role in systems engineering and analysis. Involvement included initial system design, continued system conceptual design, timeline analysis studies, airspace characterization studies, requirements development and management, architecture development and management, concept of employment development, operations and procedures manual development, full Marine Corps training package development, the conduct of two weeklong training sessions, assistance with safety case development (functional hazard analysis, fault trees, etc.), assistance with drafting the CoA application documentation, defense of the GBSAA system at multiple FAA/OSD forums, and data analysis metrics reports development (initial and continued after the system became operational).

CONCLUSION

As the service lead for GBSAA, Program Manager for Unmanned Aircraft Systems Unmanned Systems Airspace Integration Concepts has led GBSAA activities through the development of an initial system installed at El Mirage, California; the development of a follow-on system for multiple locations and access profiles; and collaboration activities on safety case development, requirements, and software certification methodologies. Their GBSAA development methodology uses an incremental approach for UA access to nonsegregated airspace. The initial system used safe states to ensure that a UA could get out of the airspace before a potential conflict, and the current system is exploring more complex algorithms that would allow a UA to operate in the same airspace as manned aircraft. This incremental approach is not only important for enabling data collection to support the safety case but also for the general acceptance of manned and UA operations in a common airspace.

Currently, the Army activity is progressing with the implementation of a dynamic access profile at Dugway Proving Grounds. The Army is preparing for a test in 2014 to investigate the proposed concept, algorithms, and technology implementation for UASs to dynamically avoid other aircraft by using ground-based sensors.

The Navy’s current system involves the operational implementation of an incremental GBSAA approach for lateral-transit operations. This system is currently in place at MCAS Cherry Point and was awarded a CoA in June 2013 to authorize UAS flights from the airfield to the local restricted areas, transiting through the surrounding Class E airspace.

Unfettered airspace access for UASs is a difficult problem, both technically and politically. Concerns for the safety of manned aircraft have to be balanced with the needs for UAS operations, and the collaboration of all stakeholders is required to come to an acceptable solution. GBSAA is seen as only one step in UASs gaining access to nonsegregated airspace and more general airspace integration. It is envisioned that GBSAA will play a major role in the final airspace integration solution, which will also implement airborne sense-and-avoid technologies.

REFERENCES


The Authors

Thomas P. Spriesterbach is an engineer and the Group Supervisor for the Special Concepts and Engineering Development Group in APL's Force Projection Sector (FPS). He has been working with both the Army and Navy supporting UA airspace integration. Kelly A. Bruns is an aerospace engineer in FPS. In support of the airspace integration effort, she is responsible for providing systems engineering guidance and technical leadership to the GBSAA tasks. Lauren I. Baron is a systems engineer and the Assistant Group Supervisor for FPS's Operational System Analysis Group. Previously, she served as a project manager of the UAS airspace integration project and contributed to the development of the requirements and architecture for the Army and Navy GBSAA systems. She also served on the OSD Sense and Avoid Science and Research Board. Jason E. Sohlke is a systems engineer in FPS. He has served as a system architect and as a project manager of the UAS airspace integration project. For further information on the work reported here, contact Thomas Spriesterbach. His e-mail address is thomas.spriesterbach@jhuapl.edu.

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.
Recent Enhancements to Mobile Bimanual Robotic Teleoperation with Insight Toward Improving Operator Control

Edward W. Tunstel Jr., Kevin C. Wolfe, Michael D. M. Kutzer, Matthew S. Johannes, Christopher Y. Brown, Kapil D. Katyal, Matthew P. Para, and Michael J. Zeher

In a number of environments and scenarios where it is dangerous, impractical, or impossible to send humans, robotic teleoperation is used. It enables human-teleoperated robotic systems to move around and manipulate such environments from a distance, effectively projecting human capabilities into those environments to perform complex tasks. Bimanual robots are equipped with two arms/hands and have the capacity to perform many tasks, particularly when integrated with mobile platforms. This article provides an overview of two mobile bimanual robotic system prototypes designed to be teleoperated by human operators to perform unmanned ground vehicle missions. It highlights limitations in robot sensing and control that were observed during the course of conducting research on improving the effectiveness of human operator control. System enhancements are discussed that are aimed at achieving this goal through robot sensor augmentation, increased operator situational awareness, and a robot control approach for facilitating human operator control. These enhancements are expected to improve human–robot capabilities for future unmanned vehicle applications.

INTRODUCTION

Teleoperation refers to the direct human control of a machine from a distance. It is commonly associated with systems wherein the machine is represented by robotic mechanisms or mobile robots. Such systems are increasingly used on unmanned vehicles to enable a range of applications across a variety of domains including military, first responder, law enforcement, construction, hazardous material handling, and exploration of deep ocean, space, and planetary surface environments. The ability to affect environments through manipulation of robotic arms and attached end-effectors (hands and tools) lets human operators project their intent and capability from safe locations, facilitated by sensor feedback from a teleoperated system at a remote location,
providing a level of telepresence to the human operator. In some cases, the technology is necessitated by remote environments that are inaccessible or inhospitable to humans (e.g., space or planetary surface locations). In other cases, it is preferred to keep humans out of harm’s way by providing them a safe standoff distance [e.g., military explosive ordnance disposal (EOD) missions and hazardous material sites]. To date, many teleoperated unmanned ground vehicles (UGVs) with single robotic arms have been deployed in most of the aforementioned application domains. As tasks become more complex or require additional capability, the demand has increased for UGVs with two robotic arms and added manipulation dexterity. Generally speaking, manipulation tasks requiring the application of torque or leverage tend to fall into this category. Particular examples of tasks for which dual-arm dexterity is required or preferred include lifting objects that exceed the lift capacity or balancing capability of a single arm, unscrewing the cap from a container (or other general screw assembly), operating a shovel or a similar tool, and separating objects (such as a blasting cap from an explosive). Robotic systems employing two arms and two hands are referred to as bimanual manipulation systems, in the same sense of the term as applied to humans; thus, many bimanual coordination tasks that humans can perform may also be performed by a bimanual robot, depending on its dexterity.

Bimanual teleoperation is an enabling technology for dexterous standoff operations, which are often preferred for telerobotic UGV missions that require human-like capabilities in environments or conditions that may put humans in harm’s way. Bimanual teleoperation allows robots to manipulate certain objects and perform complex tasks that are not feasible with either single end-effector control or joystick-like control of two end-effectors. Although the feasibility and underlying capabilities of bimanual manipulation systems have been demonstrated, a number of challenges associated with accurately controlling such complex systems by using inputs from a single teleoperator remain and are discussed herein. This article focuses in particular on the use of bimanual teleoperation in the application domain of EOD and prototype research systems for bimanual teleoperation that comprise a mobile dexterous robotic platform and associated teleoperator/user interfaces. The content is also relevant to similar robotic systems and UGV applications.

After a discussion of related work from the broader research community, the APL research theme of human capabilities projection (HCP) is discussed. An overview of the APL prototype research systems for mobile bimanual teleoperation is presented next. System enhancements to address hardware limitations revealed during recent research at APL, and aimed at improving the ease and effectiveness of operator control, are then described. Finally, we discuss insights gained toward a reduced-order control paradigm to manage complexities associated with human control of the many actuators comprising mobile bimanual teleoperation systems.

**RECENT RELATED RESEARCH**

Fundamental problems of robotics and control theory that underlie telerobotics over communications links with and without appreciable time-delay were addressed decades ago and with considerable vigor during the 1970s and 1980s. Bimanual teleoperation systems were subjects of some research during that time, although most systems involved a single teleoperated robot manipulator. With advances in computing, communications, sensing, and robotics technology, new implementations and enhancements of single and bimanual teleoperated manipulators have emerged and continue to evolve.

Research at APL is addressing ways to improve operator control of such bimanual manipulators during teleoperation, as presented in later sections of this article. For broader context, this section provides a sense of the recent research related to bimanual manipulation, teleoperated or otherwise.

Research continues toward advancing the underlying issues in kinematics and control specific to bimanual robotic manipulators. Results representative of the state of the art include a generic control approach for highly redundant manipulator systems such as bimanual manipulators as well as techniques for coordination and control of bimanual systems (with multi-fingered hands and finger-embedded force sensors) including contact variables (modeled as passive joints) for force control and feedback. Other work advocates development of benchmarks for bimanual robotic manipulation that can be applied for different robots and hands, signifying a sensitivity to the need for evaluating and comparing bimanual systems.

A recent survey paper summarizes the state of the art and developments in bimanual manipulator control, modeling, planning, and learning. Research covering both bimanual manipulation (physical interaction with a single object) and goal-coordinated dual-arm manipulation (arms not physically interacting with each other but solving the same task, e.g., two-handed typing) is included in the survey. The survey projects that future research will address capabilities for performing complex, coordinated dual-arm tasks by building on the integration of these developments with computer vision, learning, and cognition in particular. The survey also projects a substantial increase in applications for collaborative human–robot manipulation. Robot learning of bimanual tasks is among the newest facets of this research topic area wherein a number of techniques are being pursued. An emerging theme is the use of models inspired by research on bimanual coordination.
in human movement science and how tasks can be learned by robots using computational techniques of learning by imitation and programming by demonstration. Such work seeks to address the challenge of robust execution of bimanual tasks by using machine learning techniques, which incurs the additional challenges of getting the motor system to reliably reproduce learned tasks and adapt to external perturbations. Future APL research seeks to address such challenges by building on the mobile bimanual teleoperation infrastructure described in this article.

With increased attention to the use of robots in environments designed for humans, a segment of the community is focused on robot manipulation technology that is effective for human environments, including bimanual systems allowing high-level teleoperation combined with onboard autonomous perception and autonomous control behaviors that can be overridden by a teleoperator. Similar mobile bimanual systems have been considered for performing human-like tasks (e.g., using tools designed for human use) and to assist humans in remote, unstructured environments. A common aim of such work is to reduce the human teleoperator effort or the cognitive load associated with successfully executing bimanual tasks while striking a balance between what humans and robots do best within the context of the task. The ongoing bimanual teleoperation research at APL contributes to these objectives using the bimanual systems described later. Additional challenges include reducing overall task completion time and developing a single human–robot interface that can be used for all modes of robot control from teleoperation to autonomous operations.

More germane to the primary application domain discussed herein, researchers are addressing challenges of teleoperated bimanual systems on UGVs for EOD operations. The research of Kron et al. is representative of work using mobile bimanual teleoperation systems that are similar to platforms used at APL. Their work involves a bimanual manipulator system with haptic (tactile feedback) devices and a head-mounted display as a teleoperator interface that can ultimately be part of a larger system comprising a mobile platform and stereo vision system. The EOD task focus is tele-demining of anti-personnel mines. The research has similar motivations in its focus on bimanual teleoperation for EOD tasks facilitated by multiple modes of sensor feedback, a bimanual human–robot interface, and stereo visualization. More recent work seeks to advance the level of autonomy of dual-arm dexterity for inspection and EOD tasks in urban terrain by using a limb coordination system that can be applied to autonomously control multiple dexterous manipulators for object grasping and manipulation. The limb coordination technology may enable task-specific capabilities such as dual-arm pick up and use of a shovel.

This research also has similar motivations to research pursued at APL in its focus on increased robotic autonomy for EOD missions. With that said, the bimanual systems used for each of these research efforts are less dexterous than the bimanual systems used for research at APL and described later in this article. In particular, they are limited in manipulation dexterity and mobility relative to bimanual systems used at APL, and the research was conducted for a partially integrated system (manipulation only) in one case and in simulation in the other case. As alluded to earlier, a goal of such research, despite the degree of system dexterity, is to enable autonomy within a system that is also applicable to mobile bimanual teleoperation wherein a proper balance is achieved that benefits from the best of human and robot capabilities. To this end, APL research discussed in this article is currently focused on improving human operator control during teleoperation.

**HUMAN CAPABILITIES PROJECTION**

HCP is a research theme at APL that encompasses mobile bimanual teleoperation. Specifically, HCP refers to the robotic manifestation of human-like dexterity and sensory perception through robotic telemanipulation. Our current HCP research is motivated by the notion that telemanipulation effectiveness is enhanced by combining intuitive teleoperator interfaces (based on high-resolution motion tracking of the human operator’s native limbs) to control anthropomorphic robot manipulators with haptic feedback and with stereo remote vision. The key enabler of effective HCP is the robotic system, which acts as a surrogate to the human operator for executing downrange or remote operations. APL focuses its development on core components of the robotic system that are most directly tied to achieving HCP: bimanual dexterous manipulation capabilities, immersive stereo remote vision, intuitive control modalities, sensor integration for operator feedback, platform mobility, and user-in-the-loop semiautonomy.

The framework for HCP research on mobile bimanual teleoperation consists of human–robot elements as well as robot–environment elements, with relationships loosely shown in Fig. 1 for an example EOD application. The human–robot elements are concerned with operator intent via command and control using intuitive teleoperation interfaces and with feedback of information and sensor data from the robot to the operator. Operator intent is realized by using techniques of direct control of each degree of freedom (DOF) of the robot, reduced-order control (achieving comparable control performance considering a control model based on a reduced state-space), and supervised autonomy (commanding and guiding the robot via high-level commands executed using onboard autonomy). Robot–environment elements are concerned with the robot’s sensing of its
internal state (e.g., positions and velocities of its joints and mobile platform) and external local environment, its physical interaction with and modification of the environment via direct manipulation and use of tools, and its mobility through its physical environment.

**PROTOTYPE SYSTEM OVERVIEW**

Two prototype mobile bimanual teleoperation systems have been the focus of recent development at APL. They are referred to as Johnny-T and RoboSally and are variants of several dexterous robotic platform configurations used for HCP research to date. Both robots comprise a dexterous bimanual manipulation subsystem mounted on an all-terrain mobility subsystem. They differ primarily in arm configurations and dexterity for the bimanual manipulation subsystems, being more or less anthropomorphic, while sharing similar four-wheel mobility subsystems, as shown in Fig. 2 and further described in the following subsections. Each robot stands at a height of approximately 1.3 m and has a ground footprint of roughly 1 m by 1.2 m depending on the mobile platform used.

**Dexterous Manipulation Subsystem**

The Johnny-T configuration uses a Highly Dexterous Manipulator (HDM) integrated with wrist-attached end-effectors and a head that houses cameras for stereoscopic vision. The HDM mechanism is a prototype dual-arm system developed as a technology demonstrator for the Advanced EOD Robotic System (AEODRS) program of record. The HDM is a product of HDT

![Figure 2](image-url)  
**Figure 2.** Prototype mobile bimanual robots: (a) Johnny-T; (b) RoboSally.
Robotics, developed with funding from the Joint Ground Robotics Enterprise for the AEODRS program, which is sponsored and led by the Naval Explosive Ordnance Disposal Technology Division.) A “T”-shaped torso with 3 DOFs at the waist (roll, pitch, yaw) and two 7-DOF arms comprise the HDM. Each robotic arm has a fully extended reach of 65 cm (shoulder to wrist) and terminates in an interface to which a variety of robotic hands or end-effectors (having the proper mating interface) can attach. This common interface enables using interchangeable end-effectors designed for various manipulation purposes. To date, several types of end-effectors designed for EOD dexterous manipulation tasks have been used on Johnny-T, including robotic hands from Continuo Robotics (now Telefactor Robotics; see right hand in Fig. 2a), RE2, Inc. (left hand in Fig. 2a), and HDT Robotics; each of these end-effectors add 4 DOFs. The integrated stereo head is a commercial off-the-shelf unit produced by Telefactor Robotics that adds 2 DOFs to the assembly (excluding the zoom capability of its embedded cameras).

The RoboSally manipulation subsystem configuration is an integration of the same HDM torso and stereo head used on Johnny-T, along with more dexterous manipulators in the form of two Modular Prosthetic Limbs (MPLs) developed by a multi-institutional team led by APL as part of the Defense Advanced Research Projects Agency Revolutionizing Prosthetics program. Each MPL is highly anthropomorphic in form factor and weight relative to the human arm, providing 7 DOFs from shoulder to wrist plus 10 DOFs in its five-fingered anthropomorphic robotic hand, resulting in 26 articulating joints in total (4 DOF of which are in the thumb for dexterous grasping and hand shaping). With its many DOFs, along with fingertip-embedded sensors for tactile perception and haptic feedback, the overall MPL design allows for demonstration of human-like dexterity and feedback to a human teleoperator.

Mobility Subsystem

The ground mobility subsystem transports the bimanual dexterous manipulation subsystem throughout the operational environment to sites for performing manipulation tasks. The mobility subsystem for the Johnny-T and RoboSally configurations is a high-mobility, heavy-payload UGV available commercially from Synbotics—two model varieties from its Ground Robotic Platform (GRP) Series have been used to date, namely the GRP 2400x and the GRP 4400x models. These fully electric UGV models feature four wheels with independent drive motors and independent passive suspension as well as active four-wheel steering. The GRP 2400x model is capable of four steering modes (crab, rear-only, front-only, and skid steer for turns in place) wherein the front and/or rear wheels are oriented at the same angle within the steering range of motion. The GRP 4400x is capable of the same steering modes plus an additional steering mode wherein all wheels are maximally oriented toward the central longitudinal axis of the UGV, enabling a four-wheel turn-in-place capability. With multiple steering modes and ground clearance of approximately 20 cm and 45° slope traversability (without payload), the mobility subsystem provides high maneuverability on various ground surfaces, from indoor floors to paved roads and natural off-road and rough terrain. Such high maneuverability is essential for accessing varied terrain to bring the next work site within reach of the bimanual dexterous manipulation subsystem.

Teleoperation Interfaces

Controllable DOFs of both mobile bimanual teleoperation system prototypes can be teleoperated via a variety of user interfaces and devices. Provided with visual imagery from cameras in the robot’s stereoscopic head, and in some cases haptic feedback from sensors on the robot’s manipulators, a human operator can teleoperate the robot to perform tasks by using any of the HCP control modes mentioned above. To achieve this, appropriate interfaces are required to deliver visual and haptic feedback to the operator and to deliver motion control inputs to the robot, all over a wireless communications link. An operator of the Johnny-T or RoboSally system wears a head-mounted display to receive stereo views of the robot’s local environment as seen by cameras in the robot’s stereoscopic head and transmitted over the communications link. This provides the operator with a degree of visual immersion or telepresence in the remote environment, which facilitates effectively executing teleoperated tasks. Teleoperation of the mobility subsystem for both robots is achieved by using a separate remote control device enabling the operator to drive and steer the platform between manipulation work sites.

To achieve bimanual teleoperation, human operator motions expressing intended motions of the robot’s DOFs must be transmitted to the robot. Likewise, to benefit from haptic feedback, the forces/torques or physical contact measured or detected by sensors on the robot manipulators must be interpreted and transmitted to the human operator via suitable haptic devices. Two sets of different teleoperation interface types have been used thus far to deliver motion control inputs to the robots and haptic feedback to the operator. These sets are referred to herein as human-joint mimicry (HJM) interfaces and joystick endpoint control (JEC) interfaces. As noted earlier, our HCP research advocates the use of intuitive teleoperator interfaces based on high-resolution motion tracking of the human operator’s native limbs to achieve enhanced telemanipulation effectiveness. HJM inter-
HJM interfaces have been used for both the Johnny-T and RoboSally systems. A second approach to bimanual teleoperation uses advanced multi-DOF joystick-like devices to provide motion control inputs to the robotic manipulators and to apply forces/torques to the operator's hand(s), representing haptic feedback received from the robot. JEC interfaces embody this approach with Harris RedHawk intuitive haptic controllers (Fig. 3). The operator's left and right hands each manage a single haptic control.

