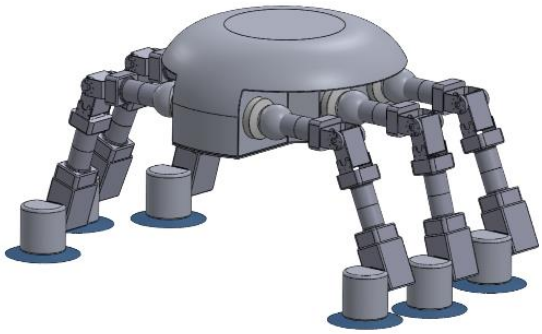
NASA's Gateway to Blue Skies Competition
RepAir: Advancing Aircraft Maintenance

South Dakota State University



Surface **P**reservation **A**nd **R**ust **K**iller (S.P.A.R.K.) Crawler

Team Members:

Teyla Hanson
Sophomore
Mechanical Engineer

Meloray Linderer
Sophomore
Mechanical Engineer

Noah Richardson
Sophomore
Mechanical Engineer

Alexis Smith
Sophomore
Mechanical Engineer

John Worth
Freshman
Mechanical Engineer

Faculty Advisor: Dr. Todd Letcher, Associate Professor
Department of Mechanical Engineering, Jerome Lohr College of Engineering

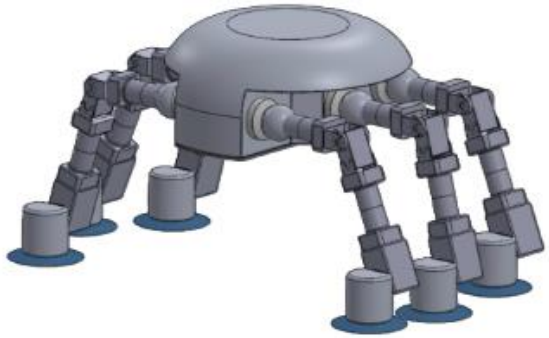
S.P.A.R.K. Crawler

(Surface Preservation And Rust Killer)

Project Summary:

- Corrosion Control.
- Aging and heavily used aircrafts experience varying levels of pitting corrosion, leading to risks in the safety and structural soundness of aircrafts.
- Repairing corrosion through autonomous programs, the S.P.A.R.K. Crawler will autonomously repair corrosion identified through visual inspection drones already integrated into maintenance.

Project Image:



Team Composition/Roles:

South Dakota State Club Team

- Alexis Smith: So M.E. – Team Lead
- Noah Richardson: So M.E.- Mobility Lead
- Meloray Linderer: So M.E. – Corrosion Lead
- Teyla Hanson: So M.E.- Coding Lead
- John Worth: Fr M.E.- Chassis Lead

The team is experienced in identifying issues and creating solutions through prototyping and testing a variety of different rovers, UAVS, and other mechanical products.

Proposed deployment timeline:

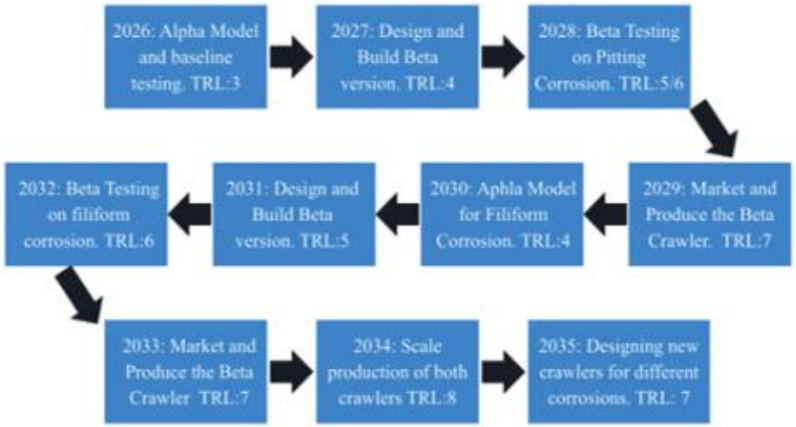


Table of Contents

Executive Summary

Corrosion remains a $2.2 billion annual “silent killer” in the aviation industry, driving extensive manual labor in hazardous environments and contributing to costly aircraft downtime. Addressing this persistent challenge requires a fundamental shift in how maintenance is performed. The **S**urface **P**reservation **A**nd **R**ust **K**iller (S.P.A.R.K.) Crawler is an autonomous, bio-inspired robotic platform designed to transform corrosion treatment from a reactive, labor-intensive process into a proactive, “set-and-forget” operation. Drawing inspiration from the stability of formicary locomotion, the system employs a six-legged alternating tripod gait combined with active suction adhesion, enabling reliable traversal of complex, vertical, and even inverted aircraft surfaces with a robust 9.6 factor of safety.

The S.P.A.R.K. system operates through an integrated “Detect-and-Destroy” architecture. Autonomous thermographic drones first identify and map subsurface corrosion, transmitting high-resolution repair paths to the crawler via a centralized Home Base. Once deployed, the crawler executes a fully autonomous restoration sequence, including precision grinding, debris removal, chemical sealant application, and repainting, without human intervention. By combining advanced robotics with scalable, swarm-capable autonomy, the system significantly reduces technician exposure to hazardous materials while delivering projected 10-year savings of $1.2 million per aircraft. This proposal outlines a phased development path from the current alpha prototype to a fleet-ready TRL 8 system, positioning the S.P.A.R.K. Crawler as a transformative step toward intelligent, autonomous aircraft health management.

1. Introduction

Corrosion is one of the most persistent and costly challenges in commercial aircraft maintenance, requiring extensive manual labor, prolonged aircraft downtime, and increased safety risk to technicians. The S.P.A.R.K. Crawler addresses this challenge by transforming corrosion treatment from a labor-intensive process to an autonomous, surface-level operation. The S.P.A.R.K. Crawler brings the solution directly to the problem, robotically.

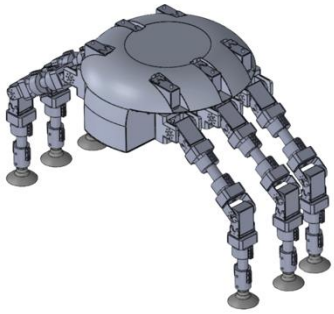


***Figure 1**: S.P.A.R.K. Crawler*

Inspired by the efficiency and stability of an ant, S.P.A.R.K. Crawler utilizes a six-legged locomotion system that allows it to move deliberately and securely across complex aircraft geometries, including vertical and inverted surfaces. Each leg is equipped with active suction, enabling stable adhesion while the crawler autonomously removes corrosion, applies protective sealant, and sprays paint to restore treated areas. By integrating inspection, repair, and surface restoration into a single robotic platform, S.P.A.R.K. Crawler reduces maintenance time, improves consistency, and significantly enhances technician safety. This system represents a shift from reactive maintenance to intelligent, automated aircraft surface care.

2. Situation Assessment

Corrosion is the gradual deterioration of metal caused by chemical or electrochemical reactions with its surrounding environment, including exposure to air, moisture, or other reactive agents. When inspecting an aircraft structure, it is essential to examine both exterior and interior surfaces for signs of corrosion. Although corrosion maintenance is often tedious, it is critical to the aircraft's safety and long-term performance. Maintenance personnel spend countless hours conducting detailed inspections to meet strict regulatory requirements. According to the Federal Aviation Administration (FAA) [28], maintenance workers must observe numerous precautions during corrosion removal and follow emergency safety procedures if paint removers, chemicals, or other liquids come into contact with the skin or eyes. They are also required to wear personal protective equipment that shields the eyes, skin, and respiratory system and to be trained in safety protocols. Potential chemical intoxication starts the moment the maintenance workers step onto the aircraft. Aircraft paint coatings contain hexavalent chromium and isocyanates, which, when ingested, cause serious respiratory issues. The main form of ingestion occurs when workers grind and sand the aircraft while repairing and assessing corrosion [16]. Each maintenance employee is required to remove street clothes and shower to remove the contaminants; they must document the number of hours spent around the chemicals and provide cardiopulmonary tests every six to twelve months to monitor their respiratory system.

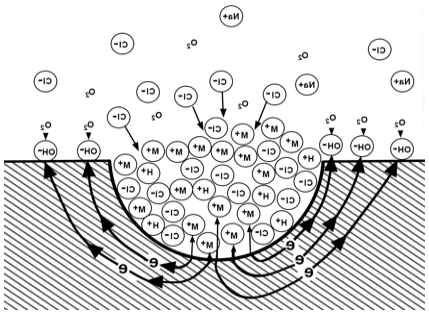


***Figure 2**: Corrosion Diagram*

***Figure 3**: Pitting Corrosion*

Beyond safety knowledge, inspectors must also recognize the primary forms of exterior corrosion affecting aircraft: pitting, filiform, and uniform surface corrosion. Pitting corrosion produces small cavities or holes in the metal [24], while filiform corrosion develops on poorly prepared aluminum surfaces beneath polyurethane paint, appearing as fine, worm-like lines that can progress to bubbling and flaking. Uniform surface corrosion occurs when exposed metal reacts with oxygen in the air, such as in areas where protective paint has worn away from the wing skin or fuselage [5]. Pitting corrosion is a critical concern in aircraft maintenance due to the structural weakening it causes, its hidden progression beneath protective coatings, and the potential for galvanic acceleration [5]. Because pits can form underneath paint, they may remain invisible without proper inspection, and even a single pit can initiate cracks that ultimately compromise structural integrity. Several

detection methods are used in practice, including visual inspection, dye penetrant testing, and thermal or X-ray scanning. Visual inspection is limited to early or obvious surface damage. In contrast, dye-penetrant testing is a non-destructive technique that reveals surface cracks and pits but requires extensive surface preparation and cleaning [16]. Thermal or X-ray methods provide the most effective detection, allowing maintenance personnel to identify corrosion beneath coatings and determine whether repair or component replacement is necessary. Mitigating pitting corrosion on an aircraft's exterior generally involves three key steps. First, the affected surface is prepared through grinding, deep scoring, or abrasive blast cleaning to remove corrosion and contaminants [17]. Next, protective coatings and treatments are applied to improve resistance to future corrosion [29]. Last, the surface is refinished and painted to restore a smooth protective barrier that blends with the surrounding aircraft structure [20]. Pitting corrosion is particularly difficult to detect, predict, and prevent because corrosion products often conceal the pits themselves and appears as small, narrow cavities with minimal overall metal loss, yet it can still lead to the failure of an entire engineering system [24]. This type of corrosion is aggressive in aluminum and commonly develops in environments with elevated moisture and chloride levels [11].