User interface

3-D video

Processing

Mobile bimanual robotic system

Figure 3. The Harris RedHawk dual controller.

Figure 4. Overview of the various system components, connections, and signals. IMU, inertial measurement unit.
pler to provide motion inputs to the robot for positioning its respective left and right arms. Each haptic controller can control 6 DOFs as well as end-effector grip motions. Hand-controlled motions are mapped to the controllable DOFs on the robot and transmitted to the robot for execution, thus mimicking the intended motions of the human operator. The operator receives interpreted feedback of forces/torques experienced by the robot’s manipulators via the haptic hand controllers for the six controllable DOFs and the grip control. JEC interfaces have been used for the Johnny-T system. Figure 4 depicts a high-level block diagram of the overall mobile bimanual teleoperation system.

HARDWARE LIMITATIONS AND ENHANCEMENTS

Through the course of conducting recent research at APL with the types of bimanual teleoperation systems described herein, as well as particular robot hardware components associated with them, we have observed limitations in the achievable quality of manipulator state estimates and haptic feedback as well as a need to improve visibility for reaching and grasping occluded objects. The robot’s onboard estimates of its bimanual manipulator joint positions, and thus the overall posture of the manipulation subsystem, are important for accurately tracking operator motions under direct teleoperation control. The same is true for accurately executing any automatic or autonomous motions under reduced-order or supervised (operator-assisted) autonomous control. Uncertainty in manipulator joint positions associated with the actuators used in our work has been found to be a limiting factor in bimanual manipulator state estimation. More specifically, backlash in the drive train of the manipulator actuators, coupled with errors in encoder alignment and motor control, causes a divergence of the estimated state from the true state as the system is operating. This divergence is a direct consequence of actuator design, which in the MPL and HDM systems has favored high speed and torque in a lightweight and low-profile package traded for ultimate precision. Although an experienced operator may be able to compensate for such divergence during direct teleoperation, the need to do so unnecessarily adds to the physical and cognitive load associated with operating the system. Without knowledge of bimanual manipulator state via precise sensor measurements or adequate state estimates, robust implementation of automatic control functions (e.g., gravity compensation) and autonomous control behaviors (e.g., sensor-based avoidance of self-collision and designated keep-out regions) would clearly be a major challenge.

Haptic feedback in the form of forces, torques, and contact indicators provides information about the physical nature of the robot’s interactions with its environment and enhances the operator’s telepresence. The quality of such feedback affects the robot’s ability to successfully execute teleoperated tasks. Torque feedback from the bimanual manipulators described herein is currently derived from dedicated strain gauges within the joints of the upper arms. However, the location of these sensors and the drivetrain friction inherent in the joints can lead to inaccurate readings, limiting the quality of haptic information fed back to the operator. It is anticipated that sufficient experimental calibration could serve to overcome some of these issues. With that said, certain limitations would remain as impediments to accomplishing tasks such as manipulating objects occluded from operator view. For example, certain EOD tasks require reaching into cavities or behind obstructions to retrieve objects. Although a rich set of haptic information may enable an operator to explore within the occluded area and “feel” for the object, our current prototype configurations do not provide an adequate level of haptic feedback.

The authors are considering a variety of design and implementation enhancements for the bimanual systems described herein to address their limitations and improve manipulator state estimation, haptic feedback, or both. One approach would incorporate a dual-encoder system, adding a high-count encoder before and after gear trains. In this way, the amount of divergence in joint angle positions from the true states can be measured and used to estimate joint torques. This approach has been designed into recent industrial robot manipulators available on the commercial market. The drive torque measurement challenge can be addressed using other specially designed sensors or intelligent control techniques.17,18 Another approach to enhancing state estimation as well as contact detection for haptic feedback would integrate accelerometers into each link of the bimanual manipulator. Low-frequency portions of the signal provide additional information that can be used to improve the accuracy of the system’s state estimation. High-frequency response can be used to provide the user with feedback related to sudden contact with objects or obstacles.19,20 Accelerometer elements in the fingertip sensors of the MPL system and on the limb controller of the MPL in the RoboSally configuration provide this frequency response.

To enhance the operator’s visibility for reaching and grasping occluded objects, the sensor suite for the right hand on the Johnny-T bimanual manipulator was augmented with a lidar sensor (Hokuyo URG-04LX-UG01) and auxiliary camera with LED illumination. These sensors were located on the end-effector and provided visibility during the end-effector’s approach to and within occluded areas, thus compensating for limited haptic feedback from the hand and facilitating guidance during the reach and grasp actions. Figure 5 illustrates the position and orientation of the lidar sensor and the type of planar distance information that it provides. As shown
mounted lidar, camera, and illuminator is expected to increase efficiency of the reach and grasp task. Such enhancements to manipulator state estimation, haptic feedback, and visibility for grasping occluded objects increase teleoperation effectiveness and capabilities for the Johnny-T and RoboSally systems.

REDUCED-ORDER MODELING CONSIDERATIONS

While the hardware enhancements identified in the preceding section can improve the controllability of the bimanual manipulation hardware, enhancements are also considered for improving the operator's capability to teleoperate the overall system. In this regard, the aim is to manage the complexity of controlling the many DOFs of highly dexterous robotic platforms via human interfaces limited to control of fewer DOFs. A related aim is to reduce the burden on the human operator by obviating the need for direct, serial control of each DOF. A further enhancement that can address such issues is the application of reduced-order models of highly dexterous robot manipulators.

Considering the number of controllable DOFs for the Johnny-T and RoboSally systems and their associated observable states, the fundamental complexity involved is due to an extensive control space (29–42 independent active joints and 27–74 available sources of sensor feedback as indicated in Figs. 7 and 8, exclusive of the aforementioned end-effector camera plus lidar enhancement). APL research experience to date has identified a number of problems with accurately controlling such complex systems by using inputs from a single human operator's limbs or common handheld operator control units. These insights have led to consideration of techniques.

![Figure 5. Illustration of end-effector-mounted lidar sensor detection profile intersecting a cylindrical object to be grasped.](image)

in Fig. 6, this information is provided to the operator through an augmented view window within the operator's head-mounted display, showing the hand–camera view and lidar range data. In Fig. 6, the lidar range data view indicates the detected presence of an object inside an enclosure reached into using the right hand of Johnny-T. With this approach, visual information presented to the operator enables the operator to see within the occluded area, locate an object within, and consider object orientation (and other obstructions) in deciding how to grasp it directly or otherwise manipulate it to ensure an effective grasp. This is an improvement on limited haptic feedback, which would at best allow the operator to blindly "feel" around within the occluded area while inferring information about object presence, location, orientation, and spatial relationship to other obstructions within the occluded area. The gain in increased situational awareness provided by the hand-

![Figure 6. Operator head-mounted display during a teleoperated reach into an enclosure to grasp an occluded object, showing hand inserted into an opening in the enclosure (left), bimanual manipulator kinematic posture (top right), hand–lidar range data (middle right), and hand–camera view (bottom right).](image)
to control our systems on the basis of reduced-order models as a means to manage bimanual teleoperation complexities as well as the physical and cognitive burden associated with the teleoperation experience.

The mobility subsystem is already teleoperated in a reduced-order manner in that only 2 DOFs (i.e., the gross UGV direction and speed) are controlled by the human operator as opposed to additional DOFs for each individual wheel speed and steering control. Similarly, a reduction in operator-controlled DOFs is desired for bimanual teleoperation of the dexterous manipulation subsystem considered in combination with the mobility subsystem. This is achieved by using the redundancy in the system and the associated null space of the Jacobian matrix\(^2\) to optimize secondary aspects of the system while achieving the desired goals (Wolfe, K., Kutzer, M., and Tunstel, E., “Leveraging Torso and Manipulator Redundancy to Improve Bimanual Telepresence,” unpublished manuscript, 2013). Note that, for this system, the dimension of the Jacobian matrix is

\[ \text{Right arm} \]

- 7 joint positions

\[ \text{Neck} \]

- 2 joint positions (pan-tilt style)
- 3 joint positions (anthropomorphic)

\[ \text{Right hand} \]

- 10 joint positions (MPL)
- 4 joint positions (non-MPL hands)

\[ \text{Left arm} \]

- 7 joint positions

\[ \text{Left hand} \]

- 10 joint positions (MPL)
- 4 joint positions (non-MPL hands)

\[ \text{Torso} \]

- 3 joint positions

\[ \text{Mobile base} \]

- 1 steering direction
- 1 speed

\[ \text{Human operator} \]

\[ \text{Head} \]

- 2 independent camera feeds

\[ \text{Right arm} \]

- 7 joint torques

\[ \text{Right hand} \]

- 10 joint torques (MPL)
- 10 discrete contact sensors (MPL)
- 3-direction contact sensing/finger (MPL)
- 3-axis acceleration sensing/finger (MPL)
- 1 temperature sensor/finger (MPL)
- 4 joint torques (non-MPL hands)

\[ \text{Left hand} \]

- 10 joint torques (MPL)
- 10 discrete contact sensors (MPL)
- 3-direction contact sensing/finger (MPL)
- 3-axis acceleration sensing/finger (MPL)
- 1 temperature sensor/finger (MPL)
- 4 joint torques (non-MPL hands)

\[ \text{Left arm} \]

- 7 joint torques

\[ \text{Torso} \]

- 3 joint torques

\[ \text{Figure 7.} \] Control state-space breakdown assuming 1:1 mapping between user and system.

\[ \text{Figure 8.} \] Sensory feedback options for hardware variants of the dexterous manipulation subsystem.
ENHANCEMENTS TO MOBILE BIMANUAL ROBOTIC TELEOPERATION

CONCLUSIONS

Enabling technologies for mobile bimanual teleoperation and improvements on existing approaches are impacting teleoperated UGV systems and their operational techniques. As advances continue to occur in parallel, it is important to reassess the synthesis of integrated system solutions and component technologies to achieve desired effects for increasingly complex UGV tasks. With an aim toward advancing the effect of HCP for dexterous standoff operations, we have concentrated on improving the effectiveness of the human operator by using intuitive teleoperation interfaces to control many robotic DOFs.

Recent enhancements described herein aim to address limitations encountered through teleoperation experience at APL with two high-DOF prototype mobile bimanual robotic systems. The use of high-count dual-encoders and the integration of accelerometers into all links of the bimanual manipulator are considered for improving the quality of manipulator state estimates and haptic feedback. The integration of optical ranging, imaging, and illumination sensors into robot end-effectors, with associated presentation of the sensor data in augmented head-mounted display views, has proven effective for improving the capability to reach for and grasp objects otherwise occluded from the operator’s view of the remote scene. Techniques for controlling the many DOFs of mobile bimanual systems via reduced-order modeling offer an effective and more manageable teleoperation approach and experience for the operator than direct serial teleoperation of each DOF.

Incorporating such enhancements into existing and future mobile bimanual teleoperation systems is expected to improve the human–robot capabilities for executing complex and increasingly dexterous tasks associated with future UGV applications. The expected impact is increased utility and deployment of highly dexterous UGVs for military, first responder, law enforcement, construction, hazardous material handling, and exploration missions.

REFERENCES


The Authors

All authors are Senior Professional Staff members in APL’s Research and Exploratory Development Department (REDD) working within its Robotics and Autonomous Systems Group. Edward W. Tunstel Jr. contributed to research and technology assessment for bimanual robotic systems and dexterous manipulation. Kevin C. Wolfe developed and implemented control strategies and algorithms for bimanual teleoperation of the Johnny-T system. Michael D. M. Kutzer served as co-Principal Investigator of the Independent Research and Development (IR&D) project on bimanual operations for HCP and served as the lead for efforts relating to the use of reduced-order control in bimanual teleoperation. Matthew S. Johannes is Project Manager for the MPL system. He served as co-Principal Investigator of IR&D projects on bimanual teleoperation control for dexterous robotic platforms and bimanual operations for HCP. Christopher Y. Brown supported system integration and preliminary control for both the RoboSally and Johnny-T systems. Kapil D. Katyal is the Project Manager for autonomous controls of the MPL system. He worked on integration of the MPL and vision system for the dexterous robotic platform. Matthew P. Para is the Controls Lead on the Revolutionizing Prosthetics program. He worked on integration of the MPL systems and the mobile robotic platform as well as overall testing and control of the system. Michael J. Zeher is the Robotics Section Supervisor in the Robotics and Autonomous Systems Group. He served as Principal Investigator of the IR&D project on bimanual operations for HCP. For further information on the work reported here, contact Dr. Kutzer or Dr. Johannes. Their e-mail addresses are michael.kutzer@jhuapl.edu and matthew.johannes@jhuapl.edu, respectively.

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.
Integration of Advanced Explosive Ordnance Disposal Robotic Systems Within a Modular Open Systems Architecture

Mark A. Hinton, James M. Burck, Kristine R. Collins, Matthew S. Johannes, Edward W. Tunstel Jr., and Michael J. Zeher

The Advanced Explosive Ordnance Disposal Robotic System (AEODRS) is a Navy-sponsored acquisition program developing a new generation of open, modular explosive ordnance disposal (EOD) robotic systems. The program has developed a common architecture for a family of systems. The foundation of that common architecture is the careful partitioning of EOD robotic systems into capability modules and the definition of intermodule interfaces based on recognized and accepted open standards. This partitioning facilitates module-level interoperability for more rapid technology insertion and removes interface incompatibilities that are barriers to reusing modules among members of the family of systems. In this article, we discuss our experience with the integration and testing of a modular EOD robotic system based on an open systems architecture for the AEODRS family of systems. We describe a phased approach to module and system testing focused first on verification of module compliance with interface and performance specifications and subsequently on system performance and operational reliability. Finally, we share lessons learned in standards assessment and open systems architecture specification through the specification, integration, and testing of multiple independently developed capability modules.

INTRODUCTION

During the past decade, unmanned ground vehicle (UGV) systems have been used successfully by military services and are increasingly being used to conduct dangerous and life-threatening tasks for law enforcement as well as first responder and disaster relief efforts around the world. As the application of UGV systems becomes more pervasive, their deployment and life cycle logistics support become major concerns. As UGV missions call for execution of more complex tasks, it becomes necessary to integrate advanced technolo-
The AEODRS program conducted a study of EOD mission types. This study, conducted by Penn State's Applied Research Laboratory, concluded that the EOD mission space could best be addressed by a family of three classes of UGV systems based on a common architecture that would enable maximizing reuse of components between the family of systems (FoS) members to reduce the cost of supporting the FoS in theater. The FoS resulting from this analysis comprises the dismounted operations system, the tactical operations system, and the base/infrastructure operations system.

AEODRS is a Joint Service Explosive Ordnance Disposal (JSEOD) program, executed through the Naval Explosive Ordnance Disposal Technology Division (NAVEODTECHDIV) via the Navy Program Management Office for Explosive Ordnance Disposal/Counter Remote Controlled Improvised Explosive Device Electronic Warfare (PMS-408). The foundation of the AEODRS common architecture is the careful partitioning of explosive ordnance disposal (EOD) robotic systems into capability modules (CMs; subsystems serving specific functions within the vehicle architecture) and the definition of common intermodule interfaces at the physical, electrical, and logical levels, all based on recognized and accepted open standards. The parallel concept of an open systems business model, in which the government owns and controls the architecture definition and the specifications of the intermodule interfaces, is critical to understanding and successful implementation of such a modular, open system. In a previous Johns Hopkins APL Technical Digest article, we introduced the common architecture and described an implementation approach that would demonstrate its contribution to subsystem and payload interoperability. We presented a strategy for incremental integration and testing of independently developed subsystems and payloads leveraging a mixed-simulation system test bed to enable independent assessment of their architectural compliance. This incremental integration and test strategy also reduces integration schedule dependencies on the order in which the independently developed subsystems and payloads are delivered for integration.

This article focuses on the experience gained thus far and the lessons learned while assessing architectural compliance and performing the integration of independently developed CMs delivered by multiple vendors.

THE AEODRS FAMILY OF SYSTEMS

The AEODRS program conducted a study of EOD mission types. This study, conducted by Penn State's Applied Research Laboratory, concluded that the EOD mission space could best be addressed by a family of three classes of UGV systems based on a common architecture that would enable maximizing reuse of components between the family of systems (FoS) members to reduce the cost of supporting the FoS in theater. The FoS resulting from this analysis comprises the dismounted operations system, the tactical operations system, and the base/infrastructure operations system.
The dismounted operations system is the first AEODRS system to be fielded and is the smallest UGV in the FoS. It must be able to fit fully into a backpack, which places a premium on size and weight. It includes a compact, lightweight UGV and a lightweight handheld operator control unit (OCU). It is intended to focus on reconnaissance tasks but is also capable of supporting the placement of countercharges to disrupt a device. A photograph of the dismounted UGV is shown in Fig. 1; Fig. 2 provides a view of the UGV showing its constituent CMs.

The tactical operations system is the second AEODRS system; its primary mission focus is in-depth reconnaissance and wide-range item prosecution. It is a medium-sized system that must be able to be transported in a vehicle, and it must be capable of being carried by two technicians over a moderate distance. The base/infrastructure System is the third AEODRS system and is the largest variant of the FoS. It requires transportation via a response vehicle or trailer and has a primary mission focus on tasks requiring heavy load and lift capabilities and the widest range of neutralization, render-safe, and other special capabilities. The tactical operations and base/infrastructure systems each include a larger, portable OCU that supports their increased functionality. In addition, the basic UGV functionality of each system can be controlled by the lightweight handheld OCU of the smaller dismounted operations system.

**AEODRS COMMON ARCHITECTURE OVERVIEW**

CMs for the AEODRS system derive from a decomposition of key capabilities identified by the EOD community as important for AEODRS UGVs. The CMs implemented for the dismounted operations system UGV are partitioned and interconnected as illustrated in Fig. 3 and listed with brief descriptions in Table 1.

The AEODRS common architecture is a distributed architecture, as suggested by the module partitioning. In this architecture, the system consists of two primary sub-systems: an OCU and a UGV. The UGV subsystem is itself a distributed system consisting of a set of intercommunicating CMs connected by a single network. This on-vehicle network, referred to as the intrasubsystem network, is separate and distinct from the intersubsystem network linking the OCU subsystem and the UGV subsystem. The routing of messages between the two networks is one of the primary tasks of the master CM.

The logical architecture builds on the Joint Architecture for Unmanned Systems (JAUS) standard, which specifies transport, protocols, and messages to be used in the control of unmanned systems. JAUS is now a Society of Automotive Engineers (SAE) standard supported by a suite of SAE specification and guidelines documents. The JAUS standard has been successfully demonstrated in multiple advanced prototypes tested in operational scenarios under realistic conditions.
The AEODRS program developed a documentation set that provides architectural descriptions, performance specifications, and interface specifications for the physical, electrical, and logical interfaces of each module. Concurrent with the development of the architecture, and the document set expressing the architecture, the program developed an integration and testing strategy intended to detect, mitigate, and manage critical risks associated with system architecture implementation.

### COMMON ARCHITECTURE IMPLEMENTATION

The AEODRS document set was provided to members of industry to support the development, design, and construction of prototype CMs for a proof-of-concept prototype UGV system satisfying the requirements of the dismounted operations system. Allocating all modules to a single vendor weakens the architectural demonstration, in that it is more likely that a single performer will make consistent assumptions when ambiguities exist within the document set than that multiple unrelated organizations will do so. Allocating modules to different vendors increases the potential for identifying and resolving any ambiguous, incomplete, or conflicting specifications. Maintaining close communication with each vendor during its CM development provided the APL team, in its system integration role, with further opportunity to identify and eliminate latent ambiguities from the AEODRS document set.

Each of the selected vendors independently developed and delivered at least one of the CMs listed in Table 1. Although the CMs were independently developed to comply with the AEODRS document set, each vendor used its preferred JAUS framework to meet architectural requirements and the unique functionality for its contracted CM. The various JAUS frameworks used by the vendors include the JAUS Tool Set (JTS), OpenJAUS, JAUS++, and vendor-specific frameworks such as mjJAUS and micro-JAUS. The vendors’ use of different frameworks, different processor architectures, and different development toolchains and styles provided a test of the completeness of the logical interface definitions and the architecture description.

### Incremental Integration and Test Concept

Early in the program, a simplified simulation of the system was constructed and used to build confidence in the open system architecture approach for AEODRS. This simulation test bed, known as the “architecture test bed,” leveraged an existing EOD UGV training simulator to provide a physics-based simulation capability. This simulation capability was then used by a family of “surrogate CMs,” each of which substituted for one CM of the FoS by providing an AEODRS-compliant interface for that CM and interfacing to the modified training simulator for physics-based simulation of the CM’s behavior. Thus, each surrogate CM provided a standards-compliant facade referred to in AEODRS documentation as an AEODRS adaptor. This concept was carried forward into the development of a mixed-simulation system test bed for incremental integration and testing.

Although we made our integration plans on the basis of a specific order of CM delivery from the vendors, we realized that it was unlikely that the CMs would become available for integration and testing in the order originally planned. As mitigation, the APL team, as the system integrator, developed an integration approach

<table>
<thead>
<tr>
<th>CM Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
</tr>
<tr>
<td>Communications</td>
</tr>
<tr>
<td>Master</td>
</tr>
<tr>
<td>Mobility</td>
</tr>
<tr>
<td>Manipulator</td>
</tr>
<tr>
<td>End-effector</td>
</tr>
<tr>
<td>Visual sensors</td>
</tr>
<tr>
<td>Autonomous behavior</td>
</tr>
</tbody>
</table>
that relied on the use of simulation in the early stages. This incremental integration approach relied on the use of the system test bed and its simulation-based surrogates for each CM defined in the AEODRS system. The modular simulation support provided by the system test bed enabled us to substitute CM simulations for CMs not yet available for integration. The use of this mixed-simulation environment for integration relaxed our program dependence on a given fixed sequence of module delivery. This proved to be valuable to the program because the sequence in which CM prototypes were delivered differed significantly from the planned delivery sequence. The use of our mixed-simulation system test bed also allowed us to test a delivered prototype CM in a system composed of previously integrated CM surrogates, enabling us to examine the behavior of a prototype CM in a previously tested configuration. This in turn enabled us to pursue incremental (stepwise) module integration, controlling the scope of each integration increment and maintaining a controlled integration environment.

**SYSTEM TEST BED**

In the following paragraphs, we discuss the construction of the mixed-simulation system test bed in further detail.

**Basic Configuration**

The system test bed used COTS computer hardware as the platform for the OCU subsystem. This “desktop OCU” was used to host the OCU software developed by Space and Naval Warfare Systems Command (SPAWAR) Systems Center as a variant of SPAWAR’s Multi-Robot OCU (MOCU).

Figure 4 illustrates a block diagram of the AEODRS system test bed depicting the interconnection of the surrogate CMs and the MOCU-based OCU.

The OCU connects via the intersubsystem network to the intersubsystem network port of the master CM; the intersubsystem network is presently represented in the test bed by an Ethernet switch, as shown in Fig. 4. The test bed provides a functional surrogate for the final master CM; the surrogate is implemented on an embedded PC (test bed node 1) supporting two network interfaces (for the OCU-facing intersubsystem network and the intrasubsystem network onboard the UGV). The functionality provided by the surrogate is as described for the master CM. In addition to replicating the features of a given CM, a surrogate CM can also incorporate optional fault-injection and other testing features useful in system evaluation but not necessary or not desired in the actual CM. In this spirit, the surrogate master was used as a convenient point at which to introduce communications delay and to corrupt packet data to evaluate...
the impact of latency and packet loss on system performance. As with the intersubsystem network, external connection to the intrasubsystem network was implemented via an Ethernet switch in the test bed.

The remaining CM surrogates serve as AEODRS adaptor front ends for the simulation engine, these surrogates run on a second embedded PC (test bed node 2) distinct from the computer running the master CM surrogate. Each of these surrogates supports the AEODRS logical system interfaces specified for the CM they represent. As AEODRS adaptors, the surrogates provide an interface to the simulation engine.

The simulation engine provides a physics-based simulation of the UGV EOD system within virtual spatial environments representative of common EOD missions. The simulator engine is based on an operator training system (the EOD Robot Training Simulator, or ERTS) developed for NAVOEODTECHDIV by a team at Battelle Memorial Institute. The simulator accepts and executes commands, updating status and representative video. The video simulation is provided via AEODRS messaging compliant with the visual sensor CM interfaces.

Each surrogate CM supports and participates in the AEODRS discovery process (detection, identification, registration, and publication of services). Thus, if a given CM surrogate is not started, its services will not be published in the discovery registration tables. This provides the flexibility the system test bed requires to support substitution of simulated CMs for physical CM realizations (and vice versa), which enables the test bed to support the AEODRS incremental integration and testing approach.

Figure 5 is a photograph of one of the early system test beds in APL's AEODRS Lab. The photograph is labeled with the identification of each node as shown in Fig. 4. These labeled nodes show the COTS computers running the OCU software (MOCU) and the simulation engine (ERTS), as well as a laptop hosting the surrogate CMs (CM-server), a laptop hosting the surrogate master CM (CM-MAS), and a laptop hosting the surrogate autonomous behaviors CM (CM-AB).

**Incremental Integration and Testing Implementation**

Initial system testing used a full-simulation configuration of the system test bed in which simulations were employed for all CMs. This enabled the team to examine the expected interactions between CMs and the observed behavior of the system prior to the availability of any hardware CMs. A conceptual diagram of this configuration appears in Fig. 6.

Before the availability of any prototype CMs, the system test bed consisted of simulation-based surrogate CMs, as shown in Fig. 6. This enabled testing and integration of the surrogate CMs.

The first hardware CM realization to be incorporated was the master CM (CM-MAS). The master CM simulation was disabled, and the hardware CM-MAS connected into the test bed using the standard CM interfaces. This configuration is depicted in Fig. 7.

This stepwise integration process was continued, replacing simulated CM capabilities with hardware CMs as those CMs were delivered by the vendors. The endpoint of stepwise integration was a system in which each CM surrogate had been replaced by its corresponding CM prototype unit. During the stepwise integration process, the simulation-based system test bed became a mixed-simulation test bed, with the endpoint being the fully integrated prototype UGV.

When a prototype CM was received for integration and testing, the CM was first subjected to a unit test regime, which exercised its interfaces for logical architecture compliance, followed by functional testing in a system context with known and tested surrogates of all other CMs. Once the prototype CM under test demonstrated implementation of conforming interfaces and basic expected module functionality, the behavior of the prototype CM was evaluated in the system test bed using surrogate CMs for the other CMs in the system. After successful integration with the surrogate CMs, the prototype was then integrated with other previously tested prototype CMs, still within the system test bed environment. The team was able to evaluate the interac-
interface testing and initial integration testing in house before shipping their prototype CM to APL for integration and testing. We embraced their desire and provided each vendor with test bed documentation and software and testing scripts that enabled each vendor to stand up their own system test bed and perform initial testing. We found that this reduced the number of initial test failures encountered by the APL integration team.

We believe that the ability to respond to vendor requests for in-house initial testing and integration support further demonstrates the flexibility of COTS-based mixed-simulation test bed support for testing and integration.

Analysis Tools

The intersubsystem network switch and the intrasubsystem network switch provide means by which additional laptop computers (labeled “Analysis laptop access” in Fig. 4) may be attached to perform packet capture and message stream analysis on both networks.
To facilitate unit testing, we developed a stimulus/response software package. Running on the OCU platform, this package incorporated a script language for expressing sequences of messages to be sent to, and the expected responses from, the CM under test. The script language, based on XML, also provided means to specify the timing between stimulus messages and timeouts associated with response receipt. The script engine constructed messages and expected responses based on human-readable script files written and maintained by our test team; the script engine relied on the publicly available JSIDL definitions of the messages; thus, adding new messages or updating message support required only updating or adding JSIDL message definitions to the JSIDL message definition files used by the script engine. We did not need to update the script engine software to incorporate updates or extensions to the message set.

We found that these tools enabled us to test more efficiently, reducing the tedium of testing and enabling as part of compliance assessment and performance measurement. Analysis of packets captured on the intrasubsystem network supported logical interface compliance testing as described above; packets captured on the intersubsystem network supported analysis of OCU logical interface message compliance. The use of a common time base for capture of packets from both networks enabled us to assess latency incurred in the communications link and the master CM message routing. We used the popular Wireshark network protocol analyzer tool (http://www.wireshark.org) for this purpose, along with a Wireshark plug-in designed to dissect JAUS messages. The JAUS dissector “dissects” (parses and displays) JAUS messages. The JAUS standard documents define services and message content formally, using the JAUS Service Interface Definition Language (JSIDL); this enabled the JAUS community to develop a Wireshark dissector plug-in whose parser obtains its knowledge of message content by reading the JSIDL provided with the standards documents.

Figure 7. System test bed with CM-MAS hardware.

To facilitate unit testing, we developed a stimulus/response software package. Running on the OCU platform, this package incorporated a script language for expressing sequences of messages to be sent to, and the expected responses from, the CM under test. The script language, based on XML, also provided means to specify the timing between stimulus messages and timeouts associated with response receipt. The script engine constructed messages and expected responses based on human-readable script files written and maintained by our test team; the script engine relied on the publicly available JSIDL definitions of the messages; thus, adding new messages or updating message support required only updating or adding JSIDL message definitions to the JSIDL message definition files used by the script engine. We did not need to update the script engine software to incorporate updates or extensions to the message set.

We found that these tools enabled us to test more efficiently, reducing the tedium of testing and enabling
us to focus more effectively on the testing of the system, less distracted by the “care and feeding” of our tools.

LESSONS LEARNED

The pre-prototype integration exercise has provided feedback and refinement for the architecture, its interface definitions, and the associated documentation. In the following paragraphs, we share a few of the broader lessons, which should be applicable to other projects seeking to develop and implement a modular open systems architecture.

1. Our pre-prototype strategy involved subcontracting the development of the individual CMs to multiple independent vendors. We felt that ambiguities in specifications are often bridged by the reader’s assumptions and that a single developer would likely make consistent assumptions, with the result that ambiguities would not be exposed during prototype development. For the best verification of interface specification completeness, we felt it best that the prototype modules be implemented on at least two different processor architecture platforms using at least two different JAUS frameworks and software development toolchains. In our system integration efforts, we uncovered several minor incompatibilities between the JAUS frameworks our vendors used. Early CM unit testing within the mixed-simulation test bed enabled us to quickly identify the incompatibilities. In all cases, the incompatibilities were avoidable and could be mitigated by providing guidance in the use of framework features not required for successful implementation.

2. Early exposure contributed to identifying issues with the DHCP client configuration for the intersubsystem network interface of CM-MAS. Insufficient client configuration guidance had been provided in the interface control document for CM-MAS. After initial testing at APL, guidance was added to the document and provided to the vendor. Similar issues were identified and resolved with respect to DHCP lease parameters for the DHCP intersubsystem network server (provided by the MOCU subsystem) and the DHCP intrasubsystem network server (provided by CM-MAS). Again, the critical observation is that such issues are readily resolved when identified early; early exposure, both by document reviews and by early testing facilitated by the mixed-simulation test bed, is helpful.

3. Successful unit testing of each CM (including the exercise of interfaces, and specifically the exercise of all specified messages) before integration of that CM reduced system integration. Thorough unit testing in the controlled test bed environment minimizes the number of issues encountered during system integration. Providing a system test bed that supports testing of an individual CM in isolation from other vendor CMs is valuable.

4. Early testing uncovered omissions in the early architecture definition document for the AEODRS common architecture (ADD-CA). Through this testing, we identified weaknesses in the JAUS discovery process and added guidance to the ADD-CA to assure robust, complete registration of services. Independent CM development resulting in independent discovery implementations, followed by careful system integration testing and analysis, made it possible to isolate these issues and track them to the root cause in a timely manner.

5. Despite document review efforts by APL and our partners in the AEODRS system development and integration team (SDIT), inconsistencies in electrical interface definitions remained in the initial release of the document set. Interface inconsistencies are possible, so it is important to exercise all interfaces—logical, electrical, and physical—during unit testing and early system integration testing.

6. Our system test bed configuration implemented the intersubsystem network and the intrasubsystem network using simple managed Ethernet switches. This enabled us to use port mirroring for packet capture and logging. This, in turn, enabled us to configure our packet capture to encompass all packets on the network, or all packet traffic to and from a specific CM. This proved a valuable feature in investigation of interactions between CMs.

7. At the system performance testing stage, we found it necessary to isolate latencies associated with the propagation of video packets through the system. We found that use of a dual-homed analysis workstation configuration (supporting connections to two different networks by means of two independent network interfaces), with packet capture software capable of concurrent capture from multiple network interfaces using a common time base, simplified our latency analysis efforts significantly.

8. We encouraged each module vendor to assemble a copy of our mixed-simulation system test bed and provided them with copies of our stimulus/response testing package; this enabled the module vendors to perform development testing and predelivery testing equivalent to our own. The vendors embraced the mixed-simulation test bed and tools. The APL team and the vendor teams all believe that their pretesting accelerated the system integration effort at APL and reduced the need for vendor personnel to be present through integration.
9. All of the vendors were given direct access to our bug tracking tool, GForge. The GForge tool allowed us to set priorities for problem resolution and provided direct feedback to the vendors on the status of the integration of their equipment. This tight loop for problem identification and resolution reduced the system integration time.

ACKNOWLEDGMENTS: The AEODRS program has benefited from the participation of several organizations within the framework of the AEODRS SDIT. The SDIT members come from the following participating organizations: NAVEODTECHDIV (Indian Head, MD); APL (Laurel, MD); Penn State University Applied Research Laboratory (State College, PA); Battelle Memorial Institute (Columbus, OH); and SPAWAR Systems Center Pacific (San Diego, CA). The integration and testing approach as well as implementation and assessment activities described herein could not have been possible without the collective innovation, active participation, and full cooperation of all SDIT members.

REFERENCES

7“AEODRS Common Information,” EXT-AEODRS-10-2180-Common_Information Version 1.2, Naval EOD Technology Division, Indian Head, MD (June 2012).

The Authors

Mark A. Hinton is a member of APL’s Senior Technical Staff and serves APL’s AEODRS team as Systems Architect and as an integration lead. James M. Burck is a senior system and software engineer; he serves as the Lead Engineer for the APL system test bed and assists with systems engineering. Kristine R. Collins is a member of the APL Associate Technical Staff and serves as a contributor to the system test bed development and during system integration. Matthew S. Johannes is a member of the APL Senior Technical Staff; he serves as Lead Mechanical Engineer for the APL AEODRS team. Edward W. Tunstel Jr. is a senior robotics engineer in APL’s Research and Exploratory Development Department. Dr. Tunstel contributed to the AEODRS system test bed development and use during incremental system integration and testing and assisted with systems engineering activities. Michael J. Zeher is a member of the APL Senior Technical Staff and serves as Project Manager for APL’s AEODRS team. For further information on the work reported here, contact Mark Hinton. His e-mail address is mark.hinton@jhuapl.edu.

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.
In this article, we present preliminary work on motion planning and mapping algorithms for the Buckybot mobile robotic platform. We investigated implementation of wall-following algorithms and mapping unknown indoor environments by relying on rudimentary dead-reckoning and ultrasonic range finders. Buckybot is a ground-based platform whose geometry is based on a truncated icosahedron (a soccer ball shape with flattened sides). This platform has 20 passive hexagonal faces on which it can stably rest and 12 rounded pentagonal faces that can be extended linearly, allowing Buckybot to move. Because the robot is operational in any configuration, it is ideal for a variety of deployment scenarios, including throwing or dropping. Simulations grounded in experimental results show preliminary feasibility of Buckybot for indoor mapping applications.

INTRODUCTION

Buckybot is a new mobile robotic platform based on a truncated icosahedron (i.e., a soccer ball shape with flattened sides). It can rest stably on any of its 20 hexagonal faces and has 12 linearly actuated pentagonal faces that can be used to tip from hexagonal face to hexagonal face (Fig. 1). Each hexagonal face is adjacent to three actuators, allowing reliable movement in three directions. Pentagonal faces are rounded to prevent Buckybot from resting on a single actuator. In its current configuration, Buckybot moves by extending a single pentagonal face until the center of mass shifts sufficiently to incite a passive tip onto an adjacent hexagonal face.

The isotropic geometry of Buckybot makes platform locomotion independent of orientation. This can be advantageous when traversing unknown environments and in deployment scenarios in which there is a tumbling motion. As a result, one can deploy Buckybot in a variety of unconventional ways; Buckybot can be conceivably thrown, kicked, rolled, dropped, launched, etc., without compromising postdeployment locomotion. Additionally, the nearly spherical shape of Buckybot provides the possibility of both passive and active rolling, which can be ideal for fast locomotion scenarios and descending steep slopes.
Other groups have investigated and developed spherical rolling platforms. Unlike these platforms, Buckybot is unique in that it currently uses a configuration with flat faces and relies on linear extensions of its sides to locomote. Although polyhedrons cannot roll as quickly and require more energy for rolling because of impact with the ground, they have several benefits. For example, polyhedrons can rest on modest slopes without any actuation and can move in low-gravity or low-friction environments in which traditional wheel-based robots cannot operate.

Given the current tipping locomotion strategy, Buckybot is constrained to walk on what can be described as a honeycomb grid. Although this does not impose profound operational constraints, it prohibits the use of many traditional trajectory and wall-following algorithms. For example, for systems with continuous dynamics and continuous sensing, transfer functions can describe the relationship between user input and wall distance. With a continuous transfer function defined, feedback controllers can be designed to stabilize the systems. With Buckybot, we have a quantized input space, which is a function of both the current position and orientation. Some groups have performed research on motion planning subject to kinodynamic constraints, with lattices using linear integer programming techniques, and with the A* algorithm. However, these algorithms work better for motion planning and obstacle avoidance than for wall following. As a result, we propose new algorithms for wall following, with the goal of incorporating them into these more established algorithms in future work.

Given the geometry of Buckybot, to allow for equal sensing capabilities in all valid orientations, we propose the addition of sensors on all 20 hexagonal faces. For the proposed wall-following and mapping application, we will investigate the use of inexpensive range finders placed at the center of each passive hexagonal face and pointing radially outward. For the purposes of this preliminary work, we created an experimentally grounded simulation of the Buckybot platform with integrated ultrasonic range finders. To do so, we tested Buckybot and our proposed range finders independently. The purpose of these tests was to realistically define the pose uncertainty of Buckybot and the sensor noise associated with the range finder. Using this simulation, we evaluate the possibility of using a range finder-integrated Buckybot as a platform for autonomous navigation and mapping of unknown indoor environments. As part of this work, we also develop algorithms for identification of walls and wall following for Buckybot.