Through interviews with maintenance personnel, several key challenges in aircraft maintenance were identified, including a shortage of personnel, lengthy inspection processes, and significant time required to complete inspections. These factors often delay maintenance operations and the return of aircraft to service. S.P.A.R.K. is designed to reduce the time technicians spend on repetitive, time-intensive inspection and minor repair tasks, allowing personnel to focus on other critical maintenance needs. Compared to traditional corrosion inspection and removal methods, S.P.A.R.K. introduces several key advantages. Current practices rely entirely on manual labor, requiring extensive time for surface preparation, inspection, and repair, whereas S.P.A.R.K. streamlines these processes through automation, significantly improving efficiency. The system also enhances worker safety by minimizing direct exposure to hazardous substances, such as hexavalent chromium and isocyanates, and reducing reliance on prolonged use of personal protective equipment. By enabling earlier detection and more consistent treatment of corrosion, S.P.A.R.K. can lower long-term maintenance costs and extend the service life of aircraft components. These combined improvements position S.P.A.R.K. as a meaningful advancement over existing corrosion maintenance practices. S.P.A.R.K. Crawler is designed to target pitting corrosion, as this form presents a greater risk to aircraft structural integrity than other corrosion types.

3. System Architecture

Visual inspection remains the most widely used method for detecting corrosion; however, the industry is increasingly transitioning toward autonomous, data-driven approaches. Advanced inspection drones offer improved accuracy and efficiency while reducing the likelihood of human error [26]. In 2024, Delta Air Lines received FAA approval to deploy autonomous drones for fleet-wide lightning damage inspections, demonstrating the growing maturity and acceptance of this technology [26]. These systems integrate high-resolution RGB imaging, thermal infrared sensing for subsurface detection, structured-light scanning for precise 3D surface reconstruction, and AI-enabled classification and auto-routing capabilities [9]. When applied to corrosion detection, such technologies enable rapid identification of affected areas, quantification of damage extent, and assessment of severity. The S.P.A.R.K. Crawler leverages similar drone-based systems to enable fast, accurate detection of surface defects, forming the foundation of an autonomous maintenance workflow.

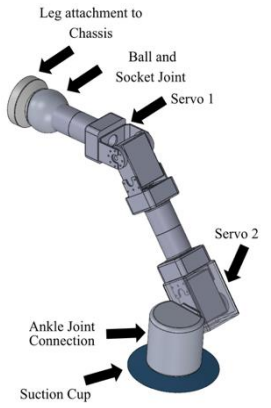
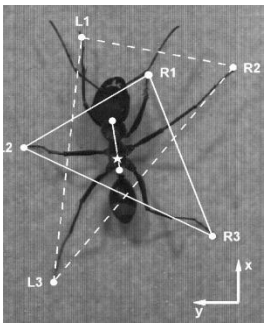


***Figure 4**: Crawler Leg*

The complete S.P.A.R.K. system consists of two primary components: a centralized Home Base that provides computational and control capabilities, and the autonomous crawler responsible for corrosion repair (Figure 1). Inspection drones first survey the aircraft and identify corrosion hotspots, transmitting this data to the Home Base for processing. Each identified location is mapped onto a high-fidelity 3D model of the aircraft, and the resulting coordinates are used to generate optimized traversal paths for the crawler.

This information is transmitted through a wired connection, allowing simultaneous data transfer and charging while ensuring high accuracy and minimizing cybersecurity risks. Before operation, a technician removes the crawler from its charging station, places it on the aircraft surface, and initiates the system, providing a layer of human oversight for safety and verification. The crawler adheres to the aircraft using six active suction cups, enabling traversal across horizontal, vertical, and inverted surfaces. Using onboard SLAM-based localization, it follows precomputed paths while maintaining continuous stability through a biomimetic alternating tripod gait, in which three legs remain in contact while the other three reposition.



igure 4**: Ant Locomotion [16]**Figure

The system is designed to withstand worst-case loading conditions, including fully inverted operation beneath the aircraft fuselage. With all six suction cups engaged, the crawler achieves a factor of safety of approximately 9.6 considering its fully loaded weight. During locomotion, when only three suction cups are actively supporting the system, a safety factor of 4.78 is maintained. Even in the event of a single suction cup failure during operation, the crawler retains a safety factor greater than 3, meeting minimum design requirements for safe operation. These performance characteristics are enabled by a locomotion strategy inspired by ant movement in natural environments, where stability is maintained across complex, curved, and inverted surfaces. By coordinating leg motion in alternating tripod groups, the crawler ensures its center of mass remains within a stable support polygon at all times. This biomimetic approach reduces dynamic loading on individual suction interfaces, enhances stability on complex aircraft geometries, and enables smooth, controlled motion throughout the repair process.

Air will be delivered through tubing routed along each leg, with slip rings integrated at the knee joints to prevent pinching and maintain uninterrupted flow during articulation. The crawler navigates to each corrosion site using its onboard computing system, which communicates continuously with the Home Base. A downward-facing camera mounted on the underside provides real-time visual confirmation of the target location and supports precise alignment during repair operations. Upon arrival, the crawler engages all six suction cups to establish a stable attachment to the aircraft surface. It then deploys a retractable tool assembly equipped with a 30 mm grinding disk to remove the corrosion layer. At this point, all six suction cups are engaged and can allow for 70 pounds of downforce per cup (calculation in 9. Appendices). The underside camera is subsequently used to verify complete material removal before proceeding.

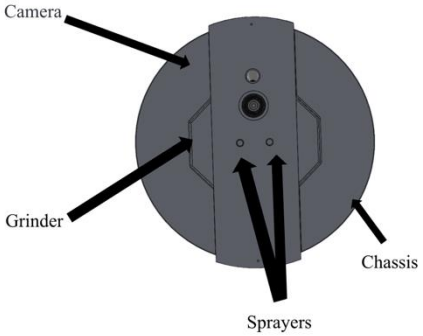


***Figure 5**: Underside of S.P.A.R.K.*

Once corrosion removal is confirmed, the grinding mechanism retracts, and an integrated air nozzle, fed by the onboard vacuum pump, clears residual debris from the surface. A secondary nozzle mounted on the underside then applies a protective sealant coating, with the capability to switch to paint application, as required. After completing the repair cycle, the crawler either advances to the next designated location or returns to the Home Base. The crawler features a compact aluminum shell measuring 0.36 m in diameter and 0.1 m in height, with legs extending up to 0.23 m, also constructed from aluminum for strength and weight efficiency. All critical subsystems, including the vacuum pump, power supply, onboard computer, and sealant and paint reservoirs, are housed within the main body. These reservoirs are accessible from the top of the unit, enabling straightforward maintenance and refilling between operations.

The S.P.A.R.K. Crawler has an estimated total mass of approximately 30 kg, including mechanical, electrical, and operational subsystems to balance structural integrity, mobility, and functional performance – see Table 1. The largest contributors are the motor assemblies, 20 kg, and the onboard vacuum pump, 8 kg, both of which are essential for active suction, leg actuation, and corrosion treatment processes. The chassis accounts for 6.75 kg, providing a rigid and compact structural backbone that supports all integrated subsystems. Additional components include the six-leg assembly, 2.3 kg, battery system, 2.5 kg, onboard electronics, 0.5 kg, and corrosion treatment tools, including a sprayer, 2 kg, and grinder,1 kg.

***Table 1**: Mass Allocation*

Mass of SPARK Crawler	
Componets	Mass (kg)
Chasis (Empty)	6.75
Legs (6)	2.3
Electronics	0.5
Batteries (2)	2.5
Sprayer	2
Grinder System	1
Vacuum Pump System	1.5
Total	30.55

This mass allocation is intentionally minimized to maximize adhesion factor of safety and ensure stable locomotion across complex aircraft geometries, including vertical and inverted surfaces. At the same time, the overall configuration preserves a compact and efficient form factor, enabling the system to navigate constrained areas while delivering consistent, high-performance corrosion detection and repair.

4. Concept of Operations

The S.P.A.R.K. Crawler is designed to augment and extend existing corrosion detection methods that rely on drones and advanced imaging technologies [28]. By leveraging these capabilities, the system can interpret corrosion detection data to accurately identify repair locations and autonomously generate optimized repair paths. This integration transforms detection data into actionable maintenance operations, enabling a seamless transition from inspection to repair. As a result, maintenance personnel are freed from time-intensive corrosion mitigation tasks and can instead focus on higher-priority structural and system-level work. This approach not only reduces overall labor demands but also improves efficiency and turnaround time in aircraft maintenance operations, positioning the S.P.A.R.K. system as a force multiplier within modern maintenance workflows.