**BUCKYBOT PLATFORM**

**Actuation and Control**

Our current Buckybot prototype is approximately 26.0 cm (10.23 in.) in diameter with the pentagonal faces fully retracted. The distance between opposite hexagonal faces for this scale platform is approximately 23.9 cm (9.42 in.). Pentagonal faces are actuated using Haydon size-11 noncaptive stepper motor linear actuators. These actuators enable a reliable extension of up to 6.7 cm (2.64 in.) to enable tipping from face to face. Because of the slow speed of these actuators, tipping is currently the only feasible gait for the current Buckybot prototype. New actuators are currently under development for a smaller, faster, more agile Buckybot. A review of this effort will be discussed in Conclusions and Future Work.

The Buckybot is controlled wirelessly via Bluetooth. Commands are received and echoed using Roving Networks FireFly (RN-240/422) serial adapters connected to a communication board containing a mixed-signal microcontroller (C8051F410, Silicon Laboratories Inc.) that interprets and relays commands to a motor communication bus. Each motor is controlled independently using a motor board also containing a mixed-signal microcontroller (C8051F410, Silicon Laboratories Inc.). Motor boards are given unique addresses and are connected to the communication bus in series. Buckybot also contains an inertial measurement unit leveraging a Honeywell HMC6343 tilt-compensated magnetometer and its own unique Bluetooth module (RN-41, Roving Networks). This independent wireless connection enables streaming inertial measurement unit data to a remote interface without affecting communication with Buckybot.

All electrical components are powered using six 3.7-V, 1050-mAh polymer lithium-ion cells (Powerizer PL-553562-10C) wired in two parallel sets. These batteries offer an operating life of greater than 4.7 h (assum-
Geometry and Sensing

As mentioned in the Introduction, Buckybot is geometrically based on a truncated icosahedron. However, because the robot can only rest stably on its 20 hexagonal faces, Buckybot can be considered a regular icosahedron for the sake of motion planning. While on a given face, Buckybot can tip in one of three directions. With each step, Buckybot’s possible tipping directions shift by $60^\circ$. As a result, Buckybot is constrained to walk on a honeycomb lattice. To identify orientation, each hexagonal face is numbered from 1 to 20, and each actuator is labeled A through L.

Using the accelerometer available on the inertial measurement unit, Buckybot’s current resting face can easily be determined. To sense the world outside the robot, we simulated range finders placed in the center of each hexagonal face and pointing radially outward. While sitting on a given face, we assume that the four bottom- and top-most range finder integrated faces will not provide useful information about surrounding obstacles. Of the 12 remaining range finders, the upper six are oriented $19.47^\circ$ above the horizontal, and the bottom six are oriented $19.47^\circ$ below the horizontal. Assuming that walls and obstacles are vertical and floors remain level relative to Buckybot, the bottom faces have a visibility of only about $23.0$ cm ($9$ in.) before the sensor picks up the floor. As a result, we consider only the six sensors oriented $19.47^\circ$ above the horizontal for our algorithms.

As shown in Fig. 2, these remaining sensors are not spread uniformly around the robot in the horizontal plane. Adjacent range finders are separated by $44.5^\circ$, and range finders separated by an actuator are separated by $75.5^\circ$. As Buckybot moves, the orientation of the sensing lines changes relative to the global frame. This must be considered and accounted for in control algorithms.

WALL-FOLLOWING AND MAPPING ALGORITHMS

In this section, we discuss wall-following algorithms for Buckybot by using feedback from the six sensors mentioned in the preceding section. We assume that all walls are vertical and sufficiently tall that Buckybot can sense them. At the maximum sensing distance, $183$ cm ($72$ in.), walls need only be $80$ cm (31.5 in.) tall to be sensed.

Identifying and Locating Walls

An important and basic function of many ground-based mobile robots is the ability to locate and follow a wall at a prescribed distance by using feedback. For Buckybot, the first step to this is locating and properly identifying a wall by using the proposed range finder network. To complete this first step, we developed criteria for wall identification by using the following assumptions. First, we assumed that the walls are long enough to ensure that two adjacent sensors can pick up the same wall. If only one sensor registers an obstacle, then we assumed that the sensor is picking up an object and not a wall. Second, we defined a cutoff range for wall detection for all sensors. If an object is detected farther away than the cutoff range, in this case $183$ cm ($72$ in.), then Buckybot will ignore it.

By using basic geometry, both the distance of the robot to the wall and the orientation of the wall can be determined. To make these determinations, we use a combination of law of cosines, Heron’s formula, and law of sines.

Using Fig. 3, we first solve for $z$ using the range finder readings $(x, y)$ and the law of cosines. Once $z$ is defined, we apply Heron’s formula to solve for the area $(A)$ of the associated triangle,

$$A = \sqrt{s(s-x)(s-y)(s-z)}, \quad (1)$$

where $s = \frac{x+y+z}{2}$ is the semi-perimeter. Noting that the area of the associated triangle can be equivalently defined,

$$A = \frac{1}{2}d_1z, \quad (2)$$

where $d$ is the orthogonal distance from the wall to the center of Buckybot (Fig. 3). Combining Eqs. 1 and 2, and solving for $d$, we find that the associated distance is defined:

$$d = \frac{2}{z} \sqrt{s(s-x)(s-y)(s-z)}. \quad (3)$$
Before successful wall following, the wall’s inclination, $\alpha$ (Fig. 3), relative to the closest range finder must be determined. Using law of sines to find $\phi$ (Fig. 3), and noting that the sum of angles in a triangle is 180°, we find:

$$\alpha = \arcsin \left( \frac{x}{2} \sin(\phi) \right) - \frac{180 - \phi}{2}. \quad (4)$$

**Wall Following**

With the distance to the wall $d$ and inclination of the wall $\alpha$ determined, algorithms can be created to select a wall and follow it. It should be noted that with walls identified, wall-following algorithms can be chosen freely to suit the needs of the task at hand. For this preliminary work, we will follow the closest wall in a counterclockwise direction at a fixed distance of $d^*$. To accomplish this, we define a cost function (Eq. 5) weighing the importance of accurate wall following with the wall-following speed:

$$J(u(k)) = \sum_{i=1}^{k} \| (d(i) - d^*) \| + \| x(i) - (x(i-1) + x_d) \|. \quad (5)$$

Here, $x_d$ is a factor used to weigh the importance of moving along the wall as opposed to maintaining a distance $d^*$ from the wall. The control input $u$ defines the step sequence, and $k$ defines the total number of steps in a given sequence.

The cost function is minimized on a per-step basis. Before each step, we generate a target for the robot’s movement. To generate this target, we first find the point $d^*$ away from the wall along the shortest path connecting Buckybot’s center to the wall. We then add a vector parallel to the wall of a prescribed length $x_d$, as illustrated in Fig. 4. By increasing this vector’s magnitude, we promote fast traversal of the wall rather than maintaining an ideal distance of $d^*$ from the wall.

To impose the counterclockwise traversal constraint, we define the positive direction parallel to the wall as the cross product of the $+Z$ (out of page) direction with the shortest vector connecting the center of Buckybot to the wall. With the target determined, the robot considers its three viable movements and selects the one that minimizes the distance to the target. An alternative method could use the A* algorithm, applying heuristics to penalize walking too close or too far from the wall.

**Anticipated Implementation Issues**

As with many control algorithms, there are certain scenarios that may produce undesirable results. For this approach, we see two potential problems. First, corners, most notably 90° corners, are difficult to navigate. Second, narrow hallways and confined spaces can cause unnecessary confusion. To compensate, we propose the addition of a simple set of rules.

The first issue arises because Buckybot considers only the closest wall when planning its next move. Depending on which wall is determined to be closer, the target for the next step changes. In the right circumstances, this can cause the robot to get stuck in the corner, alternating between faces until perturbations in the system cause the eventual navigation of the corner. Of the solutions that compensate for this issue, the simplest involves an increase in the value of $x_d$, which results in improved navigation of corners. Additionally, a rule prohibiting backtracking movements can be applied; however, this has its own set of drawbacks, as one can easily imagine a scenario in which Buckybot might need to backtrack to successfully continue wall following. In future work,
we will consider a more robust solution using the four remaining range finders to detect upcoming corners and allowing us to plan accordingly.

Narrow hallways and confined spaces can also cause confusion. This confusion arises from the imposed constraint that forces Buckybot to follow the closest wall in a counterclockwise direction. In a scenario in which Buckybot is placed nearly equidistant from a sufficiently parallel pair of walls, a situation can arise in which Buckybot’s wall-following algorithm will alternate between walls. In this situation, Buckybot will get stuck alternating between faces until perturbations in the system cause it to select one wall over the other. This can be overcome by adding an exception to our imposed counterclockwise constraint on the wall-following direction. Specifically, if a wall is sensed on the opposite side of the robot, the robot will begin following in a clockwise direction. Note that this simple fix can cause further issues if sharp corners are present in the environment. As with the previous issue, we will address a more robust solution or set of solutions in future work.

EXPERIMENTAL ERROR DETERMINATION

Currently, Buckybot has yet to be equipped with range finders. To compensate, we independently evaluated both Buckybot’s tipping performance and the range finders’ performance to create a realistic simulation environment to validate our algorithms.

Test Setup

For range finder testing, we identified the LV-MaxSonar-EZ ultrasonic range finder (MaxBotix Inc.) as an accurate, easy to use, and cost-effective sensor. To determine the accuracy and noise of the LV-MaxSonar-EZ, we created a test rig to mount sensors 19.47° above horizontal (Fig. 5). For testing, we considered three individual sensors, each tested at distances ranging from 15.24 cm (6.0 in.) to 243.84 cm (96.0 in.) on an interval of 7.62 cm (3.0 in). At each distance, 25 measurements (consisting of five time-averaged readings each) were taken from each individual sensor. An Arduino Pro Mini using a 10-bit analog-to-digital converter was used to take readings from the range finders. Data were collected to the nearest 2.54 cm (1 in.) due to the automatic filtering of the LV-MaxSonar-EZ.

To test Buckybot, we evaluated every possible tipping combination a total of 10 times. We used a Vicon tracking system (Vicon MX Giganet controller with four Vicon MX-T10 1.0-megapixel cameras). For this system, the position accuracy determined during calibration is consistently better than ±0.7 mm. For each tipping combination, two rigid bodies containing reflective markers were attached to Buckybot. One rigid body was attached opposite the resting face of Buckybot. The second rigid body was attached opposite the face Buckybot will tip to. An image of Buckybot with the two rigid bodies attached is shown in Fig. 6. Once the rigid bodies were attached, Buckybot was tipped using its relevant actuator a total of 10 times for each of the

Figure 5. Image of the test rig (left) designed to mimic the 19.47° angle of the six faces used for distance measurements. Faces 2, 12, and 18 on Buckybot (right) represent three of these six faces.

Figure 6. Image of Buckybot with Vicon tracking markers attached to faces opposite the current resting face and the face that Buckybot will tip to.
60 tip combinations, tracking its movement (position and orientation) through the entire sequence. In this work, we considered only the first and last frame of this sequence corresponding to the two resting states of Buckybot (pre- and post-tip) to determine the repeatability of tipping.

Test Results

We found that the range finders had low noise for ranges rate, from 30.48 cm (12.0 in.) to 182.88 cm (72.0 in.). At ranges closer than 30.48 cm (12.0 in.), the sensors did not act reliably and fluctuated greatly. At ranges farther than 182.88 cm (72.0 in.), two of the sensors notably lost accuracy. These issues will be addressed in Simulation and Results. Using this information, we used least-squares methods to develop a best-fit noise model. This yields the following:

\[ d_{\text{measured}} = d_{\text{actual}} + (0.013d_{\text{actual}} + 0.448)w, \] (6)

where \( d_{\text{measured}} \) is a simulated noisy measured distance (cm), \( d_{\text{actual}} \) is the true distance (cm), and \( w \) is normally distributed noise with mean zero and unit variance.

With the Buckybot test results, we found some differences in the accuracy of tipping onto certain faces. However, differences were not significant enough to justify accounting for them individually in this initial effort. As such, we define the mean and standard deviation in position and orientation over all possible moves combined. We further simplify the system by assuming that there is no correlation between position and orientation measurements. In doing so, we find the mean and standard deviations for our positions as \( \mu_x = 0.745 \text{ cm}, \sigma_x = 0.149 \text{ cm}, \sigma_y = 9.266 \text{ cm}, \text{ and } \sigma_z = 0.105 \text{ cm}, \) where the +y direction is defined as the tipping direction, and the +x direction is defined such that the +z direction points out of the floor. The mean and standard deviation associated with our heading measurements are \( \mu_0 = 0.31^\circ \text{ and } \sigma_0 = 0.44^\circ, \) respectively. Note that \( \theta \) is defined about the +z axis relative to the initial +y direction. With these values, equations matching the form of Eq. 6 can easily be assembled.

SIMULATION AND RESULTS

A simulation environment was created in MATLAB in which a virtual Buckybot (Fig. 7) can detect obstacles and walls. By using this simulation, we can evaluate the algorithms detailed in Wall Following. At each step, noise is added into the range finder measurements according to Eq. 6 and then reported to the nearest 2.54 cm (1 in.) to account for the automatic filtering of the LV-MaxSonar-EZ. As mentioned before, the sensors do not work reliably at distances less than 30.48 cm (12.0 in.) or greater than 182.88 cm (72.0 in.). At distances too close, the sensors do not settle to a final value. In our setup, we assume a static environment and that Buckybot is static between tips. Thus, if sensors do not settle in a fixed time interval, it can be reasoned that Buckybot is very close to an obstacle. At distances too far, the sensors jump to large values unexpectedly. Thus, if a distance greater than 182.88 cm (72.0 in.) is detected, we will ignore the sensor reading entirely.

The position of Buckybot is also altered at each step according to the statistics reported in Test Results. These noise parameters are added into the Buckybot position estimates with each step using equations similar to Eq. 6. When running the simulation, range finder measurements are recorded in a matrix in which each column represents a sensor number and each row represents a step taken by the robot. The sequence of actuators fired is also recorded. After the simulation runs, we repeat the ideal trajectory of the robot using the sequence of actuators fired, only this time omitting noise. We can then replot the points where Buckybot sensed an obstacle. Figure 8 shows a sample of a single trial. After running multiple simulations of different environments, we found that the additive position error was more substantial than originally thought. By running Monte Carlo simulations, we found, for example, that the standard deviation in uncertainty in position after 100 steps (roughly traveling 929 cm) was 90.42 cm (35.6 in.) and the standard deviation in orientation was 11.5°. However, several methods to reduce this noise are discussed in the following section.

CONCLUSIONS AND FUTURE WORK

In this article, we presented the preliminary implementation of wall-following algorithms and the mapping of unknown indoor environments relying on rudimentary dead-reckoning and ultrasonic range finders. The system of interest was the Buckybot mobile platform whose
significantly. For example, to reduce impact forces, we can simultaneously extend counteracting actuators. This will allow Buckybot to gradually come to rest on a tipping face, reducing the overall tipping speed and thereby reducing or eliminating the impact after a tip. Additionally, the rigid faces of Buckybot could be replaced with a compliant material to absorb some of the energy of impact.

Our implementation of a wall-following algorithm on the Buckybot platform was repeatedly demonstrated in simulation with positive results, even with realistic noise parameters applied. Mapping was also investigated, with results that were greatly affected by the position uncertainty associated with Buckybot’s locomotion. In future work, we plan to incorporate our wall-following algorithms into a unified control scheme that is more robust and requires less special considerations. This control scheme will likely build on applications of linear programming or the A* algorithm. We also will investigate use of the Kalman filter and simultaneous localization and mapping algorithms to achieve better mapping capabilities. Lastly, we will consider other sensing systems such as networks of cameras to increase sensing capabilities.

As mentioned previously, work has been conducted in collaboration with The Johns Hopkins University Department of Mechanical Engineering senior design program to develop actuators for a softball-sized (8.9 cm) Buckybot. Although initial efforts were promising, we are still working to improve overall actuator performance. The current actuator design involves a three-stage mechanism consisting of a charging phase during which a small DC motor slowly charges the actuator spring, a fire phase during which the energy from the actuator spring is released in a high-impulse linear extension, and a fast retract phase during which the extension is quickly retracted and locked back into place for charging. The specific goal of this work is to develop actuators capable of producing extension impulses high enough to enable both jumping and fast rolling of the isotropic geometry allows for deployment in unconventional ways and is ideal for low-gravity or low-friction environments. The system currently locomotes by tipping using linear actuators; however, work is currently underway to replace these actuators with high-impulse spring actuators. This would allow for smoother, faster tipping as well as the possibility of other modes of locomotion such as continuous rolling or jumping.

For the purposes of rapid capabilities assessment, the Arduino Pro Mini was used for simple data acquisition from range finding hardware. This choice of hardware was solely for the purposes of this testing, and this data acquisition capability can easily be incorporated into an extension of Buckybot’s onboard control system without requiring the explicit addition of an Arduino Pro Mini.

Although intuition suggested that Buckybot’s locomotion would result in a relatively low position uncertainty due to its discrete dynamics, we found through experimentation that the uncertainty was more significant than previously thought. We believe that this is an engineering problem and that solutions can be proposed to reduce the uncertainty in position and orientation.

Figure 8. Map created by Buckybot simulation. Circles represent theoretical positions, red x’s represent actual positions, and black x’s represent locations where Buckybot simulation detected an obstacle assuming no uncertainty in position.
Buckybot platform. Actuators are designed to be fired in sequence to enable predictable parabolic jumping for increased terrain handling, including stair climbing. To enable rolling, the fast retract phase was added to the actuator functionality to retract actuators before they hinder rolling.

In addition to traditional range finders for mapping, a variety of tools and sensors are being considered for the Buckybot platform. Specifically, the geometry lends itself to a variety of sensing modalities, including acoustic sensing, chem/bio, and visual sensing. In circumstances in which identical sensors can be distributed around the body of Buckybot, simple gradient-based path generation techniques can be used to guide the robot toward or away from a desired source (e.g., a chemical plume or an acoustic source). In a case in which multiple cameras are distributed around the surface of Buckybot, camera calibration and image meshing techniques can be used to give an operator a stable view from any direction as the robot moves. Additionally, these meshed video feeds can also enable an operator to effectively look around without requiring movement from the robot.

ACKNOWLEDGMENTS: The authors thank Mr. Wolfger Schneider and Mr. Rafal Szczepanowski for their extensive contributions to the development of the Buckybot platform and Dr. Gregory Chirikjian for providing range finders for this work. The authors also thank Mr. Aaron Goldblum and Ms. Anuparna Challa for their assistance with the sizable data collection associated with this work. This work was supported by the Janney Publication Program and Independent Research and Development funds provided by APL.

REFERENCES

The Authors

This investigation was conducted out of curiosity during the team’s off-hours during the summer of 2011 while Robert C. Grande was interning as part of the operations team of STEREO in the Civilian Space Department. The test bed was completed by Michael D. M. Kutzer and Christopher Y. Brown and in 2007 using Independent Research and Development funds. The original Buckybot concept was initially developed by Mehran Armand and colleagues in 2005. Since this effort, Robert Grande has continued his education and is currently working toward a M.S. in aerospace engineering at MIT. Michael Kutzer is investigating methods associated with the modeling and control of highly dexterous robotic manipulation systems. Christopher Brown’s work currently focuses on efficient, effective, and predictable small-scale locomotion. Mehran Armand also continues to work in robotics research, with a focus on computer-assisted surgical systems. For further information on the work reported here, contact Michael Kutzer. His e-mail address is michael.kutzer@jhuapl.edu.

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.
We present a biologically inspired morphable mechanism for a small multi-legged robot and characterize the mechanism for statically stable climbing up convex/cylindrical structures. The design includes an analysis of the robot's static stability while climbing such structures. Preliminary experimental results highlight the merits and limitations of the overall approach.

INTRODUCTION

Robots with climbing mobility in structured and unstructured environments remain a focus of research aiming to achieve robust or general capabilities for a range of missions. In this article, we focus on the narrowly defined problem of using a robot to climb convex structures such as utility poles and trees. Such climbing machines would afford a wide range of possible practical applications, such as power grid maintenance and bridge inspection. Moreover, the problem is defined narrowly enough to enable a principled design approach.

Finding ways of climbing convex structures presents a host of challenges for the research community. This article focuses on a biologically inspired, morphable mechanism and associated robot for climbing known convex/cylindrical structures in man-made environments (e.g., pipes, electrical conduits, small pillars, chair/table legs, large ropes, large wires, etc.). We envision that the approach defined in this article would also be applicable to locomotion on natural surfaces, such as trees, with modest additional development. Specifically, we present and characterize a climbing approach inspired by simple observations of beetle climbing behavior.

This article focuses on small robots that climb using a set of morphable limbs. In this context, morphable limbs refer to legs with many degrees of freedom. Constraining the operating criteria to be specific to convex/cylindrical structures permits simplification of the overall problem. Specifically, studying this subset of structures makes it feasible to rely solely on simple friction interactions rather than requiring advanced materials, microspines, or both [e.g., as used on existing systems such as RiSE (Robotics in Scansorial Environments), StickyBot, or SpinyBot systems].

We present simple principles and equations for statically stable climbing by a multi-legged robot with morphable limbs. Preliminary experimental results suggest the strengths and limitations of the overall approach.

BACKGROUND

Researchers have proposed a number of robot designs and prototypes that are capable of locomotion on convex/
Morphable Limb for Climbing Locomotion

We observed qualitatively that beetles, when climbing relatively small grasses and stalks, use a “hugging” grip to maintain sufficient climbing forces (Fig. 1). Additionally, they appear to use a “hug-shimmy-hug” gait to traverse upward. Using this beetle climbing behavior as inspiration, we investigated leg designs that would enable the robot to conform to the outer surface of the structure of interest. Our proposed solution involves a simple cable-driven “pull-pull-pull” system capable of wrapping around structures using varied cable actuations. Individual rigid vertebrae are spaced along a thin superelastic nitinol rod. This rod maintains spacing between vertebrae while allowing the overall structure to flex in response to tensioning cables and interactions with the environment. An alternative mechanism, designed for medical applications, that also relies on drive cables and elastic deformation of superelastic nitinol is described in Ref. 15. Representative climbing concepts are depicted in Figs. 2a and 2b. When considering this type of articulated leg, additional climbing methods, such as the capability to traverse within hollow structures, can also be imagined.

Characterization of Symmetric External Climbing

As a preliminary effort, we consider the robot climbing a straight cylinder such as a pipe. Given this simple structure, and assuming a multi-legged robot with legs offset sufficiently to enable overlapped wrapping, we see that formulation of the climbing problem can be simplified to consideration of a set of basic parameters. For sim-
plicity, we reference robot-specific terms relative to a potential tipping point on the robot. To define this tipping point, we assume that the robot selects an orientation on the underside of a pipe (i.e., it hangs below the pipe for all non-vertical pipe orientations). Using this point, illustrated in Fig. 3, we define the lengths $L_1$ and $L_2$ as the distances between the mean position of pairs of legs and the tipping point (a third length, $L_3$, can be defined in a similar fashion, measuring from the tipping point to the forward leg). Additionally, the system center of mass is defined using $\lambda$ as the axial offset from the tipping point and $h$ as the offset from the pipe wall perpendicular to the pipe axis. The pipe itself is characterized by a rise angle, $0^\circ < \alpha \leq 90^\circ$ defined above the horizontal, and a diameter $D$. With these parameters, and assuming that the robot’s movements are sufficiently slow to treat the system as effectively static, we see that the minimum friction force to keep the robot from sliding down the pipe is defined as

$$F_{\text{slide}} = mg \sin(\alpha),$$

where $m$ is the robot net mass, and $g$ the acceleration due to gravity. Likewise, the minimum torque required to keep the robot from tipping is defined as

$$\tau_{\text{tip}} = mg(h \sin \alpha + \lambda \cos \alpha).$$

In the plane orthogonal to the cylinder axis, we add an angular offset parameter $-90^\circ \leq \theta \leq 90^\circ$ to define the robot’s effective orientation on the pipe. Using this term, we introduce a minimum required torque to keep the robot from twisting or rolling on the pipe as

$$\tau_{\text{roll}} = \left(\frac{D}{2} + h\right)(mg \cos \alpha \sin \theta).$$

Assuming that the robot is symmetric in the plane orthogonal to the cylinder axis, we introduce a single term, $w$, to describe the robot’s body width, which, given the pipe diameter, corresponds to a term that we will refer to as the angular width of the robot (Fig. 4),

$$\beta = \tan^{-1} \frac{2w}{D}.$$  \hspace{1cm} (4)

Assuming a leg consisting of $m$ discrete, evenly spaced elements, we introduce parameter $\gamma$ to describe their angular displacement when wrapped around a pipe (Fig. 4b) and defined as

$$\gamma = \tan^{-1} \frac{2v}{D},$$

where $v$ represents the linear spacing between leg elements on an uncurved leg.

Using these parameters, we can calculate the net force for each pair of legs as

$$F_{\text{legs}} = -2 \sum_{i=1}^{m} N_i \cos(\beta + i\gamma),$$

where $F_{\text{legs}}$ is defined in the opposite direction of $N_{\text{body}}$ shown in Fig. 4b. In Eq. 6, $N_i$ represents the normal force of the $i$th segment of the leg. The term $N_{\text{body}}$ repre-
sents the net normal force seen by the body of the robot. Applying these fundamental equations, we see that for our system to maintain static stability, the following net force would apply for a set of paired legs:

\[ F_{\text{legs}} \leq 2 \sum_{i=1}^{12} N_i \cos(\beta + i\gamma) + 2\mu \sum_{i=1}^{12} N_i \sin(\beta + i\gamma). \]  

(7)

Here, \( \mu \) represents the coefficient of static friction between the leg and climbing surface. Although this relationship is potentially useful, not all terms used can be readily defined or measured using onboard sensors and environmental understanding. Specifically, defining the relationship between segment normal forces (N) and applied cable tensions requires experimental testing.

**Preliminary Test Results**

A preliminary prototype integrating a superelastic nitinol backbone with a series of 12 evenly spaced commercially available shaft couplings was used to evaluate the feasibility of climbing using this basic approach. To evaluate force distributions of the elements of the leg, FlexiForce pressure sensors were mounted along the outside of two pipes of different diameters, 2.375 in. (~6 cm) and 4.3125 in. (~11 cm). As shown in Fig. 5, the sensor spacing ensured that vertebrae of the leg would interact roughly with the center of each force sensor.

Preliminary pressure-testing results show a non-uniform distribution of forces around climbing surfaces (Fig. 6) and an inverse relation between cable tension and robot angular width on the pipe (Fig. 7). As shown in Fig. 5, a 12-segment leg is wrapped around a 2.375-in. pipe from the side starting in a vertical position. FlexiForce sensors are placed between the pipe and segments to measure contact force. Cable tension is then applied, starting with 1 lb of tension and increasing in intervals of approximately 1 lb until 11 lb of tension is reached. As the cable tension is increased, the change in contact force distribution is measured (Fig. 6). Green bars represent both the assumed (normal) direction and relative magnitude of the contact force.

To quantify the potential for climbing, Fig. 7 shows the body angle (\( \beta \), Eq. 4) required to achieve a non-zero lifting force (i.e., a force pressing the body of the climber to the surface of the pipe) from the legs.

The results in Fig. 6 show that increased cable tension results in a more uniform distribution of contact forces along the pipe. Additionally, these data suggest that, given the current method of actuation, leg length should be reduced given the negligible contact forces of the latter segments of the leg (i = \{8,9,10,11,12\}).

Possible causes for the variability in segment contact forces for lower cable tensions include variance in tendon moment arms (due to manufacturing tolerances),

![Figure 5. Force distribution testing with pressure sensors and morphable leg wrapping around the 2.375-in. pipe.](image)

![Figure 6. Contact force distribution as a function of drive cable tension.](image)
uneven curvature and buckling in the superelastic backbone, and unpredictable frictional effects between the tendon and segments. Some recommendations for future improvements include reducing cable friction through material selection, reducing the weight of individual segments to prevent uneven loading and buckling in the superelastic backbone, and improved manufacturing techniques for greater reliability.

Results from Fig. 7 suggest that a larger angular width ($\beta$, Eq. 4) is advantageous to climbing; however, this assumes that the body conforms perfectly to the pipe in question. Additionally, given the limitations of the current prototype leg, there is an apparent advantage to climbing small pipes.

These results also suggest that robot body mass must remain low for realistic climbing scenarios. Results will likely improve as a result of decreased cable friction effects to maximize the ratio between cable tension and contact force, designed asymmetry in backbone stiffness to encourage more evenly distributed contact forces along climbing surfaces, and optimization of leg segment geometry to improve contact force distribution.

These pressure test results show a clear relation between body angle ($\beta$, Eq. 4) and the required cable tension to achieve a positive holding force. These results collectively inform minimum next-step design decisions and functional requirements of a complete multi-legged robotic system based on morphable limbs.

CONCLUSIONS
A goal of this research is to realize robots with morphable limbs that are capable of agile locomotion in the plane, transition maneuvers from planar locomotion to climbing, and climbing on a variety of convex structures. Actively flexible leg designs that can morph to and generate forces necessary to adhere to various surfaces represent an advance beyond existing robotic models. Limitations of the actuation strategy, however, suggest that dominating friction forces and scaling may limit the current approach. Several technologies apply for candidate materials and actuation, such as electroactive polymers, shape memory alloy wires, and/or conventional tendon-driven structures with conventional DC motors. For real-time gait adaptation and feedback control, future plans consider integration of small-scale tactile sensors in the legs of the robot.

ACKNOWLEDGMENTS: This work was supported by The Johns Hopkins University and Independent Research and Development funds provided by APL.

REFERENCES


This work was initiated by Mehran Armand in collaboration with Michael D. M. Kutzer and Christopher Y. Brown in 2008 under an Independent Research and Development effort investigating bio-inspired robotic systems. Noah J. Cowan, of The Johns Hopkins University’s Department of Mechanical Engineering, was recruited because of his expertise in biological systems and his prior work with the APL team studying undulating fin locomotion. Cowan’s student Danoosh Vahdat studied and documented beetle climbing behavior. Jonathan E. Clark, of Florida State University’s Department of Mechanical Engineering, was invited to join the effort to develop conceptual climbing models and static experiments to demonstrate stable climbing. Leg development, prototyping, experimentation, and characterization were performed by Mike Kutzer and Chris Brown, with follow-on work conducted by Edward W. Tunstel Jr. and Mike Kutzer. Lessons learned from this work have directly translated into continuum surgical manipulator development, which is currently ongoing at APL. For further information on the work reported here, contact Mike Kutzer. His e-mail address is michael.kutzer@jhuapl.edu.

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.
PL and the Marshall Space Flight Center have been working together since 2005 to develop technologies and mission concepts for a new generation of small, versatile robotic landers to land on airless bodies, including the moon and asteroids, in our solar system. As part of this larger effort, APL and the Marshall Space Flight Center worked with the Von Braun Center for Science and Innovation to construct a prototype monopropellant-fueled robotic lander that has been given the name Mighty Eagle. This article provides an overview of the lander’s architecture; describes the guidance, navigation, and control system that was developed at APL; and summarizes the flight test program of this autonomous vehicle.

INTRODUCTION/PROJECT BACKGROUND

APL and the Marshall Space Flight Center (MSFC) have been working together since 2005 to develop technologies and mission concepts for a new generation of small, autonomous robotic landers to land on airless bodies, including the moon and asteroids, in our solar system. This risk-reduction effort is part of the Robotic Lunar Lander Development Project (RLLDP) that is directed by NASA’s Planetary Science Division, Headquarters Science Mission Directorate. During this ongoing collaboration, APL has led development of several subsystems, including guidance, navigation, and control (GNC); flight software; and mechanical design. MSFC has led development of several subsystems, including propulsion and power systems. A variety of technology risk-reduction efforts, illustrated in Fig. 1, have been performed to explore technologies to enable low-cost missions.

As part of this larger effort, MSFC and APL also worked with the Von Braun Center for Science and Innovation (VCSI) and several subcontractors to construct the Mighty Eagle, a prototype monopropellant-fueled robotic lander. The primary objective for the lander was to test and mature the GNC system for the final 30–50 m (98–164 ft) of the descent stage before landing. The prototype development effort also created an integrated team of engineers at MSFC and APL who could quickly transition to potential space missions. The lander has proven itself as a test bed to provide hands-on
A blowdown 90% pure hydrogen peroxide monopropellant propulsion system that is pressurized using regulated high-purity nitrogen provides actuation for both the attitude control system (ACS) and the descent control systems. Hydrogen peroxide was chosen for the prototype system because its decomposition byproducts, steam and oxygen, are both nontoxic, and it provides sufficient energy density to achieve the target flight times. The propulsion system, built by Dynetics in collaboration with MSFC, APL, and VCSI, feeds 16 monopropellant thrusters: twelve 44.5 N (10 lbf) attitude control thrusters, three 267 N (60 lbf) descent engines, and a throttleable engine with a maximum thrust of approximately 3114 N (700 lbf). The thruster configuration is illustrated in Fig. 3. The 12 attitude thrusters are grouped into six coupled pairs to allow torque to be applied independently to each of the three rotation axes of the vehicle. The three fixed descent engines provide the vertical thrust to control the vehicle’s altitude and descent rate. The large throttleable engine provides Earth gravity cancellation (EGC). The EGC engine nominally produces a thrust of five-sixths the weight of the lander throughout the flight to approximately simulate lunar gravity for the rest of the system by nulling the difference between Earth and lunar gravity.

**VEHICLE HARDWARE DESCRIPTION**

The prototype lander has a three-legged configuration, as shown in Fig. 2. The footpads of the three legs form a triangle with sides 2.13 m (7.0 ft) long. The distance between the two decks is 0.61 m (2.0 ft). The dry mass of the vehicle is approximately 206 kg (454 lbm), and it can fly with 116 kg (256 lbm) of propellant and 7 kg (15 lbm) of pressurant. This allows a maximum flight time of approximately 45–50 s with 5% reserves at touchdown.

**Propulsion System**

This article provides an overview of the prototype lander, with an emphasis on its GNC system, and outlines the testing performed on the lander both leading up to and including multiple flight test campaigns. The article is divided into three main sections. The first section provides an overview of the vehicle hardware, including actuators and sensors. The second section describes the software architecture and algorithms used by the GNC subsystem, including several optical navigation systems. The final section summarizes the various simulation, processor-in-the-loop, and flight tests that were performed on the prototype lander.
gravity. A fixed EGC engine was chosen over a gimbaled design to minimize system complexity, cost, and schedule constraints.

Sensors

The GNC sensors, shown in Fig. 4, were selected to provide flight-like feedback to ensure applicability to a lunar mission and include an inertial measurement unit (IMU), radar altimeter, optical camera, and ground contact sensors.

The LN200 IMU provides angular rates and three-axis accelerations at 400 Hz that are used to propagate the vehicle attitude, velocity, and position. The Northrop Grumman-produced LN200 was chosen because the non-space-qualified version used on the Mighty Eagle is an affordable solution that meets the performance requirements, while the space-rated version offers a potential solution for an actual lunar mission.

The Type 2 Miniature Radar Altimeter (MRA) provides vertical position from 0.2 to 100 m (0.66–328 ft), covering the full altitude range of the Mighty Eagle. The Roke-produced MRA provides a low-cost and low-power option for the prototype. For actual lunar missions, several sensors, including a Honeywell HG8500 series altimeter and the APL-developed miniature lidar altimeter, are under consideration. The Honeywell HG8500 has heritage with Mars landing missions, and the lidar altimeter is being developed as a low-power, low-mass, and low-cost alternative for a variety of planetary missions. Parallel testing of these sensors was conducted during a helicopter field test as part of the larger RLLDP.

A commercially available digital camera, the illunis RMV-4201 with an Active Silicon Phoenix D48CL Frame Grabber, allows testing of optical navigation algorithms on the lander. The images from this nadir-facing camera can be used to derive lateral velocity and terrain relative position. This camera provides representative images for developing the algorithms but would be replaced with a space-qualified sensor for the lunar missions.

Contact switches mounted on the main pivot point of the legs provide a positive indication that the vehicle is on the ground and can, along with other inputs, safely terminate the propulsion system.

A NovAtel ProPak global positioning system (GPS) receiver is included on the vehicle, although its data are not made available to the primary GNC system because no equivalent sensor is available on actual lunar flights. The GPS information is logged for postprocessing and also used by onboard autonomy software that can declare autonomous abort sequences if the vehicle exceeds predefined flight envelopes.

Processor

The lander avionics unit (LAU), developed by Southwestern Research Institute (SwRI), performs the onboard processing, including processing GNC sensor data, running GNC algorithms, performing data logging, and running optional optical navigation algorithms. The LAU includes a flight-like BAE Systems RAD750, a common processor on space missions, with 256 MB of memory running at 133 MHz. The avionics could support alternate single-board computers when faster models are developed and qualified. Although this processor has a low rate compared with modern non-space-qualified processors, it was chosen to represent
the team with experience working with these materials, which could be used to reduce structure mass on an actual space-flight vehicle. A second pair of aluminum decks were also designed and manufactured in-house at MSFC. Mechanical and thermal modeling and analysis were performed on both sets of decks, although the aluminum decks were installed on the prototype because they were available at an earlier date.

Mechanical engineers at APL designed and tested shock-absorbing legs for the prototype. The damping mechanisms include both a hydraulic telescoping damper for the nominal loading conditions and a single-use, crushable honeycomb for larger loads. Through experience gained through the leg testing process, the APL team was able to refine the leg design, removing approximately 13 kg (29 lbm) of mass per leg, corresponding to a reduction of leg mass by one third. The design was highly successful, accommodating the harsh testing environments and varying landing conditions.\textsuperscript{13,14}

Currently available flight processors for actual space missions. A large multi-megabyte memory card was also added to the LAU to perform data archiving of GNC flight data, various temperature and pressure measurements, and acquired images.

**Structures and Legs**

The majority of the vehicle structure, including the two circular deck plates, the legs, and the propellant tanks, is constructed of aluminum and composite materials. All lander materials were chosen to meet compatibility requirements with hydrogen peroxide.

The circular decks provide a large part of the vehicle structure and protection for the high-price components of the vehicle, including the LAU, the IMU, and the propellant tanks, which are located between the decks. Two independent sets of decks were designed and built for the lander. The first set was designed by using a composite/aluminum honeycomb sandwich to provide
To perform initial shakeout flights, APL engineers also designed a tether system, shown in Fig. 5, to constrain vehicle motion. The tether geometry allows adequate translation and attitude motion of the vehicle to perform short vertical flights approximately a foot off of the ground, while preventing the vehicle from tipping over in the event of a malfunction. Each of the three tethers connects a lander foot to the ground and consists of steel cable inline with a Miller SofStop Shock Absorber. These off-the-shelf shock absorbers, nominally used as part of equipment to protect workers from falls, provide damping to absorb any kinetic energy of the vehicle if the tethers are pulled taught. Additional steel cables, in parallel with the shock absorbers, limit the absorbers’ maximum travel. This innovative, low-cost system allows lower-risk testing of the vehicle after any design modifications or maintenance.