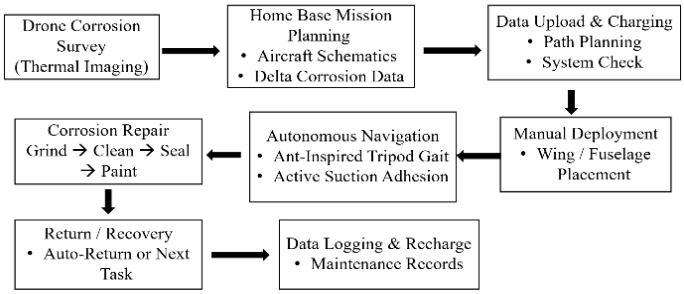


***Figure 6**: CONOPS for the S.P.A.R.K. Crawler, illustrating the end-to-end workflow from corrosion detection through autonomous repair and system recovery*

The Home Base serves as the central hub for the S.P.A.R.K. system, providing three core functions: mission planning, system communication, and charging. A critical role of the Home Base is managing data flow between the drone-based inspection system and the S.P.A.R.K. Crawler. It receives and processes high-resolution imagery from the drones, identifies corrosion sites, and generates optimized repair paths. These paths, along with precise location data, are transmitted to the crawler through a direct physical connection that also supports battery charging and software updates. This wired interface ensures reliable, high-integrity data transfer, reduces setup complexity, and establishes a consistent docking and readiness point when the crawler is not in use. The base is contoured to match the Crawler's geometry and features a secure plug-in interface that is manually connected by an operator, enabling a controlled transition from human oversight to autonomous operation. By integrating seamlessly with existing drone-based inspection workflows, the S.P.A.R.K. system eliminates the need for reconfiguration of current technologies while significantly reducing manual data handling and repair time.

Once corrosion positional data has been uploaded, a technician deploys the crawler at a specified initial location on the aircraft giving the crawler a starting frame of reference, typically the wing, after which it operates fully autonomously using a LiDAR scanner and SLAM algorithms. Each crawler is configured for a specific corrosion type, allowing for targeted, optimized repair strategies and enabling

multiple units to operate collaboratively in swarm configurations for large-scale maintenance tasks. The development roadmap envisions incremental capability expansion, introducing new crawler variants approximately every two years to address additional corrosion mechanisms and various scenarios following validation. The current system is optimized for pitting corrosion and is equipped with all necessary onboard tools, including a retractable grinding mechanism, vacuum-assisted debris removal and air delivery, a chemical sealant applicator, and a paint sprayer. Upon reaching a repair site, the crawler secures all six legs and lowers itself approximately 10 cm to precisely align its tools, minimizing mechanical strain and improving operational accuracy. A downward-facing camera verifies the target area before controlled grinding removes paint and corrosion, with real-time visual feedback guiding the process. The grinder is designed to operate effectively around fasteners and within irregular geometries, while its extendable configuration maintains a safe standoff distance to reduce leg articulation and prevent suction cup interference, particularly on vertical and inverted surfaces.

Following material removal, compressed air is used clearing debris and preparing the surface for treatment. The same system that applies chemical sealant, which neutralizes and protects the repaired area, is also used to apply a final paint layer to restore the aircraft surface. Each stage is governed by a calibrated timing sequence to ensure sufficient debris removal and proper curing of coatings. Upon completing its assigned repairs, the crawler either proceeds to the next location or returns to its deployment point for technician retrieval. A return indicates either mission completion or a low-battery condition, at which point the system notifies maintenance personnel via the Home Base. The Crawler is then manually removed and reconnected for charging, data transfer, and inspection. This operational framework enables efficient, largely autonomous maintenance, potentially conducted overnight, while reducing technician workload, improving safety, and maintaining human oversight at key control points.

5. Future and Beyond

5.1 Operational Integration and Stakeholder Implementation Strategy

The S.P.A.R.K. Crawler is designed for seamless integration into existing maintenance ecosystems, minimizing workflow disruption by operating within established corrosion inspection and repair processes rather than introducing entirely new methodologies. This design philosophy significantly lowers the barrier to adoption for maintenance organizations operating under strict procedural and regulatory constraints, enabling immediate value without requiring fundamental changes to current practices.

5.1.1 Stakeholder Ecosystem and Adoption Drivers

Key stakeholders include airline Maintenance & Engineering (M&E) teams, Aircraft Maintenance Technicians (AMTs), maintenance planners, and safety and compliance personnel. For M&E leadership, S.P.A.R.K. delivers measurable reductions in labor costs, aircraft downtime, and safety risk, directly supporting fleet efficiency objectives and cost-control pressures. For AMTs, the system removes one of the most hazardous and labor-intensive aspects of corrosion repair, shifting their role toward supervision and system oversight without necessitating extensive retraining. Maintenance planners benefit from the system's autonomous operation and digital logging capabilities, which enable more predictable scheduling, improved corrosion tracking, and better utilization of overnight or off-peak maintenance windows.

A major driver of adoption is the non-discretionary nature of corrosion maintenance. FAA-mandated Corrosion Prevention and Control Programs (CPCPs) require routine inspection and remediation across both commercial and military fleets, ensuring that operators are already committed to these activities and associated costs. S.P.A.R.K. provides a safer, more cost-effective method of meeting these existing compliance requirements, reducing adoption risk while accelerating return on investment without requiring changes to underlying maintenance philosophies.

5.1.2 Workflow Integration

S.P.A.R.K. integrates directly downstream of current corrosion detection workflows, including drone-assisted inspections, allowing inspection teams to retain existing identification and documentation procedures. After corrosion is identified, technicians deploy the crawler to autonomously execute standard remediation processes using approved materials and established repair methods, maintaining full compatibility with current maintenance records and practices. Human involvement remains focused on

deployment, monitoring, consumable replenishment, and basic troubleshooting, with training requirements limited to system setup and modular servicing.

5.1.3 Regulatory Pathway and Scalability

Because S.P.A.R.K. performs conventional repair operations using standardized, already approved materials and methods, certification efforts can focus on system reliability, fail-safe behavior, and documentation traceability rather than validating new repair methodologies. This targeted regulatory scope reduces both time-to-deployment and certification costs for early adopters. The platform's modular architecture also supports scalable implementation, enabling organizations to begin with single-unit deployments and expand incrementally to multi-unit or swarm-based operations as confidence grows. At the fleet level, consistent autonomous repair improves maintenance standardization and enables aggregation of corrosion trend data across aircraft, supporting predictive maintenance strategies and optimized inspection intervals.

Overall, S.P.A.R.K. enhances existing maintenance infrastructure by improving safety, efficiency, and regulatory compliance. Its alignment with established workflows, regulatory frameworks, and the non-discretionary demand driven by CPCP requirements positions it for near-term adoption while delivering compounding long-term value across the aircraft maintenance lifecycle.

5.2 Cost Analysis and Economic Impact

The S.P.A.R.K. Crawler is designed to significantly reduce aircraft maintenance costs and downtime through autonomous corrosion removal and surface restoration. Cost estimates, based on market-ready production systems utilizing commercial off-the-shelf (COTS) components, modular fabrication, and a standardized chassis shared across corrosion-specific variants, demonstrate strong economic viability. Prototype testing suggests that scaling production could reduce per-unit costs by approximately 30–40% through bulk procurement, standardized manufacturing, and subsystem reuse.

Under this model, the Gen-1 Pitting Corrosion Crawler is estimated to cost between $9,000 and $10,000 per unit, while the more advanced Gen-2 Filiform Corrosion Crawler is projected at $11,500 to $13,000, reflecting enhanced sensing capabilities and expanded surface-treatment functionality. Each unit is expected to have an operational lifespan of 5 to10 years under routine deployment across multiple aircraft. Annual operating and maintenance costs are estimated at $1,600 to $2,000, including consumables such as sealant, paint, and abrasive media, as well as routine servicing and modular component replacement enabled by the Crawler's maintainable subsystem architecture.

***Table 2**: Single Crawler ROI [23][25][27][29]*

Single Crawler — Lifecycle ROI (Per Aircraft)						
Metric	**Year 1**	**Year 2**	**Year 3**	**Year 5**	**Year 7**	**Year 10**
Gross savings ($)	$23,250	$46,500	$69,750	$116,250	$162,750	$232,500
Annual operating cost ($)	$2,000	$4,000	$6,000	$10,000	$14,000	$20,000
Net savings — annual ($)	$21,250	$42,500	$63,750	$106,250	$148,750	$212,500
Cumulative gross savings ($)	**$23,250**	**$46,500**	**$69,750**	**$116,250**	**$162,750**	**$232,500**
Cumulative operating costs ($)	**$2,000**	**$4,000**	**$6,000**	**$10,000**	**$14,000**	**$20,000**
Cumulative net savings ($)	**$21,250**	**$42,500**	**$63,750**	**$106,250**	**$148,750**	**$212,500**
Hours returned to technician	310 h	620 h	930 h	1,550 h	2,170 h	3,100 h
Cumulative hours returned	**310 h**	**620 h**	**930 h**	**1,550 h**	**2,170 h**	**3,100 h**

Estimated lifecycle ownership costs for the S.P.A.R.K. Crawler remain modest when compared to traditional corrosion maintenance expenditures. A single production unit is projected to incur approximately $18,000–$20,000 in total cost over five years, and $27,000–$30,000 over ten years, depending on utilization. These figures are minimal relative to the recurring costs of manual corrosion repair, which represent a substantial portion of overall aircraft maintenance spending [29][11].

5.2.1 Industry Cost Context and Adoption Drivers

Corrosion maintenance continues to be one of the most persistent and costly challenges in aviation. In the United States alone, direct corrosion-related costs are estimated at approximately $2.2 billion annually [29], driven by labor-intensive inspection processes, hazardous chemical treatments, and extended aircraft-on-ground (AOG) time. Individual repair events typically range from $15,000 to $30,000 per

aircraft when accounting for labor, access equipment, materials, and downtime [11]. These costs are further amplified by lost operational revenue, which can exceed $10,000 per hour for commercial aircraft [13].

Beyond direct financial impact, corrosion management is a regulatory necessity. FAA airworthiness directives and Corrosion Prevention and Control Programs (CPCPs) mandate routine inspection and remediation across both commercial and military fleets. This creates a consistent, non-discretionary demand for corrosion maintenance, ensuring that operators must already allocate resources to these activities. The S.P.A.R.K. system aligns directly with these existing requirements, offering a more efficient and cost-effective method of compliance without requiring changes to established maintenance philosophies. This alignment significantly reduces adoption risk and strengthens the economic case for implementation.