AUTONOMOUS GNC SYSTEM

The onboard flight software, including the GNC algorithms, was designed to match that of an actual lunar lander vehicle as closely as possible within the time and schedule constraints of the program. The flight software and GNC subsystems are responsible for processing sensor data, estimating the vehicle state, and commanding thrusters control the vehicle along the desired trajectory.

Software Architecture

The software architecture is built around the core flight executive (cFE) modular software environment developed by NASA Goddard Space Flight Center. The cFE provides abstraction layers and multiple key services, shown in Fig. 6, including board initialization, event logging, and a software bus, to promote reusability across processors and missions. Additional modular applications are added as needed to handle various vehicle tasks including GNC, optical navigation algorithms, sensor input preprocessing, actuator control, and data logging. All communications between applications are messages handled in a “publish and subscribe” manner. Applications subscribe to individual messages but are unaware of the publishers of these messages. Applications publishing messages are unaware of the recipients, if any, of these messages. Isolating module interaction to message passing allows applications to be swapped out with no effect as long as the published messages remain constant. This modular architecture allowed incorporation of various software components from lander team members at different locations. For example, engineers at APL designed the primary GNC module, GNCA, and an optical navigation module to estimate lateral velocity from images. Engineers working at VCSI and MSFC designed the backup GNC module, GNCB, a control module to drive the throttleable EGC engine, and the vehicle state man-

![Figure 5. Tether system for initial shakeout flights. (NASA)](image)

![Figure 6. Modular software architecture.](image)
ager that performs all sequencing operations, ensures a unique commanding source between the primary and backup GNC systems, and initiates autonomous soft-aborintially performed by commanding an open-loop constant abort sequences if the GNC navigation states or commands exceed predefined thresholds. These components velocity command in the desired direction of motion were seamlessly integrated into the final system by using were set independently in either position or velocity mode. This setup allowed four discrete translation control modes to be commanded, as illustrated in Table 1.

**Guidance, Navigation, and Control**

The algorithms and software for the GNC systems, with exception of the image processing algorithms, were designed using MathWorks’ Simulink and Embedded MATLAB software packages. The algorithms were first created and tested entirely within the Simulink simulation environment. The GNC engineers then used the Real-Time Workshop package by MathWorks to autogenerate C code to execute the GNC system. The autogenerated algorithm code was functionally wrapped with hand-generated code to handle all application and task creation, initialization, and interfaces with the cFE communications. The wrapper code takes all input data packets and converts them into the format and units expected by the GNC blocks, executes the GNC blocks at the appropriate times, and converts the output of the GNC blocks into telemetry packets that are published at the desired rates. The majority of the GNC code on the flight system, including state propagation, guidance, and thruster commanding, runs at 50 Hz. A second 1-Hz navigation filter incorporates lower-rate sensor data including altimeter measurements and optical measurements.

The 50-Hz input data consist of sensor data, desired state commands, and messages from the image processing system. The IMU data contain buffered 400-Hz data from the previous 20-ms window and are first passed through a preprocessing module that performs data checking, removes known biases, and outputs average angular rates and translational accelerations rotated into the vehicle body frame. These data are then sent to the attitude determination system (ADS) and the navigation filter, which estimate the vehicle state. The commanded lateral vehicle state is used by the lateral guidance module to determine the commanded vehicle orientation. This desired attitude command, along with vertical motion commands, is sent to the control module that determines the appropriate thrusters to fire. The architecture of the GNC software is illustrated in Fig. 7.

**ADS and Navigation Filter**

Before takeoff, the IMU is used to determine the initial vehicle attitude relative to the topocentric frame. Accelerometer data are used to estimate the ground normal force and, thus, the vehicle attitude relative to vertical. The magnitude of the sensed normal
force is used to estimate accelerometer bias along the normal force direction. During this initialization, the gyroscope measurements are also used to measure the direction of Earth’s rotation. These data are provided to the navigation system to compensate for the rotation of Earth relative to the inertial frame. Once the flight sequence begins, the ADS can propagate only the initial vehicle orientation by using measured angular rates from the IMU and the estimated Earth rotation rate. On a real mission, the attitude determination would also incorporate attitude measurements from an onboard star tracker.

The translation navigation filter is decoupled from the attitude filter and consists of two parts: a fast 50-Hz propagator and a 1-Hz filter. The fast component of the filter propagates the position and velocity by using an onboard gravity model and measured accelerations from the IMU that are rotated using the attitude estimate from the ADS. Other measurements from the altimeter and optical navigation system are buffered along with the IMU data and sent to the 1-Hz navigation filter that incorporates them using Kalman filtering techniques. The 1-Hz filter maintains a covariance estimate for the vehicle state and provides state corrections to the 50-Hz propagator. This approach, using filters with multiple rates, is based on heritage from previous APL spacecraft that used a similar approach for incorporating star tracker measurements in the ADS and allows for measurements that are either out of sequence or delayed within the 1-s window to be time-ordered and processed. This approach also allows the computationally expensive steps of the navigation filter to be run at lower priority and spread over a larger time frame to even out computational loading.

Because optical navigation measurements can take multiple seconds to process, they are incorporated into the navigation filter by using state augmentation and fixed-point smoothing. Although the actual image processing algorithms can take several seconds to run, the fact that images are taken is known immediately. At the time each image is taken, the navigation filter state is augmented to include the position of the vehicle when the image was taken in addition to the vehicle position at the current time. When an optical measurement arrives after computation processing is completed, it contains information about the difference in the vehicle location at two times in the past. By using augmented states, this measurement can be treated by using a standard Kalman filter where the measurement is not dependent on the current vehicle state but on the augmented states.

### Lateral Guidance

The lateral position and velocity of the vehicle are controlled by tilting the vehicle to apply a portion of the vertical descent engine thrust in the lateral direction. The inputs to the lateral guidance law are the commanded position and velocity in the topocentric frame and the estimated vehicle position, velocity, and acceleration. A weighted sum of the components of the position error, velocity error, and acceleration that are perpendicular to the local vertical are used to define a total lateral error. A correction to the vehicle attitude is calculated to tilt the vehicle relative to vertical to provide a lateral force proportional to this lateral error up to a saturation limit. This attitude correction is also fed through a low-pass filter to prevent high-frequency changes in the commanded attitude. This attitude correction relative to vertical is combined with the nominal commanded attitude that defines the roll of the vehicle around the local vertical to determine the total commanded attitude.
**Thruster Control**

The thruster control logic is divided into three discrete parts: the ACS, the descent engine control, and EGC control. The ACS determines how to fire the small attitude control thrusters to achieve the desired attitude. The descent engine logic determines when to fire the three descent engines to track the vertical position and velocity commands. The ACS control system throttles the EGC to target a thrust equal to five-sixths of the Earth weight of the vehicle.

The lander ACS is based on phase plane logic similar to that used on previous APL missions, including STEREO (Solar TErrestrial RElations Observatory). The direct inputs to the ACS are the commanded and estimated vehicle quaternion, the commanded and estimated angular velocities, and the angular acceleration that is estimated by filtering the measured velocity. The commanded quaternion is calculated by the lateral guidance law, and the commanded angular velocity is always set to zero on the prototype. The angular error is also integrated to allow integral control. This allows the ACS to mitigate the effects of one-sided dead banding resulting from thrust torques. A weighted sum of the angular position, velocity, acceleration, and integrated angular error is calculated and projected onto the pitch axis of each of the ACS thrusters. For each thruster, if the projected error exceeds a threshold, that thruster is commanded to fire. Additional logic in this phase plane control limits the maximum angular rates, either by always commanding a thruster to fire if the angular rate is below a negative threshold in its thrust direction or by preventing firing if its angular velocity exceeds a second threshold.

The three descent engines are controlled in unison by a single vertical fire command. The inputs to this logic block are the commanded vertical position and velocity, the estimated vehicle position and velocity from the navigation system, and the commanded input mode for either vertical position or velocity mode. When the vehicle is in vertical position control mode, the vertical velocity command is forced to zero. When the vehicle is in vertical velocity mode, the vertical position error is nulled. The vertical position and velocity errors are sent into phase plane logic that compares a weighted sum of these errors with a threshold. Hysteresis is added to the system by changing the threshold on the basis of whether the thrusters are currently firing. This hysteresis prevents rapid pulsing of the descent engines. Similar to the ACS control, additional logic limits the magnitude of the vertical velocity by always commanding a fire if the vehicle is descending too quickly or preventing firing if the vehicle is ascending too quickly.

Because an actual lunar lander would not have a thruster to cancel the difference between lunar and Earth gravity, the EGC control is performed outside of the primary GNCA control by the GNBC software block. This allows the GNCA algorithms to control a lander experiencing simulated lunar gravity. The EGC control module uses a lookup table to determine the throttle valve position on the basis of the estimated mass of the vehicle that is calculated from the initial mass and the estimated mass of the propellant used.

**Optical Navigation**

Several optical navigation strategies to estimate ground relative velocity, demonstrate autonomous rendezvous and capture, and identify landing hazards have been explored using the Mighty Eagle lander. The optical navigation algorithms tested on the lander are designed to run at the lowest priority level to prevent interference with the flight control algorithms and safety algorithms that determine abort criteria.

As part of the larger RLLDP effort, APL has continued to expand its expertise with optical navigation algorithms to estimate both terrain relative position and velocity by using passive optical cameras. These efforts have built on existing APL experience and algorithms from past studies for small body and lunar landings. To demonstrate the feasibility of these algorithms on flight computers as part of a complete navigation package, during 2010–2011, the team integrated a software module that performs one of its algorithms, Least Squares Optical Flow (LSOF), to estimate lateral velocity with a downward-looking camera on the Mighty Eagle. This type of capability offers a low-mass and low-cost alternative to Doppler velocimeters. The LSOF algorithm uses the common image processing technique of gradient-based optical flow to calculate warping matrices between two successive images. The algorithm first uses the estimated position and orientation of the camera at each image to determine an initial warping estimate. The optical flow algorithm then uses the image gradients and changes in intensity between the two images to calculate a correction to this warping. Several versions of the algorithm were explored that use different numbers of degrees of freedom for the correction warping. One version calculates an eight-degree of freedom correction to the warping. Assuming a planar surface, this warping allows estimation of the change in relative position and orientation of the camera between the two images and also the normal of the ground plane. A modified three-degree of freedom version estimates only the change in position, assuming the attitude is adequately known from independent sensors. Offline testing of the LSOF algorithm required approximately 3 s on a MCP750 running at 233 MHz to calculate a solution. While also running the GNC software on the same processor, the LSOF algorithm ran at a lower priority and required only 1 s longer. During tests on the Mighty Eagle vehicle, with a lower clock speed of 133 MHz and additional software modules to perform...
extensive data logging that would not be present on an actual mission, the LSOF algorithm required up to 11 s to calculate a solution.

In 2012, an Autonomous Rendezvous & Capture (AR&C) algorithm was added to the flight software by engineers at MSFC. AR&C is a cFE application that was developed using Embedded MATLAB and blocks from the Computer Vision System Toolbox from MathWorks. MathWorks’ Real-Time Workshop was used to auto-generate C code for integration into the flight software. The AR&C algorithm identifies a known target of four white circles in the landing area. After image acquisition of a 768×768 image, the image is thresholded into a binary image. Blob analysis is used to determine the locations and properties of the white blobs against the black background. The target is located by matching the blobs to the expected target shape. A Newton–Raphson iterative algorithm is then used to determine the target location by comparing the shape of the target in the image with the known image pattern. Finally, the determined target position is used to determine a guidance position command to provide to the GNCA algorithm to land on the target. After the AR&C algorithm was tuned for efficiency, it was able to generate one solution approximately every 1.7 s. This was more than adequate to demonstrate the concept.

Currently, during 2013, the Mighty Eagle team is working to demonstrate hazard avoidance. For this effort the illunis camera will be replaced by a stereo camera. Hazards such as boulders, slopes, and craters will be recognized from the stereo image, and safe landing sites will be identified. Hazard avoidance is much more computationally intensive than AR&C, so a dedicated image processor is needed. A commercially available laptop, the Panasonic Toughbook 31, will be installed on the lander as the image processor. The laptop will run cFE under Linux and will acquire the stereo images, generate the disparity map, find hazard-free landing sites, output the calculated landing site coordinates to GNCA, and log data. The laptop will communicate with the cFE applications on the SwRI avionics unit via the cFE software bus network. To the best of our knowledge, this will be the first time that the cFE software bus network will be used in flight.

**PREFLIGHT TESTING AND PREPARATIONS**

Before the actual flight tests of the Mighty Eagle lander, a variety of tests of the hardware and software systems of the vehicle, shown in Fig. 8, were performed to ensure proper functioning of the system. Before integration with the vehicle structure, acceptance tests of the propulsion system were performed to evaluate both individual thrusters and the integrated propulsion system. Additional hot-fire tests were performed after the integration of the propulsion system with the vehicle. Finite...
element models of the different vehicle structure configurations were developed to analyze vehicle stiffness and strength. The performance of the lander legs was demonstrated through drop tests to demonstrate the legs’ strength and shock absorption capability. A wind tunnel test was performed on a scale model of the lander to characterize the disturbance torques and forces on the vehicle from relative wind on the vehicle. Popellant slosh tests were performed to characterize the motion of the liquid propellant during flight. Several vibration table tests were also done to characterize the performance of the IMU vibration isolation mounts. Once the vehicle was assembled, a polarity test of the navigation system was also performed to ensure proper mounting and communications between the various system components.

The GNC engineers developed a high-fidelity simulation of the lander system that includes estimated sensor and actuator performance, including misalignments and known disturbances such as wind and propellant motion. By using the system knowledge and uncertainty estimates from the various component tests and system requirements, the team performed a wide variety of Monte Carlo simulations to demonstrate the robustness of the system to the full range of expected variability in flight conditions and lander configuration. Additional hardware-in-the-loop simulations were also used by the flight software, ground system, and mission operations teams to aid in the development and testing of the ground and flight software interfaces, the integration and testing of the ground and flight software, and development of the flight test operations procedures. During software integration and testing, the team embraced the concept of “test like you fly, fly like you test.” The entire life cycle for ground system and mission operations was exercised, including software installation and configuration, command and telemetry database builds, command and telemetry display development, command script development, operations product validation, real-time command and telemetry processing, telemetry playback, telemetry data archiving, and postflight data retrievals, including telemetry plots and history reports. Before the first actual Mighty Eagle test flight, the flight operations team was able to practice and refine the flight procedures used for the actual flight tests.

**FLIGHT TESTING**

After the completion of the final preflight tests in March and April of 2011, the Mighty Eagle team transitioned from the development stage to the flight testing stage. To date, there have been three successful flight test campaigns, with a fourth campaign currently under way in 2013.

**Indoor Flight Testing**

The first set of six successful flight tests was performed during June and July of 2011 at an indoor test facility at the Redstone Test Center in Huntsville, Alabama. These initial tests, summarized in Table 2, were designed to demonstrate stable flight of the vehicle and the ability to perform a variety of vertical and lat-

<table>
<thead>
<tr>
<th>Flight no.</th>
<th>Flight time (s)</th>
<th>Target altitude (m)</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>0.55</td>
<td>Tethered low hover, demonstrate stability</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>0.5</td>
<td>Tethered low hover, demonstrate “land now” abort</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>1.0</td>
<td>Low hover, demonstrate untethered stability</td>
</tr>
<tr>
<td>4</td>
<td>33</td>
<td>2.0</td>
<td>Longer hover flight over range of fill levels</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>0.9</td>
<td>4 m lateral translation at 0.5 m/s</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>5.0</td>
<td>High ascent with varying descent rates</td>
</tr>
</tbody>
</table>

![Figure 9. Tethered indoor flight test. (NASA)](image-url)
eral maneuvers. The first two flights were performed with the vehicle tethered to the ground, allowing low altitude flight but limiting vehicle travel. These tethered test flights were adopted as part of the nominal checkout procedure to demonstrate stable flight after any software or hardware configuration changes to the vehicle. In the first flight, illustrated in Fig. 9, the primary GNC system, GNCA, autonomously controlled the vehicle through the entire flight. In the second test, a manual “Land Now” soft-abort command was issued to test the functionality that commanded the backup GNCB system to land the vehicle.

The third flight was the first untethered flight of the vehicle. During all of these flights, the vehicle demonstrated good attitude control, with the angle from vertical around each axis staying below 1.2° while the primary GNC was controlling the vehicle. During the first two flights, it was observed that within approximately 0.1 m (0.33 ft) of the ground, the effectiveness of the vertical thrusters was reduced from ground effects. To mitigate against this effect, the lander was placed on small aluminium riser blocks starting with the third flight to increase thrust during takeoff. During the first three flights, the vehicle also experienced a vertical drift, indicating a bias in the navigation system. This drift was traced to several contributing factors, including a small, uncompensated bias in the accelerometer resulting from IMU vibrations and a large number of erroneous altimeter measurements caused by the altimeter sensing debris disrupted by the thrust. Additional tuning to GNC parameters on subsequent flights mitigated these effects.

During the second set of three indoor flight tests, shown in Figs. 10–12, the flight envelope of the vehicle was expanded to demonstrate the flight capabilities of the vehicle within the constraints of the indoor facility. The fourth flight extended the flight time to 33 s to demonstrate stable control over a larger range of propellant fill levels. In the fifth flight, shown in Fig. 11, the vehicle was commanded to perform its first lateral translation maneuver of 4 m (13 ft) at a rate of 0.5 m/s (1.6 fps). In the sixth and final indoor test flight, shown in Fig. 12, the vehicle was commanded to ascend to 5 m (16 ft) and then descend with an initial rate of –1.7 m/s (–5.6 fps) and then slow to –1.0 m/s (–3.3 fps) before touching down.
After the completion of indoor testing, the Mighty Eagle team transitioned operations to an outdoor test range also at the Redstone Test Center. The technical objective of the outdoor tests, flown during September through November of 2011, was to increase the flight envelope of the vehicle to include higher-rate translations at up to 2 m/s (6.6 fps), descents from 30 m (98 ft), and a 90° slew of the vehicle around the vertical axis before touchdown. A secondary objective of the outdoor test flights was to demonstrate optical velocity estimation on the vehicle. Similar to the indoor test sequence, initial outdoor tests were performed using tethers to verify vehicle performance and operation of the flight termination sequence. After these initial checkout tests, the flight envelope was gradually increased, leading to a lateral 10 m (33 ft) translation while descending from 30 m (98 ft) to approximate a terminal lunar descent, as shown by the flight profile in Fig. 13.

The largest obstacle during the outdoor test was a gradual decrease in thruster performance during later flights, resulting from gradual catalyst degradation that is

---

**Figure 12.** Sixth and final indoor test flight to 5 m altitude. (NASA)

**Figure 13.** Flight profile of outdoor flight to 30 m with lateral translation.

**Outdoor Flight Test Campaign 2011**

After the completion of indoor testing, the Mighty Eagle team transitioned operations to an outdoor test range also at the Redstone Test Center. The technical objective of the outdoor tests, flown during September through November of 2011, was to increase the flight envelope of the vehicle to include higher-rate translations at up to 2 m/s (6.6 fps), descents from 30 m (98 ft), and a 90° slew of the vehicle around the vertical axis before touchdown. A secondary objective of the outdoor test flights was to demonstrate optical velocity estimation on the vehicle. Similar to the indoor test sequence, initial outdoor tests were performed using tethers to verify vehicle performance and operation of the flight termination sequence. After these initial checkout tests, the flight envelope was gradually increased, leading to a lateral 10 m (33 ft) translation while descending from 30 m (98 ft) to approximate a terminal lunar descent, as shown by the flight profile in Fig. 13.