5.2.2 Safety and Workforce Impact

Traditional corrosion maintenance exposes technicians to a range of hazards that are difficult to fully mitigate. These include working at height on wings and fuselage crowns, operating in inverted or confined spaces, and handling abrasive materials and chemical sealants in environments with limited ventilation. OSHA data consistently identifies falls from elevation as a leading cause of serious injury and fatality in maintenance operations [15], while prolonged exposure to chemicals introduces additional long-term health risks.

The S.P.A.R.K. Crawler significantly reduces these risks by autonomously performing the most physically demanding and hazardous aspects of corrosion repair. Technician involvement is limited to system setup, supervision, and final inspection, effectively shifting the human role from direct manual labor to higher-level oversight. Over time, this transition yields measurable workforce benefits, including fewer injury-related absences, reduced workers' compensation costs, and improved retention in a field already facing skilled labor shortages.

5.2.3 Full Repair Lifecycle Cost Reduction

The economic value of the S.P.A.R.K. system extends well beyond the immediate remediation phase, delivering cost savings across the entire corrosion repair lifecycle. By enabling early detection and autonomous treatment of surface-level corrosion, the system helps prevent progression to structural corrosion, an advanced condition that often requires depot-level maintenance, component replacement, or extensive structural repair. Addressing corrosion at its earliest stage significantly reduces the likelihood of these escalated events, which can carry costs an order of magnitude higher than routine maintenance.

In addition, S.P.A.R.K.'s sensor-guided, autonomous operation minimizes the variability inherent in manual repair processes. Issues such as incomplete material removal, uneven sealant application, and performance degradation due to technician fatigue are substantially reduced. This consistency improves repair quality, lowers rework rates, and decreases the probability of premature corrosion recurrence. Over time, these improvements extend maintenance intervals and generate compounding cost savings throughout the aircraft's service life.

5.2.4 Return on Investment

***Table 3**: Task-by-Task Time & Cost Breakdown [6][11][25][27]*

Task-by-Task Time & Cost Breakdown (Per Aircraft / Year)						
Maintenance Task	**Manual (h/yr)**	**S.P.A.R.K. (h/yr)**	**Hours Saved**	**Cost Saved ($/yr)**	**% Reduction**	**Automation Method**
Surface inspection & scanning	80 h	8 h	72 h	$5,400	90.0%	**Autonomous**
Abrasive corrosion removal	120 h	12 h	108 h	$8,100	90.0%	**Autonomous**
Sealant & primer application	60 h	6 h	54 h	$4,050	90.0%	**Autonomous**
Paint restoration	80 h	4 h	76 h	$5,700	95.0%	**Autonomous**
System setup & oversight	0 h	24 h	—	—	—	**Technician**
TOTAL (automated tasks only)	**340 h**	**30 h**	**310 h**	**$23,250**	**91.2%**	

For single-Crawler deployment, conservative estimates project annual savings of approximately $22,000 per aircraft, driven by reduced labor demands, decreased aircraft downtime, and lower safety-related costs [28]. At this rate, the S.P.A.R.K. system is expected to reach cost break-even within the first year of operation, often during the initial corrosion maintenance cycle. Over a ten-year lifecycle, cumulative

net savings are projected to exceed $190,000 per aircraft, even after accounting for operating and maintenance expenses.

In higher-utilization scenarios, deploying multiple Crawlers in parallel further amplifies these benefits by reducing maintenance duration and minimizing operational disruption. A three-Crawler configuration is estimated to generate approximately $75,000 in annual savings. Over a ten-year period, such deployments yield projected net savings on the order of $650,000 to more than $1.2 million, depending on fleet scale and utilization intensity. Figure 7 illustrates these projected program-level savings across Single-Crawler, Three-Crawler Swarm, and Fleet Deployment scenarios, highlighting the strong economic case for scalable implementation.

5.2.5 Fleet-Level Scalability

The modular architecture of the S.P.A.R.K. platform enables a phased, scalable deployment strategy for maintenance organizations. Operators can begin with a single unit to validate performance within existing workflows and progressively expand to multi-Crawler swarm configurations as operational confidence and demonstrated reliability increase. This staged adoption approach reduces initial capital exposure, lowers implementation risk for maintenance providers, and allows the system to be appropriately scaled to fleet size, corrosion severity, and maintenance tempo without requiring redesign of the core robotic platform.

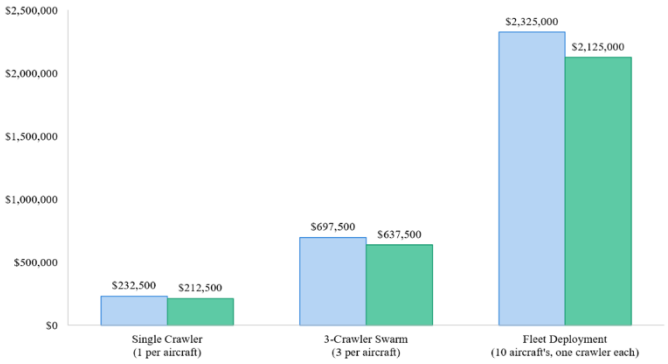


***Figure 7**: S.P.A.R.K. Crawler Gross and Net ROI*

At the fleet level, the combined effects of reduced aircraft downtime, improved repair consistency, and the prevention of corrosion escalation into structural damage position S.P.A.R.K. as a strategic maintenance asset rather than a conventional tooling purchase. Instead of functioning as a discrete maintenance aid, the system integrates into a broader lifecycle cost-reduction strategy, delivering compounding value over time.

Overall, the production-level S.P.A.R.K. Crawler architecture demonstrates strong long-term economic viability. By initially targeting high-frequency pitting corrosion and subsequently expanding into additional corrosion-specific variants built on a shared modular platform, the system establishes a scalable and cost-efficient framework for autonomous aircraft maintenance. This approach offers a clear pathway to reducing corrosion-related maintenance expenditures across commercial aviation fleets while improving operational consistency and reliability.

5.3 Timeline

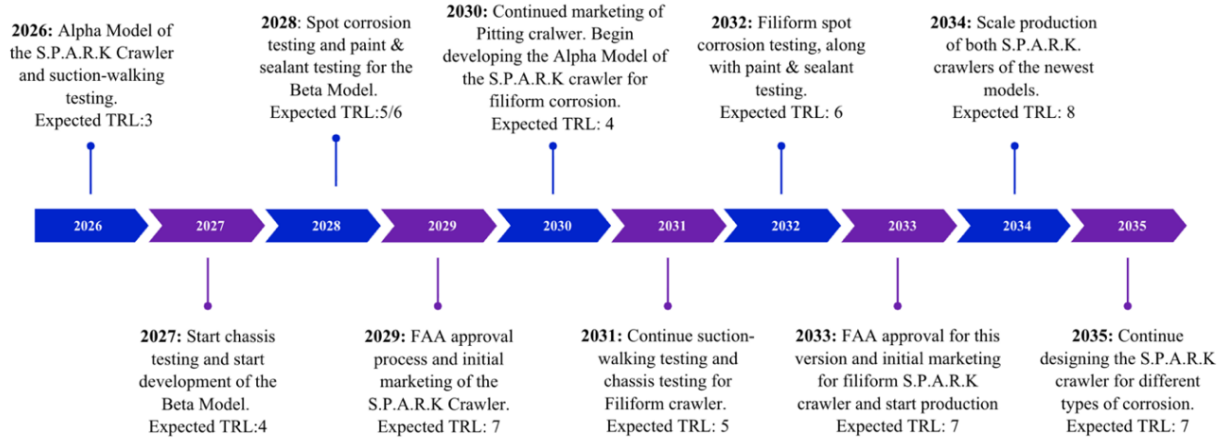


***Figure 8**: Timeline to Path to Deployment*

In Phases One and Two (2026–2027), development of the S.P.A.R.K. Crawler will focus on foundational mobility and low-risk functional validation. Testing will center on suction-cup locomotion across angled aircraft surfaces, followed by initial verification of the corrosion repair process on flat surfaces. These early experiments will establish baseline performance for adhesion, stability, and tool operation, with the system expected to achieve a Technology Readiness Level (TRL) of 3–4.

In Phase Three (beginning 2028), the Pitting Corrosion Beta Model will be introduced, with emphasis on optimizing localized corrosion detection and removal performance. The system will be evaluated across multiple commercial aircraft platforms under varied environmental conditions, addressing a range of pitting corrosion severities and locations. During this phase, swarm-based testing will also be conducted to evaluate inter-Crawler coordination, task distribution, and collective efficiency. Upon completion of validation testing, the FAA certification process will be started, with the system targeting TRL 5–6.

Phase Four (beginning 2029) marks the transition to early market deployment of the validated Pitting Corrosion Beta Crawler. At this stage, FAA approval is assumed at the onset of commercialization, and the system will continue to undergo iterative refinement based on field performance data and customer feedback. This phase is expected to achieve TRL 7.

In Phase Five (beginning 2030), development will expand to include the Alpha Model for filiform corrosion, incorporating operational insights and performance data from the pitting corrosion program. This new variant will begin at an estimated TRL of 4.

Phase Six (beginning 2031) will focus on testing the Filiform Corrosion Beta Model, following a methodology consistent with the earlier pitting corrosion validation program. Testing will include continued suction-based locomotion trials, expanded chassis evaluation, and initial swarm performance assessments across both Crawler types. This phase is expected to reach TRL 5.

In Phase Seven (beginning 2032), expanded validation of the Filiform Corrosion Beta Model will be conducted across a broader range of aircraft platforms and environmental conditions. The focus will shift toward optimizing corrosion treatment performance, including precision spot repair and integrated paint and sealant application. FAA certification efforts for the filiform variant will begin during this phase, with an anticipated TRL of 6.

Phase Eight (beginning 2033) will introduce FAA-approved deployment and market rollout of the Filiform Corrosion Beta Crawler. Similar to earlier phases, continuous refinement will be driven by operational feedback and real-world performance data, targeting TRL 7.

In Phase Nine (beginning 2034), production scaling and full market expansion will be pursued for both the Pitting and Filiform S.P.A.R.K. Crawler systems. At this stage, the combined platform portfolio is expected to achieve TRL 8.

Beyond 2035, development will continue with additional S.P.A.R.K. variants targeting other external aircraft corrosion mechanisms. These future systems will follow the established development and validation framework, entering the pipeline at an initial TRL of 4 and progressing through the same staged testing and certification pathway.

6. Conclusion

Corrosion remains a persistent, costly, and safety-critical challenge in commercial aviation, requiring extensive labor, specialized access equipment, and prolonged aircraft downtime. The S.P.A.R.K. Crawler addresses these challenges by shifting corrosion treatment from a human-centered, reactive process to an autonomous, surface-level operation. By integrating inspection, mobility, adhesion, material removal, sealing, and surface restoration into a single robotic platform, the S.P.A.R.K. Crawler fundamentally transforms how aircraft corrosion is managed, enabling safe and consistent maintenance even on vertical and inverted surfaces. Its biomimetic locomotion, active suction, and modular tool integration enable it to navigate complex geometries while maintaining robust safety factors under worst-case loading scenarios.

Beyond technical feasibility, the S.P.A.R.K. Crawler delivers measurable operational and economic benefits. Autonomous operation reduces technician labor, minimizes aircraft downtime, and

enables overnight repairs, providing a rapid return on investment while improving worker safety. The system's modular architecture and swarm-capable design support scalable deployment across fleets and corrosion types, enabling gradual expansion and long-term adaptability. Analytical validation confirms that the Crawler operates reliably under both stationary repair and active locomotion, meeting the strict safety margins required in aviation environments.

Ultimately, the S.P.A.R.K. Crawler represents a shift toward intelligent, automated aircraft surface care, aligning with industry trends in autonomy, safety, and efficiency. Bringing the repair system directly to the aircraft surface reduces costs, risk, and downtime while enhancing maintenance consistency and scalability. Positioned as both a solution to today's corrosion challenges and a foundational platform for future autonomous maintenance, the S.P.A.R.K. Crawler transforms aircraft corrosion management into a safe, efficient, and scalable process that advances the future of intelligent aircraft care.

7. Highlighted Changes since the Proposal Paper

Updates from the proposal and judges' feedback were incorporated across all sections. In the Situation Assessment, we added expanded information on current methods and regulations governing corrosion repair performed by maintenance personnel. This includes a discussion of the risks associated with the repair process, particularly exposure to chemical sealants, as well as the strict regulatory requirements surrounding corrosion treatment.

In System Architecture, we added details about scanning drones currently available in the industry and market. The effectiveness of the S.P.A.R.K. Crawler depends on obtaining precise location data for identified corrosion. In addition to drone capabilities, we included a more detailed explanation of how the drone system, and the S.P.A.R.K. Crawler operate together as an integrated unit.

In the Concept of Operations (ConOps), we expanded on how the crawler monitors repair progress. This includes the use of a downward-facing chassis-mounted camera to visually track repair execution. We also incorporated a time-based monitoring component, where repair progression is evaluated based on the curing time required for sealants and paint. These visual and temporal systems work together to provide comprehensive monitoring of repair status.

In the Path to Deployment section, we elaborated on the potential return on investment (ROI) of the S.P.A.R.K. Crawler, including a graph comparing costs and time savings between traditional maintenance personnel and the automated system. In addition, the deployment timeline was refined to include more specific testing milestones and a clearer system integration strategy.

Section 5.1 retains its core introduction and workflow integration content with minor edits for concision. Key additions include a stakeholder analysis that distinguishes M&E leadership, aircraft maintenance technicians (AMTs), and maintenance planners, each with defined value propositions. We also added a regulatory argument centered on CPCP requirements, emphasizing that corrosion maintenance is non-discretionary under FAA regulations, meaning operators are already incurring these costs regardless of tooling. The regulatory pathway discussion was expanded to explain how the limited scope of approval, using existing approved materials without introducing new repair methods, helps reduce time to deployment for early adopters.

Section 5.2 preserves all original cost figures and ROI projections without modification, while adding supporting citations. Three new subsections were introduced in response to judge feedback: one expanding the $2.2B corrosion cost estimate into a broader adoption rationale including aircraft-on-ground (AOG) costs and CPCP compliance requirements; one addressing safety and workforce impacts using OSHA fall statistics and chemical exposure risks, framed as both humanitarian and financial benefits; and one analyzing the full repair lifecycle, explaining how early autonomous intervention prevents escalation into significantly more expensive structural corrosion repairs. Seven additional citations were incorporated across both sections to address potential reviewer concerns, including OSHA CPL 03-00-025, FAA AC 43-4B, BLS OES 2024, and the Oliver Wyman MRO Forecast 2024.

8. References

[1] Advanex, “Pitting corrosion on aircraft - causes, detection and remediation,” *Katoadvanex.com*, May 06, 2025. https://blog.katoadvanex.com/pitting-corrosion-on-aircraft-causes-detection-and-remediation

[28] T. Seidl and R. Wehner, “Walking on inclines: how do desert ants monitor slope and step length?” *Frontiers in Zoology*, vol. 5, no. 1, p. 8, 2008, doi: https://doi.org/10.1186/1742-9994-5-corr8.

[23] OnFly Air, “OnFly Air,” *OnFly Air*, Mar. 26, 2026. https://onflyair.com/the-hidden-cost-of-aog-downtime-why-every-hour-grounded-costs-more-than-you-think/ (accessed May 03, 2026).

[25] R. Domingo, “Corrosion Control for Aircraft,” Sep. 2018. Available: https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_43-4B.pdf

[27] Thomas Pohl, “Cost per hour of downtime per aircraft is ~ 10,000 USD+ more,” *SAP Community*, May 02, 2013. https://community.sap.com/t5/sap-for-aerospace-and-defense-blogs/cost-per-hour-of-downtime-per-aircraft-is-10-000-usd-more/ba-p/12992392

[16] “Duroair,” *Duroair*, 2025. https://duroair.com/blog/protecting-aircraft-presents-indoor-air-risks/ (accessed May 03, 2026).

[24] “Pitting Corrosion - AMPP,” *www.ampp.org*. https://www.ampp.org/technical-research/impact/corrosion-basics/group-1/pitting-corrosion

[5] Aircraft Owners and Pilots Association, “Aircraft Corrosion,” *Aopa.org*, 2019. https://www.aopa.org/go-fly/aircraft-and-ownership/maintenance-and-inspections/aircraft-corrosion

[17] H. Wilkinson, “How to repair pitting corrosion,” *Belzona Blog*, Jan. 03, 2022. https://blog.belzona.com/why-you-should-be-concerned-about-pitting-corrosion-and-the-proven-solutions-to-combat-it/

[29] U.S. Bureau of Labor Statistics, “Aircraft and Avionics Equipment Mechanics and Technicians : Occupational Outlook Handbook:: U.S. Bureau of Labor Statistics,” *Bls.gov*, Apr. 12, 2019. https://www.bls.gov/ooh/installation-maintenance-and-repair/aircraft-and-avionics-equipment-mechanics-and-technicians.htm

[20] John, “Understanding How Much It Costs to Maintain an Airplane,” *My Aircraft Cost*, Mar. 12, 2025. https://myaircraftcost.com/how-much-does-it-cost-to-maintain-an-airplane/

[11] “Corrosion Prevention - Aircraft MRO | 3M US,” *3m.com*, 2021. https://www.3m.com/3M/en_US/aerospace-us/segment-solutions/aircraft-mro/corrosion-prevention/ (accessed Feb. 1, 2026).

[26] S. Shepard and M. Beemer, “Thermographic Detection of Near and Far-Side Corrosion,” 2018. Accessed: Feb. 1, 2026. [Online]. Available: https://events.esd.org/wp-content/uploads/2018/08/Thermographic-Detection-of-Near-and-Far-Side-Corrosion.pdf

[9]“A11103 Qbbbe3 NBS SPECIAL PUBLICATION 511-2 U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards rnpRn^JiON IN THE UNITED STATES (Appendix B) A Report to NBS by 3ttpMA CotMmhii«; laboratorieŝ,” 1978. Accessed: Feb. 1, 2026. [Online]. Available: https://www.govinfo.gov/content/pkg/GOVPUB-C13-73cebcd7fba7f2a511f7acf566c1ba0d/pdf/GOVPUB-C13-73cebcd7fba7f2a511f7acf566c1ba0d.pdf

[13] “Cost of Corrosion - Aircraft,” *Rust Bullet, LLC.*, Jun. 25, 2024. https://www.rustbullet.com/cost-of-corrosion/advantage-aircraft/

[15] D. Parker, “OSHA Instruction,” *OSHA Instruction*, Apr. 2023, Accessed: Apr. 03, 2026. [Online]. Available: https://www.osha.gov/sites/default/files/enforcement/directives/CPL_03-00-025.pdf

[6]“Aircraft Paint Ultimate Guide: Powerful Protection, Cost & Repaint Planning - Fly with Hop On Air,” *Fly with Hop On Air*, Feb. 14, 2026. https://www.flyhopon.com/aircraft-paint/ (accessed Apr 03, 2026).

[2]“Aircraft Mechanics and Service Technicians,” *Bureau of Labor Statistics*, Apr. 25, 2023. https://www.bls.gov/oes/2023/may/oes493011.htm

[3]“Aircraft Maintenance Safety | OSHA Safety Manuals,” *OSM*, Jan. 07, 2017. https://www.safetymanualosha.com/aircraft-maintenance-safety/

[4]“Aircraft Operating Costs 4-1,” 2023. Available: https://www.faa.gov/regulations_policies/policy_guidance/benefit_cost/econ-value-section-4-op-costs.pd

[7] A. Koplin, “Industry first: FAA accepts Delta’s plan to use drones for maintenance inspections,” *Delta News Hub*, Oct. 04, 2024. https://news.delta.com/industry-first-faa-accepts-deltas-plan-use-drones-maintenance-inspections

[8]A. Guide, “Aircraft Corrosion Removal and Surface Treatment Guide,” *Aero Guide*, 2026. https://www.aircraftsystemstech.com/2019/09/aircraft-corrosion-removal.html?m=1 (accessed Apr 03, 2026).