The largest obstacle during the outdoor test was a gradual decrease in thruster performance during later flights, resulting from gradual catalyst degradation that is
not uncommon with hydrogen peroxide systems. This degradation was visible during multiple flights when the exhaust plumes of the thrusters became visible from incomplete decomposition of the propellant. Figure 14 illustrates two outdoor flight tests, one before visible exhaust plumes and one with visible plumes. Although the reduced thrust did not compromise the safety of the vehicle, it did prevent several flights from fully achieving their objectives. During one flight with a larger propellant load, the vehicle had insufficient thrust to lift off, and during two flights, manual aborts were declared as a precaution when large amounts of visible exhaust limited visibility of the vehicle. The demonstration of the optical velocity algorithms was also limited by degraded propellant decomposition. The image processing code successfully ran on the flight processor and communicated with the navigation filter, although the velocity measurements were autonomously rejected because of poor image quality resulting from visible thruster exhaust in the field of view. In mid-November, a checkout and refurbishment of the catalyst beds was performed, and the final two test flights demonstrated the restoration of the system to its nominal performance. Overall, the outdoor test sequence proved to be very successful in demonstrating the flight capabilities of the vehicle. The sequence of performed outdoor tests during 2011 is summarized in Table 3.

**Outdoor Flight Test Campaign 2012**

With completion of GNC system validation from the 2011 flight test series, internal NASA independent research and development funding was obtained to demonstrate additional optical guidance in 2012. This work built on a long history of using AR&C for docking, capture, and berthing operations at MSFC. Applications for AR&C on a free-flying lander include satellite servicing, debris mitigation, sample return, and near-Earth object proximity operations. By using the nadir camera on the lander, an AR&C algorithm was added to the flight software. The AR&C target consists of four circles, as shown in Fig. 15. With knowledge of the diameters and the relative positions of the target circles, the image processing algorithm solves for the target in camera coordinates. The AR&C algorithm then uses the position estimate from GNCA and transforms the target position from the camera to topocentric coordinates. This target position is then passed to GNCA as its commanded position. This architecture allows full reuse of the validated GNCA flight code, conserving limited project resources, while demonstrating the new capability of in-flight adjustments to vehicle trajectory. A summary of the AR&C flight testing is shown in Table 4. During initial flights, the AR&C algorithm was run in an open-loop mode where the AR&C optical target detection algorithm ran onboard the vehicle but did not feed any commands to the GNCA module that followed a predefined trajectory. During later flights, the algorithm was run closed loop, and the target position was sent to the GNCA module as its guidance command.

In addition to demonstrating the AR&C capability on the lander, during the 2012 flight test campaign, the Mighty Eagle lander flight operations began to transi-
to a new team of engineers. Several team members with existing experience took leadership roles, and several new, young engineers were brought onto the team. This transition demonstrated the value of using the lander both to train and engage young engineers with hands-on experience and also to demonstrate sensors and capabilities that could extend beyond the primary mission of lunar landing.

**Ongoing Flight Test Campaign**

In 2013, the Mighty Eagle team was awarded additional NASA independent research and development funding at the MSFC to extend the AR&C effort to demonstrate hazard avoidance. The plan is to use a commercial stereo camera to identify hazards, including boulders, slopes, and craters, in a simulated lunar terrain field. Hazard avoidance flight testing is under way in 2013.

**CONCLUSIONS**

APL and the MSFC collaborated to develop and flight test the Mighty Eagle robotic lander test bed. The team has flown multiple flight test campaigns starting

---

Table 3. Mighty Eagle flight profiles performed during outdoor testing in 2011

<table>
<thead>
<tr>
<th>Objectives</th>
<th>Maximum altitude (m)</th>
<th>Flight time (s)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tethered checkout of infrastructure</td>
<td>0.8</td>
<td>12.0</td>
<td>Successful</td>
</tr>
<tr>
<td>Tethered checkout of flight termination system</td>
<td>0.8</td>
<td>12.0</td>
<td>Successful</td>
</tr>
<tr>
<td>1 m height, 10 m translate at 1 m/s</td>
<td>1.0</td>
<td>30.0</td>
<td>Unsuccessful (auto-aborted take-off after missed avionics packet)</td>
</tr>
<tr>
<td>0.6 m height, 10 m translate at 1 m/s</td>
<td>0.6</td>
<td>30.0</td>
<td>Successful</td>
</tr>
<tr>
<td>1 m height, 10 m translate at 2 m/s, slew 90°</td>
<td>1.0</td>
<td>42.0</td>
<td>Unsuccessful (insufficient thrust)</td>
</tr>
<tr>
<td>1 m height, 10 m translate at 2 m/s, slew 90°</td>
<td>1.0</td>
<td>30.0</td>
<td>Successful</td>
</tr>
<tr>
<td>Ascend to 10 m, hover, translate 10 m at 2 m/s, descend at (2, 1) m/s</td>
<td>10.0</td>
<td>30.0</td>
<td>Successful</td>
</tr>
<tr>
<td>Ascend to 10 m, hover, translate 10 m at 2 m/s, descend at (2, 1) m/s</td>
<td>10.0</td>
<td>17.0</td>
<td>Partially successful (manual abort commanded at 10 m)</td>
</tr>
<tr>
<td>Ascend at 0.5 m/s, hover 6 s, descend at 0.5 m/s</td>
<td>1.0</td>
<td>10.0</td>
<td>Successful</td>
</tr>
<tr>
<td>Ascend to 10 m, hover, translate 10 m at 2 m/s, descend at (2, 1) m/s</td>
<td>10.0</td>
<td>30.0</td>
<td>Successful</td>
</tr>
<tr>
<td>Ascend to 30 m at 3.7 m/s, translate 10 m while descending at (3.7, 2, 1) m/s, brief hover at 1 m</td>
<td>30.0</td>
<td>27.0</td>
<td>Successful</td>
</tr>
<tr>
<td>Ascend to 10 m, translate 10 m while descending to 2 m, ascend to 10 m, descend back to starting point with brief hover before touchdown</td>
<td>10.0</td>
<td>17.0</td>
<td>Partially successful (soft touchdown and manual abort after half maneuver)</td>
</tr>
<tr>
<td>Tethered checkout of catalyst bed refurbishment</td>
<td>0.8</td>
<td>13.0</td>
<td>Successful</td>
</tr>
<tr>
<td>Ascend to 10 m, hover, translate 10 m at 2 m/s, descend at (2, 1) m/s</td>
<td>10.0</td>
<td>30.0</td>
<td>Successful</td>
</tr>
</tbody>
</table>

Figure 15. Mighty Eagle descending to the AR&C target. (NASA)
in the summer of 2011 that have gradually demonstrated increased capabilities of the lander and descents from as high as 50 m (164 ft) to simulate the terminal descent to the lunar surface. In addition to serving as a system demonstration for technologies needed for an actual lunar mission, the Mighty Eagle development effort has also served as a catalyst for developing an integrated team of APL and MSFC engineers, providing young engineers with valuable hands-on experience and also providing an extensible platform that has been used to demonstrate additional mission concepts and sensor systems.

ACKNOWLEDGMENTS: A large number of talented team members across multiple disciplines and organizations contributed to the prototype lander and larger RLLDP efforts during the several years of development and flight testing. Although not all of the team members can be listed here, the authors would like to acknowledge some of the team members with whom they regularly interacted during the development and testing of the Mighty Eagle. During the initial development and flight tests, Julie Bassler was the MSFC program manager, and Larry Hill was the MSFC deputy program manager. Brian Mulac was the lead engineer of the prototype development. Brian Morse and Dewey Adams were the APL program leads through various stages of the program. At APL, several GNC engineers, including David Carrelli, Adam Fosbury, Jim Kaidy, Murty Challa, Stella Shapiro, and Wayne Dellinger, supported GNC algorithm development, verification, and testing. Tom Criss developed the initial prototype version of the LSOF algorithm, and Wen-Jong Shyong kept our development and autocode generation environment running smoothly. Gail Oxton, Scott Martucci, and Bob Davis supported development of the GNC wrapper code and integration and testing of the autogenerated GNC code on embedded processors. Doug Reid implemented the LSOF algorithm in C code, and Justin Thomas developed the camera control and emulation software. Paul Lafferty managed installation of the MIGS ground software and provided support. Sanae Kubota and Ben Ballard provided systems integration support. The mechanical team included Terry Betenbaugh (lead analyst), Deva Ponnusamy (leg design and testing), Gordon Maahs (leg analysis and testing), Marc Briere (nonlinear analysis), and a design team of Bill French (ATK) and Darrius Pergosky. A Huntsville-based team of MSFC and VCSI engineers, including Patrick O’Leary, Mike Turbe, John Wetherbee, Josh Eliser, and Dan Gunter, developed the lander operations procedures and screen displays, developed the backup GNC system, and integrated and tested the full GNC software build in the vehicle flight software. Scott Gilley was the avionics lead during the lander development, Kyle Daniel was the safety lead, Todd Freestone was the communications lead, and Charlie Adams led the alignment and weight and balance for the lander. Robert Harris led the original red crew that was responsible for vehicle flight preparations. Mark Wells, of Dynetics, was the lead propulsion designer. Other red team members included T. J. Talty, Michael Thomas, and Wayne Neumair, who also provided propulsion engineering support. Jason Adams is the current lead engineer for ongoing Mighty Eagle testing.

<table>
<thead>
<tr>
<th>Flight</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tethered 14 s flight time, 0.8 m max altitude, functional checkout, EGC throttle motor errors</td>
</tr>
<tr>
<td>2</td>
<td>Retest of flight, 14 s flight time, 0.8 m max altitude, expand understanding of temperature-dependent EGC throttle motor errors</td>
</tr>
<tr>
<td>3</td>
<td>AR&amp;C open loop, 32 s flight time, 10 m max altitude, 10-in.-diameter circular targets, no AR&amp;C solutions were generated, noted dirty lens</td>
</tr>
<tr>
<td>4</td>
<td>AR&amp;C closed loop, 32 s flight, 10 m max altitude, nine AR&amp;C solutions generated, image processing parameters updated before this flight</td>
</tr>
<tr>
<td>5</td>
<td>AR&amp;C open loop, 36 s flight, 30 m max altitude, new target (24-in.-diameter circles), 10 AR&amp;C solutions generated</td>
</tr>
<tr>
<td>6</td>
<td>AR&amp;C closed loop, 36 s flight, 30 m max altitude, 11 AR&amp;C solutions generated, culmination of AR&amp;C testing</td>
</tr>
<tr>
<td>7</td>
<td>Tethered, 10 s flight, 0.8 m altitude, test of “land now” in GNCB with descent rate a function of altitude</td>
</tr>
<tr>
<td>8</td>
<td>Envelope expansion flight (higher, faster, longer), 43 s flight, 51 m max altitude, ascend at 6 m/s</td>
</tr>
<tr>
<td>9</td>
<td>Tethered, 14 s flight, 0.8 m altitude, change to lightweight legs</td>
</tr>
<tr>
<td>10</td>
<td>Student Launch Initiative demonstration, 34 s flight, 30 m max altitude, first ascent with downrange translation component</td>
</tr>
</tbody>
</table>
REFERENCES

Timothy G. McGee led the GNC development for the Mighty Eagle lander. He is a member of the Senior Professional Staff at APL. Dr. McGee's technical areas of interest include guidance and control of autonomous systems, image processing, optical navigation, and robotic systems. David A. Artis and Timothy J. Cole were the software and mechanical leads, respectively, for the Mighty Eagle. Douglas A. Eng is currently the Group Supervisor of the Space Systems Applications Group within the Space Department at APL and provided systems engineering support for the Mighty Eagle. Cheryl L. B. Reed is the current APL Program Manager for the RLLDP. Michael R. Hannan provided additional GNC support for the project at MSFC. D. Greg Chavers and Cynthia D. Stemple currently lead the RLLDP efforts at MSFC. Logan D. Kennedy and Joshua M. Moore are engineers at MSFC who have provided systems engineering support throughout RLLDP. For further information on the work reported here, contact Timothy McGee. His e-mail address is timothy.mcgee@jhuapl.edu.

The Authors


Autonomous Geolocation of RF Emitters Using Small, Unmanned Platforms

Robert J. Bamberger, Jay G. Moore, Ravi P. Goonasekeram, and David H. Scheidt

Autonomous geolocation of RF emitters using small, unmanned systems is a game-changing technology for military, government, and commercial missions. This technique employs a novel application of a common RF direction-finding technique called pseudo-Doppler. Emergent autonomous control concepts are used to control the sensor platform and optimize flight trajectories for efficient and rapid geolocation of the target. The basic components of this concept, from sensor development to unmanned system autonomous behaviors, were tested in simulation and subsequently demonstrated in flight during the Tactical Network Topology experiment.

OVERVIEW

The mission set for geolocation of RF emitters extends across military, governmental, scientific, and commercial domains. Examples include finding adversaries’ radios, tracking RF-tagged wildlife, and searching for shipboard emergency beacons. Another application, to be addressed with future research, is the use of this technique for navigation of small unmanned aircraft systems (UASs) in areas where Global Positioning System (GPS) is jammed. Classic methods for RF-emitter geolocation include ground-based direction-finding (DF) techniques and Doppler-based airborne techniques. However, ground-based systems can be labor intensive, often require long-duration sensing, and may not have the ability to access areas that result in a solution. Doppler techniques onboard high-value airborne assets, whether manned or unmanned, require human-in-the-loop control and use sensor platforms from higher echelons that may be too expensive or difficult to schedule and typically lack stealth.

APL has developed a technique using small, inexpensive, autonomous UASs that can be organic to a squad of soldiers, a small team of first responders, or private commercial interests. Furthermore, this approach requires only supervisory control but no human in the loop for vehicle control, sensor processing, or geolocation solutions. Because the sensor platforms are small, they can be more easily made stealthy, and because they are airborne they can access areas from which ground-based systems may be restricted.

Small UASs are true force multipliers, providing soldiers or first responders organic mobile sensor platforms that can provide close-in sensing while remaining
stealthy. These next-generation UASs work individually or as autonomous, multi-vehicle collaborative units and can operate as “fire and forget” resources requiring very little human intervention for control (usually only take-off and landing). APL’s unique physics-based approach to UAS autonomy has been successfully demonstrated in dozens of flight test experiments, with mission objectives ranging from unattended ground sensor (UGS) data exfiltration and relay, to vision-based ground vehicle tracking, to chemical-biological agent plume characterization.\(^1\)\(^2\)

However, small UASs do not fly at velocities sufficient to produce Doppler shifts that result in accurate geolocation solutions. One method that does not require a high-velocity platform is known as pseudo-Doppler, also called synthetic Doppler, which is employed by both amateur radio enthusiasts and law enforcement.\(^3\) In the pseudo-Doppler approach, the receive signal is switched rapidly between a constellation of antennas, and the phase difference is measured to determine line of bearing (LOB).\(^4\)\(^5\)

For this effort, multiple receive antennas and a sensor payload were integrated onboard each UAS. COTS hardware was adapted to receive the signals, and custom hardware was developed to switch between antennas and to measure the phase shift that produced LOBs. Miniaturization of the hardware enabled implementation onboard a small UAS platform (less than 160 cm wingspan and less than 3.2 kg gross vehicle weight).

APL’s autonomy algorithms enable multiple UASs, each deriving its own LOB solution, to optimize their own flight trajectories. Kalman filters implemented on each aircraft were used in the derivation of the geolocation solution and error ellipse based on the LOBs of the UAS team. Thus, each UAS was equipped with its own processing module to implement the autonomy and derive the geolocation solution, as well as a communication module to exchange data with the other UASs and send the solution information to the user on the ground. Each UAS also had an onboard autopilot that supported true unmanned flight of the UAS solely on the basis of the direction provided to it by the onboard autonomy software; no human in the loop was required to direct the UAS flight.

ONBOARD KALMAN FILTER AND GEOLOCATION SOLUTION PROCESSING

To obtain a target geolocation solution, it is necessary to combine two or more LOB measurements from different look angles to the target RF emitter.\(^6\) This can be accomplished with a single sensor platform, but the solution is arrived at more quickly, and the error smaller, using two or more platforms (balancing resource constraints with the geolocation solution error, three vehicles seem to be optimal for this application). This is especially true for moving RF emitters.

For this effort, two UAS vehicles were used as the sensor platforms (although not optimal, two were considered adequate for proof of concept, although one can also provide a solution). Each vehicle implemented an onboard Kalman filter for fusing LOB measurements into a geolocation solution. Each vehicle broadcast its measured LOB values, along with the time and location of the measurements, to the wireless network. This allowed each vehicle to independently fuse its own measurements with the measurements of the other UASs. No terrain elevation information was available to the vehicles, so a flat-earth assumption was made, and the target RF emitter geolocation was computed in the horizontal plane only. Because the flight altitudes were low relative to standoff distance, altitude was assumed to be the same as the terrain elevation at the vehicle launch site. Each filter was initialized using the first available LOB measurement and an assumption about the maximum range at which the target might be detected. This information was used to construct an uncertainty region oriented along the LOB, with a 95% confidence interval in the range direction reaching from the vehicle position to the assumed maximum range.

Because the process by which the vehicle and target emitter positions define a LOB (i.e., the observation model) involves nonlinear trigonometric functions, a linear Kalman filter was not suitable for the task of fusing LOBs. In early simulation, an extended Kalman filter, which linearizes a nonlinear process model about some operating point, was implemented and tested but was found to diverge egregiously under conditions of high noise because of the degree of nonlinearity. An unscented Kalman filter, which uses a limited form of sampling to approximate nonlinearities to second-order accuracy, was found to converge reliably even when the LOB measurement noise reached a standard deviation of 90º.

A constant process model was used in the Kalman filter because, for these experiments, the target was nominally stationary. However, a non-zero process noise matrix was used. This served two purposes: first, it allowed the filter to better recover from unmodeled biases in the LOB measurements by discounting old measurements in favor of new ones; and second, it allowed for the tracking of moving targets, even with no prior assumptions about the direction or pattern of motion. Tests in simulation demonstrated success in tracking a moving target both with a constant speed and direction and with randomly changing direction.

LOB measurements were retained in a buffer in chronological order for a short time before they were fused in the filter. This allowed time for the data from all the UASs to arrive and be inserted into the buffer in the correct order before being fused, eliminating the need...
to “roll back” the filter to an earlier time to incorporate older data. Fused measurements were retained in the buffer for another short period so they could be broadcast to the network multiple times, reducing the chances of lost data and increasing the degree of agreement between the vehicle target estimates. Each UAS uses its own LOB measurements, as well as the LOB measurements of the other UAS, to estimate target location.

SOFTWARE DESIGN

The Java-based software implementation of the UAS autonomy was developed from a system of related subsystems, including the agent system and the belief network interface.

Agent System

At the center of the implementation is the agent system. This subsystem has interfaces to the sensor interface, the autopilot, the Kalman filter, and the belief network. It acts as a data conduit and processing system. The UAS behaviors are also implemented in this subsystem.

Belief Network Interface

The agent system interfaces with a virtual blackboard known as the belief network. This blackboard is made up of all the belief managers spread across the network. The belief managers attempt to automatically and efficiently synchronize and update the beliefs held in the blackboard. For this effort, two sensor beliefs were added to the legacy belief network. They represent the LOB output from the onboard sensor package and the uncertain target geo - locations. These are called “RangeBearingSensorBelief” and “UncertainTargetBelief,” respectively.

RangeBearingSensorBelief represents a time-indexed list of all sensor readings performed by any agent. For efficiency, the belief drops any sensor reading older than a certain decay time. This decay time is configurable at run-time. UncertainTargetBelief holds the results of the sensor data beliefs and geo location uncertainties of each individual agent. The geo location uncertainty is represented by an error ellipse about the derived geo location solution. This belief is used to display ellipses on a modified version of the standard display tool.

Custom Optimization Behaviors

Two custom behaviors were created for this effort: GhostCircularFormationBehavior and AngularDiversityTrackBehavior. Each behavior attempts to guide the aircraft in a certain direction. The agent system combines the results of these behaviors into a single command sent to the autopilot. At certain times, the system weighs some behaviors more than others, on the basis

MISSION-BASED AUTONOMY CONCEPT AND SOFTWARE DESIGN

Autonomy Concept

Previous APL research efforts pioneered the use of potential field theory to achieve an effects-based control of multiple UASs. This approach is based on insect models of cooperation and coordination. Problems are solved by heterarchically, rather than hierarchically, organized swarms that alter and react to the environment. Instead of centralized control, decision making is decentralized, occurring with each member of the swarm.

This sharing of knowledge is the cornerstone of the potential fields concept. The transmitted data packets contain information such as the vehicle’s situational awareness (e.g., sensor data), operational status (e.g., location in space), or mission parameters (e.g., user commands). These data packets are communicated over a wireless local area network (WLAN) and are referred to as “beliefs.” A communications framework was developed for belief transfer that employs a modular multi-layered architecture. This framework was designed to facilitate distributed collaboration over any mobile ad hoc network (MANET).

The potential fields are generated as a result of the worldview of the UAS, which is itself the totality of the UAS beliefs. These fields are used to influence vehicle action, most notably movement. The forces associated with these fields, which are illustrated in Fig. 1, may be attractive (directing the vehicle toward a point), repulsive (directing the vehicle away from a point), or complex (a combination of attractive and repulsive fields). At any given time, the total force on a vehicle is the summation of all attractive, repulsive, and complex forces due to all known influences.

Software Design

The Java-based software implementation of the UAS autonomy was developed from a system of related subsystems, including the agent system and the belief network interface.

Agent System

At the center of the implementation is the agent system. This subsystem has interfaces to the sensor interface, the autopilot, the Kalman filter, and the belief network. It acts as a data conduit and processing system. The UAS behaviors are also implemented in this subsystem.

Belief Network Interface

The agent system interfaces with a virtual blackboard known as the belief network. This blackboard is made up of all the belief managers spread across the network. The belief managers attempt to automatically and efficiently synchronize and update the beliefs held in the blackboard. For this effort, two sensor beliefs were added to the legacy belief network. They represent the LOB output from the onboard sensor package and the uncertain target geo - locations. These are called “RangeBearingSensorBelief” and “UncertainTargetBelief,” respectively.

RangeBearingSensorBelief represents a time-indexed list of all sensor readings performed by any agent. For efficiency, the belief drops any sensor reading older than a certain decay time. This decay time is configurable at run-time. UncertainTargetBelief holds the results of the sensor data beliefs and geo location uncertainties of each individual agent. The geo location uncertainty is represented by an error ellipse about the derived geo location solution. This belief is used to display ellipses on a modified version of the standard display tool.

Custom Optimization Behaviors

Two custom behaviors were created for this effort: GhostCircularFormationBehavior and AngularDiversityTrackBehavior. Each behavior attempts to guide the aircraft in a certain direction. The agent system combines the results of these behaviors into a single command sent to the autopilot. At certain times, the system weighs some behaviors more than others, on the basis

![Figure 1. Fields that are (a) attractive, (b) repulsive, and (c) complex.](image-url)
assumptions, the steering commands maximize the instantaneous improvement in geolocation accuracy for two cooperating vehicles, and the resulting behavior generalizes in an intuitive way to more than two vehicles. The simplified geometry for two-vehicle geolocation is depicted in Fig. 2.

Two UASs labeled Sensor 1 and Sensor 2 are shown on the left-hand side of the figure independently measuring LOB to the target with some error, resulting in the uncertainty area, shaded blue, within which the target is most likely to be found. This geometry has been simplified in the right-hand side of the figure to approximate the area of uncertainty as a trapezoid, greatly simplifying the expression for the area of the uncertainty region, \( A \):

\[
A = \frac{w_1 w_2}{\sin |\Delta \beta|} \frac{(r_1 \delta_1)(r_2 \delta_2)}{\sin |\Delta \beta|}. \tag{1}
\]

The control law will choose the direction (course) that each UAS will travel, denoted by \( \psi_1 \) and \( \psi_2 \); the UAS velocities \( V_1 \) and \( V_2 \) are assumed fixed (and will be shown later not to impact the computed course). The difference between \( \psi \) and \( \beta \) is defined as \( \theta \). This geometry is illustrated in Fig. 3.

If it is the case that the signal might be lost at any time, the best policy is, at each point in time, to choose to steer in the direction that maximizes the rate at which
the uncertainty shrinks, because the next sensor reading may be the last. At each iteration of the control law, the courses will be chosen so as to maximize the rate at which $A$ shrinks, that is, we wish to minimize $\dot{A}$, the time rate of change of $A$, with respect to $\psi_1$ and $\psi_2$. First we must derive an expression for $\dot{A}$:

\[
\dot{A} = \frac{\delta_1 \delta_2 (r_1 \dot{r}_2 + r_2 \dot{r}_1) \sin |\Delta \beta| - r_1 r_2 \cos |\Delta \beta| \| \dot{\beta} |}{\sin^2 |\Delta \beta|}
\]

\[
= \frac{\delta_1 \delta_2}{\sin^2 |\Delta \beta|} \left[ (r_1 V_2 \cos \theta_2 + r_2 V_1 \cos \theta_1) \sin |\Delta \beta| \| \mp r_1 r_2 \cos |\Delta \beta| \| (\dot{\beta}_1 - \dot{\beta}_2) \right]
\]

\[
= \frac{\delta_1 \delta_2}{\sin^2 |\Delta \beta|} \left[ (r_1 V_2 \cos \theta_2 + r_2 V_1 \cos \theta_1) \sin |\Delta \beta| \| \pm r_1 r_2 \cos |\Delta \beta| \left( \frac{V_1 \sin \theta_1}{r_1} - \frac{V_2 \sin \theta_2}{r_2} \right) \right]
\]

To choose $\psi_1$, we set $\partial \dot{A}/\partial \theta_1 = 0$ and solve for $\theta_1$:

\[
\frac{\partial \dot{A}}{\partial \theta_1} = \frac{\delta_1 \delta_2}{\sin^2 |\Delta \beta|} \left[ r_2 V_1 \sin \theta_1 \sin |\Delta \beta| \| \pm r_1 r_2 \cos |\Delta \beta| \left( \frac{V_1 \cos \theta_1}{r_1} \right) \right]
\]

\[
= 0
\]

\[
\tan \theta_1 = \pm \cot |\Delta \beta| \left[ \frac{\Delta \beta + \frac{\pi}{2}}{\Delta \beta - \frac{\pi}{2}} \Delta \beta > 0 \right] \]

\[
\theta_1 = \begin{cases} 
\frac{\Delta \beta + \frac{\pi}{2}}{\Delta \beta - \frac{\pi}{2}} & \Delta \beta > 0 \\
\frac{\Delta \beta - \frac{\pi}{2}}{\Delta \beta + \frac{\pi}{2}} & \Delta \beta < 0 
\end{cases}
\]

The steering command is then:

\[
\psi_1 = \beta_1 + \theta_1
\]

\[
= \beta_1 + \Delta \beta \pm \frac{\pi}{2}
\]

An analogous process is used to find $\psi_2$. When $\Delta \beta = 0$, the vehicles will each choose at random whether to use $\theta_1 = \Delta \beta + \pi/2$ or $\theta_1 = \Delta \beta - \pi/2$. This situation is extremely unlikely to arise in real-world applications, however, and is really a concern only in simulation where the vehicles can be initialized to exactly the same position.

Notice that $\theta_1$ depends only on $\Delta \beta$, so the velocities, distances from the target, and uncertainty in the LOBs do not affect the course command.

This control policy for two vehicles can easily be generalized to more than two vehicles by using a potential field approach. In concept, the procedure is to express the influence of each vehicle on every other as a potential field, sum the fields, and steer each vehicle down the gradient of the potential field. In practice, however, it is easier not to explicitly construct the potential field but to work directly with the gradients from the start; because gradient is a linear operator, the gradients resulting from the influence of each vehicle on a given teammate can be found individually and summed.

If the potential field gradient due to the influence of vehicle $j$ on vehicle $i$ is $\nabla \phi_{ij}$, then the potential field equivalent of the above control law is:

\[
\nabla \phi_{ij} = \cos \theta_{ij} \hat{x} + \sin \theta_{ij} \hat{y},
\]

where $\theta_{ij} = \beta_i - \beta_j$.

The influence of all $N - 1$ other vehicles on vehicle $i$ is:

\[
\nabla \phi_i = \sum_{j=1}^{N} w_{ij} \nabla \phi_{ij}
\]

\[
= \sum_{j=1}^{N} w_{ij} (\cos \theta_{ij} \hat{x} + \sin \theta_{ij} \hat{y})
\]

640
where $w_{ij} = \begin{cases} 1 & i \neq j \\ 0 & i = j \end{cases}$.

The steering command is then found from:

$$
\theta_j = \tan^{-1} \left( \frac{\nabla \phi_j \cdot \hat{y}}{\nabla \phi_j \cdot \hat{x}} \right) \\
= \tan^{-1} \left( \frac{\sum_{j=1}^{N} w_{ij} \sin \theta_{ij}}{\sum_{j=1}^{N} w_{ij} \cos \theta_{ij}} \right), \quad \psi = \beta + \theta_1
$$

Figure 4. Procerus Unicorn research airplane.

Figure 5. Onboard hardware, software, and communications architecture.

**HARDWARE DEVELOPMENT AND SYSTEMS INTEGRATION**

**Aerial Vehicle and Autopilot**

The UAS platforms that were employed for this effort were Unicorns with 153-cm wingspans from Procerus Technologies (see Fig. 4). The small size, low power, and light weight of the sensor and control payload developed for these platforms demonstrate that this technique could be implemented on a fielded military or commercial UAS of similar, or even smaller, size. The UAS autopilot is a Kestrel Autopilot v.2.22, also from Procerus Technologies. These autopilots contain three-axis angular rate and acceleration sensors, a three-axis magnetometer, a barometric altimeter, 20-point sensor temperature compensation, GPS, wind estimation, and a dead-reckoning filter for GPS-denied operation. For this effort, the autopilot was controlled over a serial interface by the onboard autonomy module. The ground station was used only to collect position data for posttest analysis, as well as to provide flight safety.

**Sensor Payload and Control System**

The sensor payload consists of the antenna system, the RF receiver, the antenna switch circuit integrated with the LOB processor module, and a PIC-based interface board. The control system consists of a COTS XScale Reduced Instruction Set Computer (RISC) processor board with integrated WLAN plug-in card. The onboard architecture is shown in Fig. 5.

**Sensor Payload**

The sensor payload consists of an array of four custom-built antennas, a COTS radio receiver, and custom processing and switching electronics. The radio is a multiband, 900-channel handheld Yaesu VX-6 transceiver that was operated in receive mode only.
In the UHF band that was used to test this geolocation concept, the radio has a sensitivity of 0.2–0.5 µV for 12 dB signal-to-noise and distortion ratio.

The radio receiver output is characterized using a custom circuit that compares the phase of the incoming signal from the four antennas and, from that phase comparison, generates an LOB value. This measurement provides the LOB in discrete steps of 0.3927 radians (22.5°). A separate PIC processor-based board converts this output to the proper format and sends the data over a serial interface to the main control system processor board. The LOB processor board also contains the antenna switch circuit that sends the antenna switching signals to a very small antenna controller board that is collocated with the antennas.

The antenna system is an array of four wire antennas, each with a small ground plane, arranged in a square pattern. The greatest challenges in the payload systems integration effort were the spacing, tuning, ground plane fabrication, impedance matching, and orientation of the antennas. Most of these challenges are a result of the small size of the platform. The size constraints on the antenna system ground plane resulted in the most significant effects to system performance.

Although a COTS transceiver was used for this proof-of-concept effort, future enhancements may include the development of a much smaller receive-only module with similar receive performance to the Yaesu radio. Four of these modules could be used to receive the RF emissions from each antenna simultaneously. Other possible future enhancements include the use of digital signal processing technology in the LOB processor and including the PIC processor data interface on the same board. These enhancements will not only reduce the size, weight, and power budget of the sensor payload but will also significantly improve performance.

**Proof-of-Concept Simulations**

Models of two core concept pieces were developed, and simulations were performed to establish proof of concept. One model represented the autonomy and was used both to test and to help develop the autonomy software. The other model represented the geolocation sensor payload, the Kalman filter, and the geolocation algorithms.

The autonomy model demonstrated the theoretical trajectories flown by each vehicle given a stable geolocation solution process. Figure 6 shows convergence to the target for different numbers of UASs and UAS starting position. Once in proximity to the target, they set up circular orbits around the target at fixed phase differences depending on the team size. For instance, for two airplanes, the phase difference is 90°, and for three airplanes it is 120°. (An example of a simulated two-vehicle team orbit is shown in Fig. 11b.)

The geolocation and Kalman filter algorithms were simulated using the second model. A single UAS was used, with the payload sensor resolution and errors included in the model. Figure 7 shows four snapshots of the simulation. At $t_1$, data collection has only recently begun, and there is little angular diversity in the data, so the error ellipse is quite large. Nevertheless, the center of the ellipse is in close proximity to the actual target. At

---

![Figure 6. Two, three, and 10 UASs converge to target in simulation.](image_url)
each airplane on the basis of self-knowledge and communications from the other UAS. That is, all geolocation and autonomy algorithms were performed independently of ground processing.

The experiment that was demonstrated at Camp Roberts consisted of two parts: (i) geolocation of an actual RF emitter with two airplanes flying preset loiter patterns and (ii) demonstration of the converge and orbit behaviors using a virtual RF emitter (location hard-coded in the onboard autonomy code). Ultimately, it is the combination of these parts that results in realization of the full geolocation concept—that is, geolocation of the actual RF emitter, and vehicle convergence to and orbit of, that emitter. However, this requires real-time calibration of the LOB with the vehicle bearing, and the constrained demonstration schedule did not allow for this calibration step. A posttest calibration was performed for one of the UASs, which resulted in the red line shown in Fig. 8. This calibration was derived by plotting true LOB versus measured LOB collected over two and a half loiter orbits; the loiter point was approximately

Figure 7. Snapshot of geolocation solution from a single UAS.

Figure 8. Calibration line (red) derived by plotting true LOB versus measured LOB.

FLIGHT DEMONSTRATIONS

To test this geolocation concept, a series of bench tests, hardware-in-the-loop tests, and field tests were conducted at APL facilities, a nearby leased field, and the U.S. Army’s Aberdeen Proving Ground. A final concept demonstration was conducted at Camp Roberts, California, during a Tactical Network Topology exercise; Tactical Network Topology is a quarterly series of experiments hosted by the Naval Postgraduate School and U.S. Special Operations Command.

Two airplanes were used during these tests. As shown in simulation, two airplanes provide a good geolocation solution in a reasonable time period. Each vehicle was outfitted with the full sensor and control payload described previously. Also as previously described, derivation of a geolocation solution and determination of flight behaviors were achieved separately on
snapshots are roughly equally spaced throughout the data collection time period, so t₀ is the initial reading at 0 s, t₁ is at t₀ + 145 s, t₂ is at t₀ + 290 s, and t₃ is the final error ellipse at t₀ + 435 s.

The plot in Fig. 10 also shows convergence to a geolocation solution over time. At the end of the 435-s data collection period, the error between estimated target location and true target location was 60 m.

Because of the calibration issue, the data in Fig. 10 and the 60-m error vector were derived from data fused from only a single aircraft loitering 550 m from the target. The flight path provided little angular diversity (a maximum of 29° with respect to the target), and data collection was over a relatively short time period (435 s). On the basis of simulation, when this data set is extended to three airplanes circling the target at the 550 m distance but 120° out of phase with respect to each other, the error collapses to 12 m and the error settling time illustrated in Fig. 9 is reduced to 30 s. Enhancements to the sensor payload and real-time calibration are expected to result in even smaller errors. Also, bringing the three airplanes closer to the target further reduces error. For instance, if the three airplanes circle the target at a distance of 100 m, the geolocation error is theoretically reduced to 2.2 m.

The second part of this experiment demonstrated the vehicle behaviors that were shown in simulation. A "virtual" RF emitter location was sent over the wireless network from a ground node to both UASs. That is, both UASs were provided the geolocation solution. This experiment, therefore, was intended to demonstrate the autonomous behaviors.

Just as in simulation, both UASs converged on the target and set up orbit around the target 90° out of phase from each other. These behaviors are shown in Figs. 11a and 11b, with the simulation trajectories shown on the left-hand sides next to the corresponding flight trajectories shown on the right. As can be seen, the actual flight paths are virtually identical to the simulation trajectories. This experiment presents strong evidence that the onboard autonomy and belief management system directed the UASs to behave as expected.

### SUMMARY

These simulations and flight experiments demonstrate the feasibility of using RF sensors onboard multiple individual aircraft operating cooperatively as a quick, inexpensive, and reliable method of geolocating radio signal emitters. The technology behind the RF sensors was adapted from a technique known as pseudo-Doppler: the target signal was received on a constellation of four rapidly switched onboard antennas, and the phase difference was measured onboard to derive LOB. The LOB values from multiple airplanes were combined to estimate the geolocation solution.
The onboard autonomy algorithms guided the UASs and optimized trajectories for the best solution. All sensor operation, signal processing, geolocation algorithms, and autonomy were performed onboard the aircraft—that is, the airplanes were operated with no human in the loop except for preflight loading of mission objectives and ground observation of the real-time solution estimation. Although some development tasks remain, this effort provides a proof of concept for geolocation of RF emitters from small, autonomous aerial platforms.

REFERENCES


The Authors

Robert J. Bamberger is a member of the APL Principal Professional Staff and Section Supervisor in the Robotics and Autonomous Systems Group of the Research and Exploratory Development Department (REDD). He acted as Principal Investigator, systems integrator, and test lead for this independent research and development (IR&D) effort. Jay G. Moore is an APL Senior Staff member and Assistant Section Supervisor in the Advanced RF Capabilities Group in the Asymmetric Operations Sector. Jay developed all of the geolocation algorithms and Kalman filters used for the geolocation solution. Jay also developed the simulation tools used for creation of the geolocation software. Ravi P. Goonasekeram is a member of the APL Principal Professional Staff in the RF & Optical Systems Section of REDD’s Electrical Engineering Systems Analysis and Design Group. Ravi acted as RF subject matter expert (SME) and lead hardware engineer. He also designed and fabricated the miniature high-performance ground beacon, as well as the onboard circuitry that derived LOB on the basis of measured received signal phase shift. David H. Scheidt is a member of the APL Principal Professional Staff and is Program Manager in the Science Applications Group in the Space Sector. David first developed the autonomy concepts that were used to control the UAS path planning and behaviors. He acted as autonomy SME for this IR&D effort. For further information about the work reported here, contact Robert Bamberger. His e-mail address is robert.bamberger@jhuapl.edu.

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.
APL has a rich history of developing autonomous systems. Examples include Buckybot (left), which shows preliminary positive results for indoor mapping applications; and Mighty Eagle (right), part of a new generation of small, autonomous robotic landers developed in conjunction with the Marshall Space Flight Center (MSFC) to land on airless bodies in our solar system including the moon and asteroids.