[10] B. Brandoli *et al.*, “Checking your browser - reCAPTCHA,” *Nih.gov*, 2024. https://pmc.ncbi.nlm.nih.gov/articles/PMC8230709/ (accessed May 03, 2026).

[12] “Corrosion Prevention and Control Program,” *Federal Register*, Oct. 03, 2002. https://www.federalregister.gov/documents/2002/10/03/02-24932/corrosion-prevention-and-control-program

[14] Didier Aldana Rodríguez, Cristian Lozano Tafur, P. Fernando, J. Armando, and J. Carlos, “Inspection of aircraft and airports using UAS: a review,” *Results in engineering*, vol. 22, pp. 102330–102330, Jun. 2024, doi: https://doi.org/10.1016/j.rineng.2024.102330.

[18] I. Perez and P. Kulowitch, “THERMOGRAPHY FOR CHARACTERIZATION OF CORROSION DAMAGE,” Feb. 1999. Accessed: Feb. 1, 2026. [Online]. Available: https://apps.dtic.mil/sti/tr/pdf/ADA378009.pdf

[19] Jack Edwards, “How Delta TechOps Pioneered FAA-Approved Drone Inspections,” OXMaint, Mar. 21, 2026. https://oxmaint.com/industries/aviation-management/delta-techops-faa-approved-drone-inspections-case-study

[21] Jordan, “The Armoloy Corporation,” *The Armoloy Corporation*, May 22, 2025. https://armoloy.com/pitting-corrosion-causes-effects-solutions/

[22] L. Li, M. Chakik, and R. Prakash, “A Review of Corrosion in Aircraft Structures and Graphene-Based Sensors for Advanced Corrosion Monitoring,” *Sensors*, vol. 21, no. 9, p. 2908, Apr. 2021, doi: https://doi.org/10.3390/s21092908.

[30] Virmani, Paul. *CORROSION COSTS and PREVENTIVE STRATEGIES in the UNITED STATES OBJECTIVES and SCOPE*. U.S. Department of Transportation, 1 Mar. 2002.

[31] Y. Zhou *et al.*, “A two-degree-of-freedom miniature ultrasonic motor driven by two modes coupled by three kinds of vibrations,” *Ultrasonics*, vol. 161, no. 107930, 2026.

9. Appendices

9.1 Suction Cup Calculations

Assumptions:

- Crawler Mass (Fully loaded up) ≈ 35 kg
- $FOS_{min} = 3$
- Suction Cup Diameter = 0.1524 m (6 in)
- Suction efficiency = 50%
- Active Vacuum = -60 kPa

Using 50% suction efficiency for a corrosion repair crawler is a conservative assumption, given that the system operates in a harsh, dynamic environment. The grinding process introduces continuous vibration, torque reactions, and shock loads that can momentarily reduce holding force. At the same time, the surface condition is actively changing as paint and corrosion are removed, which can degrade seal quality. Grinding also generates dust and debris that may contaminate the suction seal and cause micro-leaks. Given these dynamic loads, surface variability, and the high consequences of detachment, designing around 50% efficiency provides a reasonable safety margin for reliable operation.

This is for the worst-case scenario, walking across the underbelly of an aircraft.

$$W = 35\ kg \times 9.81\ \frac{m}{s^2} = 343.35\ N$$

$$F_{required} = W\ \times FOS_{min} = 1030.05\ N$$

$$A = \pi\left(\frac{0.1524\ m}{2}\right)^2 = 0.0182\ m^2$$

$$F_{cup} = eff\ \times Active\ Vacuum\ \times A = 547.244\ N$$

$$F_{cup}\ \times 6 = 3283.46\ N = F_{Allcups}$$

$$FOS_{eff} = {}^{F_{Allcups}}/_{W} = 9.56$$

Unloading 3 cups to walk across the underbelly of the aircraft

$$F_{walking} = F_{cup}\ \times 3 = 1641.732\ N$$

$$FOS_{walking} = {}^{F_{walking}}/_{W} = 4.781$$

If one of the three cups fails while walking, and only two are holding on at a time per side

$$F_{cupfailure} = F_{cup}\ \times 2 = 1098.488\ N$$

$$FOS_{cupfailure} = {}^{F_{cupfailure}}/_{W} = 3.187$$

For critical failure if all but one leg fails

$$F_{CupCritical} = F_{cup} = 343.35\ N$$

$$FOS_{Critical} = {}^{F_{cup}}/_{W} = 1.59$$

9.2 Cost Analysis Extra Data

***Table 3**: Assumptions for Cost Analysis*

S.P.A.R.K. Crawler — Model Assumptions & Inputs		
Parameter	**Value**	**Notes / Source**
Labor		
Labor rate — low ($/hr)	**65.00**	*Source: Industry avg, FAA MRO survey 2023*
Labor rate — high ($/hr)	**85.00**	*Source: Industry avg, FAA MRO survey 2023*
Labor rate — avg ($/hr)	$75.00	*Midpoint used in all calculations*
Maintenance Cycles		
Corrosion maintenance cycles / yr / aircraft	**5.00**	*Typical commercial operator; adjust as needed*
Unit Costs		
Gen-1 unit cost — low ($)	**$9,000.00**	*Production-scale estimate*
Gen-1 unit cost — high ($)	**$10,000.00**	*Production-scale estimate*
Gen-2 unit cost — low ($)	**$11,500.00**	*Expanded sensing + tooling*
Gen-2 unit cost — high ($)	**$13,000.00**	*Expanded sensing + tooling*
Operating Costs		
Annual operating cost — low ($/unit/yr)	**$1,600.00**	*Consumables, servicing, parts*
Annual operating cost — high ($/unit/yr)	**$2,000.00**	*Consumables, servicing, parts*
Annual operating cost — avg ($/unit/yr)	$800.00	*Used in lifecycle tables*
System Oversight		
Technician oversight hours / yr / aircraft	**24.00**	*Setup + supervisory time (not counted as savings)*
Downtime Value		
Aircraft downtime cost ($/hr)	**$10,000.00**	*Source: Commercial aviation ops, ref [11]*
Downtime hours saved / maintenance cycle	**2.00**	*Conservative estimate vs. manual AOG extension*

***Table 4**: Different Deployment Scenarios*

Multi-Crawler & Fleet Deployment Scenarios

Single Crawler (1 per aircraft)						
Metric	**Units**	**Year 1**	**Year 3**	**Year 5**	**Year 7**	**Year 10**
Gross savings	$	$23,250	$69,750	$116,250	$162,750	$232,500
Operating costs	$	$2,000	$6,000	$10,000	$14,000	$20,000
Net savings	$	**$21,250**	**$63,750**	**$106,250**	**$148,750**	**$212,500**
Hours returned to tech	h	310 h	930 h	1,550 h	2,170 h	3,100 h

3-Crawler Swarm (3 per aircraft)						
Metric	**Units**	**Year 1**	**Year 3**	**Year 5**	**Year 7**	**Year 10**
Gross savings	$	$69,750	$209,250	$348,750	$488,250	$697,500
Operating costs	$	$6,000	$18,000	$30,000	$42,000	$60,000
Net savings	$	**$63,750**	**$191,250**	**$318,750**	**$446,250**	**$637,500**
Hours returned to tech	h	930 h	2,790 h	4,650 h	6,510 h	9,300 h

Fleet Deployment (10 aircraft's, one crawler each)						
Metric	**Units**	**Year 1**	**Year 3**	**Year 5**	**Year 7**	**Year 10**
Gross savings	$	$232,500	$697,500	$1,162,500	$1,627,500	$2,325,000
Operating costs	$	$20,000	$60,000	$100,000	$140,000	$200,000
Net savings	$	**$212,500**	**$637,500**	**$1,062,500**	**$1,487,500**	**$2,125,000**
Hours returned to tech	h	3,100 h	9,300 h	15,500 h	21,700 h	31,000 h

***Table 5**: Unit Cost & Lifecycle Ownership Cost*

Unit Cost & Lifecycle Ownership Cost			
Item	**Gen-1 (Pitting)**	**Gen-2 (Filiform)**	**Notes**
Unit cost — low	-	$10,000	*Production-scale COTS*
Unit cost — high	$9,000	$11,500	*Production-scale COTS*
Annual operating cost	$2,000	$2,000	*Consumables + servicing*
5-yr total cost of ownership	$10,000	$20,000	*Low unit + 5yr ops*
10-yr total cost of ownership	$20,000	$30,000	*Low unit + 10yr ops*
Projected lifespan	5–10 years	5–10 years	*Routine deployment*
30–40% cost reduction vs prototype	Yes	Yes	*Bulk procurement + reuse*

9.3 MATLAB Codes

```
%% ======================================================
%  NASA GBS Crawler – Forward Kinematics
%  Single leg with 3 DOF:
%    Joint 1: Hip Rotation  (vertical Z axis)
%    Joint 2: Hip Lift      (horizontal X axis)
%    Joint 3: Knee          (horizontal X axis)
%
%  Leg geometry:
%    L1 = upper leg length (hip to knee)   = 9 in = 0.2286 m
%    L2 = lower leg length (knee to foot)  = 9 in = 0.2286 m
%
%  Coordinate frame (body-fixed):
%    X = forward
%    Y = left (leg extends in +Y at rest)
%    Z = up
% ======================================================

clc; clear; close all;

%% ── LEG GEOMETRY ──────────────────────────────────────────────────
L1 = 9 * 0.0254;   % upper leg: 9 inches → meters (0.2286 m)
L2 = 9 * 0.0254;   % lower leg: 9 inches → meters (0.2286 m)

%% ── JOINT ANGLE LIMITS (degrees, then converted) ──────────────
% 270° servos centered at 135°, so range is 0° to 270°
% Define meaningful mechanical limits for each joint:
theta1_lim = [-60,  60];   % Hip rotation:  ±60° from center
theta2_lim = [-45,  90];   % Hip lift:      -45° (down) to +90° (up)
theta3_lim = [  0, 150];   % Knee:           0° (straight) to 150° (bent)

%% ── DH PARAMETER TABLE ────────────────────────────────────────────
% Columns: [a (m), alpha (rad), d (m)]
% theta is the variable for each joint — passed in separately
DH = [
  0,    pi/2,  0;   % Joint 1: Hip Rotation
  L1,   0,     0;   % Joint 2: Hip Lift
  L2,   0,     0;   % Joint 3: Knee
];

%% ── DH TRANSFORM FUNCTION ─────────────────────────────────────────
% Returns 4x4 homogeneous transform for one DH joint
dh_transform = @(a, alpha, d, theta) [
  cos(theta),  -sin(theta)*cos(alpha),   sin(theta)*sin(alpha),  a*cos(theta);
  sin(theta),   cos(theta)*cos(alpha),  -cos(theta)*sin(alpha),  a*sin(theta);
  0,            sin(alpha),              cos(alpha),             d;
  0,            0,                       0,                      1
];

%% ── FORWARD KINEMATICS FUNCTION ──────────────────────────────────
```

```
% Inputs:  theta = [theta1, theta2, theta3] in radians
% Outputs: foot position [x; y; z] in meters (local hip frame)
%          T_total = full 4x4 transform matrix
function [foot_pos, T_total] = forwardKinematics(theta, DH, dh_transform)
  T1 = dh_transform(DH(1,1), DH(1,2), DH(1,3), theta(1));
  T2 = dh_transform(DH(2,1), DH(2,2), DH(2,3), theta(2));
  T3 = dh_transform(DH(3,1), DH(3,2), DH(3,3), theta(3));
  T_total = T1 * T2 * T3;
  foot_pos = T_total(1:3, 4);  % extract x, y, z from last column
end

%% ── EXAMPLE: TEST WITH SPECIFIC JOINT ANGLES ──────────────────
fprintf('=== Forward Kinematics Test ===\n\n');

% Test Case 1: All joints at center (135° → 0° offset from center)
theta_center = deg2rad([0, 0, 0]);
[pos1, T1] = forwardKinematics(theta_center, DH, dh_transform);
fprintf('Test 1 — All joints centered (0, 0, 0 deg):\n');
fprintf('  Foot position: x=%.4f m, y=%.4f m, z=%.4f m\n\n', pos1(1), pos1(2), pos1(3));

% Test Case 2: Leg extended straight out (crab stance)
theta_stance = deg2rad([0, -30, 60]);
[pos2, T2] = forwardKinematics(theta_stance, DH, dh_transform);
fprintf('Test 2 — Typical stance (0, -30, 60 deg):\n');
fprintf('  Foot position: x=%.4f m, y=%.4f m, z=%.4f m\n\n', pos2(1), pos2(2), pos2(3));

% Test Case 3: Leg lifted for swing
theta_swing = deg2rad([30, 45, 30]);
[pos3, T3] = forwardKinematics(theta_swing, DH, dh_transform);
fprintf('Test 3 — Swing phase lifted (30, 45, 30 deg):\n');
fprintf('  Foot position: x=%.4f m, y=%.4f m, z=%.4f m\n\n', pos3(1), pos3(2), pos3(3));

%% ── WORKSPACE VISUALIZATION ──────────────────────────────────────
% Sweep all joint angles through their ranges and plot every
% reachable foot position — this gives you the workspace bubble

fprintf('Generating workspace plot...\n');

theta1_range = deg2rad(linspace(theta1_lim(1), theta1_lim(2), 30));
theta2_range = deg2rad(linspace(theta2_lim(1), theta2_lim(2), 30));
theta3_range = deg2rad(linspace(theta3_lim(1), theta3_lim(2), 30));

% Preallocate
n_pts = length(theta1_range) * length(theta2_range) * length(theta3_range);
workspace = zeros(n_pts, 3);
idx = 1;

for i = 1:length(theta1_range)
  for j = 1:length(theta2_range)
    for k = 1:length(theta3_range)
      theta = [theta1_range(i), theta2_range(j), theta3_range(k)];
```

```
      [pos, ~] = forwardKinematics(theta, DH, dh_transform);
      workspace(idx, :) = pos';
      idx = idx + 1;
    end
  end
end

% 3D workspace plot
figure('Name', 'Leg Workspace', 'NumberTitle', 'off');
scatter3(workspace(:,1), workspace(:,2), workspace(:,3), ...
        1, workspace(:,3), 'filled');
colorbar;
xlabel('X (m) — Forward');
ylabel('Y (m) — Lateral');
zlabel('Z (m) — Up');
title('Crawler Leg Reachable Workspace');
axis equal; grid on;
colormap jet;

% Mark the test positions
hold on;
plot3(pos1(1), pos1(2), pos1(3), 'ko', 'MarkerSize', 10, 'LineWidth', 2);
plot3(pos2(1), pos2(2), pos2(3), 'gs', 'MarkerSize', 10, 'LineWidth', 2);
plot3(pos3(1), pos3(2), pos3(3), 'r^', 'MarkerSize', 10, 'LineWidth', 2);
legend('Workspace', 'Centered', 'Stance', 'Swing', 'Location', 'best');
hold off;

%% ── GAIT PATH VISUALIZATION ──────────────────────────────────────
% Plot what the foot traces during one step cycle
% Simple trapezoidal foot path: forward on ground, arc in air

fprintf('Generating gait path plot...\n');

% Stance phase: foot moves backward on ground (body moves forward)
t_stance = linspace(0, 1, 50);
theta1_s = deg2rad(linspace(-30, 30, 50));   % hip sweeps back
theta2_s = deg2rad(linspace(-30, -30, 50));  % hip lift constant
theta3_s = deg2rad(linspace(60, 60, 50));    % knee constant

% Swing phase: foot lifts and swings forward
t_swing  = linspace(0, 1, 50);
theta1_sw = deg2rad(linspace(30, -30, 50));  % hip swings forward
theta2_sw = deg2rad(linspace(-30, 30, 50));  % hip lifts up then down
theta3_sw = deg2rad(linspace(60, 30, 50));   % knee bends then extends

stance_path = zeros(50, 3);
swing_path  = zeros(50, 3);

for i = 1:50
  [p_s,  ~] = forwardKinematics([theta1_s(i),  theta2_s(i),  theta3_s(i)],  DH, dh_transform);
  [p_sw, ~] = forwardKinematics([theta1_sw(i), theta2_sw(i), theta3_sw(i)], DH, dh_transform);
```

```
  stance_path(i,:) = p_s';
  swing_path(i,:)  = p_sw';
end

full_path = [stance_path; swing_path];

figure('Name', 'Foot Path During Gait', 'NumberTitle', 'off');
subplot(2,1,1);
plot3(stance_path(:,1), stance_path(:,2), stance_path(:,3), ...
     'b-', 'LineWidth', 2); hold on;
plot3(swing_path(:,1),  swing_path(:,2),  swing_path(:,3), ...
     'r-', 'LineWidth', 2);
plot3(full_path(1,1),   full_path(1,2),   full_path(1,3), ...
     'go', 'MarkerSize', 10, 'LineWidth', 2);
xlabel('X (m)'); ylabel('Y (m)'); zlabel('Z (m)');
title('3D Foot Path — One Full Step Cycle');
legend('Stance (on ground)', 'Swing (in air)', 'Start', 'Location', 'best');
grid on; axis equal;

subplot(2,1,2);
plot(full_path(:,1), full_path(:,3), 'k-', 'LineWidth', 2);
xlabel('X (m) — Forward/Back');
ylabel('Z (m) — Up/Down');
title('Side View of Foot Path');
grid on; axis equal;

%% ============================================================
%  NASA GBS Crawler – Static Stability Analysis
%  Alternating tripod gait hexapod
%
%  Checks:
%    1. Support polygon at every gait phase
%    2. Stability margin throughout full walk cycle
%    3. Tip-over angle
%    4. Maximum stable slope angle
%
%  Coordinate frame (body-fixed, origin at body center):
%    X = forward, Y = left, Z = up
% ============================================================

clc; clear; close all;

%% ── ROBOT BODY GEOMETRY ──────────────────────────────────────
% *** UPDATE THESE TO MATCH YOUR ACTUAL CHASSIS DIMENSIONS ***
L_body = 0.50;    % body length (m) — front to rear hip spacing
W_body = 0.30;    % body width  (m) — left to right hip spacing
h_CoM  = 0.15;    % CoM height above ground (m) — estimate, update later

% Leg geometry
L1 = 9 * 0.0254;  % 0.2286 m
L2 = 9 * 0.0254;  % 0.2286 m
```

```
% Assume CoM is at the geometric center of the body
CoM_body = [0; 0];   % (x, y) in body frame — update if offset

%% ── HIP ATTACHMENT POSITIONS (body frame, x-y only) ──────
% Legs numbered 1-6:
%   1=Front Left, 2=Mid Left,  3=Rear Left
%   4=Front Right,5=Mid Right, 6=Rear Right
hip_pos = [
   L_body/2,  W_body/2;   % Leg 1: Front Left
   0,         W_body/2;   % Leg 2: Mid Left
  -L_body/2,  W_body/2;   % Leg 3: Rear Left
   L_body/2, -W_body/2;   % Leg 4: Front Right
   0,        -W_body/2;   % Leg 5: Mid Right
  -L_body/2, -W_body/2;   % Leg 6: Rear Right
];

%% ── ALTERNATING TRIPOD GAIT DEFINITION ──────────────────────
% Tripod A (stance in phase 1): Legs 1, 5, 3 (FL, MR, RL)
% Tripod B (stance in phase 2): Legs 4, 2, 6 (FR, ML, RR)
tripod_A = [1, 5, 3];
tripod_B = [4, 2, 6];

%% ── FOOT POSITIONS (from IK/FK — simplified for stability) ─
% For stability analysis we define nominal foot positions
% as hip position + step offset in x, fixed reach in y, fixed z
% These should come from your IK solution in the full controller

step_reach_y = 0.35;   % how far out the foot is laterally (m)
foot_z       = -0.10;  % foot height below hip (ground level) (m)

% Compute nominal foot positions for each leg (x, y on ground)
% Foot x position varies during gait — use hip x as neutral
foot_pos_nominal = zeros(6, 2);
for i = 1:6
  foot_pos_nominal(i, 1) = hip_pos(i, 1);                    % x = hip x
  foot_pos_nominal(i, 2) = sign(hip_pos(i,2)) * step_reach_y;     % y = ±reach
end

%% ── HELPER: POINT IN TRIANGLE CHECK ─────────────────────────
function inside = pointInTriangle(P, A, B, C)
  % Returns true if point P is inside triangle ABC
  % Uses cross product sign method
  d1 = sign((P(1)-B(1))*(A(2)-B(2)) - (A(1)-B(1))*(P(2)-B(2)));
  d2 = sign((P(1)-C(1))*(B(2)-C(2)) - (B(1)-C(1))*(P(2)-C(2)));
  d3 = sign((P(1)-A(1))*(C(2)-A(2)) - (C(1)-A(1))*(P(2)-A(2)));
  has_neg = (d1 < 0) || (d2 < 0) || (d3 < 0);
  has_pos = (d1 > 0) || (d2 > 0) || (d3 > 0);
  inside  = ~(has_neg && has_pos);
end
```

```
%% ── HELPER: DISTANCE FROM POINT TO LINE SEGMENT ──────────────
function d = pointToSegmentDist(P, A, B)
  AB = B - A;
  AP = P - A;
  t  = dot(AP, AB) / dot(AB, AB);
  t  = max(0, min(1, t));
  closest = A + t * AB;
  d = norm(P - closest);
end

%% ── HELPER: STABILITY MARGIN ─────────────────────────────────────
function SM = stabilityMargin(CoM_xy, feet_xy)
  % feet_xy = 3x2 matrix of the three stance foot positions
  A = feet_xy(1,:);
  B = feet_xy(2,:);
  C = feet_xy(3,:);
  P = CoM_xy';

  d1 = pointToSegmentDist(P, A, B);
  d2 = pointToSegmentDist(P, B, C);
  d3 = pointToSegmentDist(P, C, A);
  SM = min([d1, d2, d3]);

  % If CoM is outside the triangle, make SM negative
  if ~pointInTriangle(P, A, B, C)
    SM = -SM;
  end
end

%% ── STATIC STABILITY AT NOMINAL STANCE ──────────────────────────
fprintf('=== Static Stability Analysis ===\n\n');

% Phase 1: Tripod A on ground (legs 1, 5, 3)
feet_A = foot_pos_nominal(tripod_A, :);
SM_A   = stabilityMargin(CoM_body, feet_A);

fprintf('Phase 1 — Tripod A stance (Legs 1, 5, 3):\n');
fprintf('  Foot positions:\n');
for i = 1:3
  fprintf('    Leg %d: (%.3f, %.3f) m\n', tripod_A(i), feet_A(i,1), feet_A(i,2));
end
fprintf('  CoM projection: (%.3f, %.3f) m\n', CoM_body(1), CoM_body(2));
fprintf('  Stability Margin: %.1f mm ', SM_A*1000);
if SM_A > 0.020
  fprintf('✓ STABLE (margin > 20mm)\n\n');
elseif SM_A > 0
  fprintf('⚠ MARGINALLY STABLE (margin < 20mm)\n\n');
else
  fprintf('✗ UNSTABLE — CoM outside support polygon!\n\n');
end
```

```
% Phase 2: Tripod B on ground (legs 4, 2, 6)
feet_B = foot_pos_nominal(tripod_B, :);
SM_B   = stabilityMargin(CoM_body, feet_B);

fprintf('Phase 2 — Tripod B stance (Legs 4, 2, 6):\n');
fprintf('  Foot positions:\n');
for i = 1:3
  fprintf('    Leg %d: (%.3f, %.3f) m\n', tripod_B(i), feet_B(i,1), feet_B(i,2));
end
fprintf('  CoM projection: (%.3f, %.3f) m\n', CoM_body(1), CoM_body(2));
fprintf('  Stability Margin: %.1f mm ', SM_B*1000);
if SM_B > 0.020
  fprintf('✓ STABLE (margin > 20mm)\n\n');
elseif SM_B > 0
  fprintf('⚠ MARGINALLY STABLE (margin < 20mm)\n\n');
else
  fprintf('✗ UNSTABLE — CoM outside support polygon!\n\n');
end

%% ── TIP-OVER ANGLE ──────────────────────────────────────────────────
SM_min = min(SM_A, SM_B);
phi_tip = rad2deg(atan2(SM_min, h_CoM));

fprintf('Tip-over Analysis:\n');
fprintf('  Minimum stability margin: %.1f mm\n', SM_min*1000);
fprintf('  CoM height:              %.0f mm\n', h_CoM*1000);
fprintf('  Tip-over angle:          %.1f deg\n\n', phi_tip);

%% ── SLOPE STABILITY ─────────────────────────────────────────────────
% Maximum slope the robot can stand on without tipping
% Calculated for forward/backward and lateral directions

% Forward/backward: CoM to front and rear edges
d_fwd  = L_body/2;   % approx distance to front support edge
d_rear = L_body/2;   % approx distance to rear support edge
d_lat  = step_reach_y - W_body/2;  % distance to lateral edge

slope_fwd  = rad2deg(atan2(d_fwd,  h_CoM));
slope_rear = rad2deg(atan2(d_rear, h_CoM));
slope_lat  = rad2deg(atan2(d_lat,  h_CoM));

fprintf('Slope Stability:\n');
fprintf('  Max forward slope:  %.1f deg\n', slope_fwd);
fprintf('  Max rearward slope: %.1f deg\n', slope_rear);
fprintf('  Max lateral slope:  %.1f deg\n\n', slope_lat);

%% ── STABILITY ACROSS FULL GAIT CYCLE ────────────────────────────────
% Simulate CoM moving slightly as gait progresses
% In a real system CoM shifts as legs swing — model a ±20mm shift
```

```
fprintf('Simulating stability margin across gait cycle...\n\n');

n_steps  = 200;
CoM_x_sweep = linspace(-0.05, 0.05, n_steps);  % CoM shifts ±50mm fore/aft
SM_cycle = zeros(n_steps, 2);

for i = 1:n_steps
  CoM_test = [CoM_x_sweep(i); 0];
  SM_cycle(i, 1) = stabilityMargin(CoM_test, feet_A);
  SM_cycle(i, 2) = stabilityMargin(CoM_test, feet_B);
end

min_SM_cycle = min(SM_cycle(:));
fprintf('  Minimum stability margin across gait cycle: %.1f mm ', min_SM_cycle*1000);
if min_SM_cycle > 0.020
  fprintf('✓ STABLE throughout\n\n');
elseif min_SM_cycle > 0
  fprintf('⚠ MARGINAL at some points\n\n');
else
  fprintf('✗ UNSTABLE at some gait positions!\n\n');
end

%% ── PLOTS ──────────────────────────────────────────────────────────────

% Figure 1: Support polygons
figure('Name', 'Support Polygons', 'NumberTitle', 'off');

subplot(1,2,1);
patch(feet_A([1,2,3,1],1), feet_A([1,2,3,1],2), ...
     'cyan', 'FaceAlpha', 0.3, 'EdgeColor', 'b', 'LineWidth', 2);
hold on;
scatter(feet_A(:,1), feet_A(:,2), 100, 'b', 'filled');
for i = 1:3
  text(feet_A(i,1)+0.01, feet_A(i,2)+0.01, sprintf('Leg %d', tripod_A(i)), 'FontSize', 10);
end
plot(CoM_body(1), CoM_body(2), 'r+', 'MarkerSize', 15, 'LineWidth', 3);
scatter(hip_pos(:,1), hip_pos(:,2), 50, 'k', 's');
xlabel('X (m) — Forward'); ylabel('Y (m) — Left');
title(sprintf('Tripod A Stance\nSM = %.1f mm', SM_A*1000));
legend('Support polygon', 'Stance feet', 'CoM', 'Hips', 'Location', 'best');
axis equal; grid on;
xlim([-0.40, 0.40]); ylim([-0.50, 0.50]);

subplot(1,2,2);
patch(feet_B([1,2,3,1],1), feet_B([1,2,3,1],2), ...
     'yellow', 'FaceAlpha', 0.3, 'EdgeColor', [0.8 0.6 0], 'LineWidth', 2);
hold on;
scatter(feet_B(:,1), feet_B(:,2), 100, [0.8 0.6 0], 'filled');
for i = 1:3
```

```
  text(feet_B(i,1)+0.01, feet_B(i,2)+0.01, sprintf('Leg %d', tripod_B(i)), 'FontSize', 10);
end
plot(CoM_body(1), CoM_body(2), 'r+', 'MarkerSize', 15, 'LineWidth', 3);
scatter(hip_pos(:,1), hip_pos(:,2), 50, 'k', 's');
xlabel('X (m) — Forward'); ylabel('Y (m) — Left');
title(sprintf('Tripod B Stance\nSM = %.1f mm', SM_B*1000));
legend('Support polygon', 'Stance feet', 'CoM', 'Hips', 'Location', 'best');
axis equal; grid on;
xlim([-0.40, 0.40]); ylim([-0.50, 0.50]);

sgtitle('Support Polygons — Alternating Tripod Gait');

% Figure 2: Stability margin across gait cycle
figure('Name', 'Stability Margin vs Gait', 'NumberTitle', 'off');
plot(CoM_x_sweep*1000, SM_cycle(:,1)*1000, 'b-', 'LineWidth', 2); hold on;
plot(CoM_x_sweep*1000, SM_cycle(:,2)*1000, 'r-', 'LineWidth', 2);
yline(20, 'k--', 'Min target (20mm)', 'LineWidth', 1.5);
yline(0,  'k-',  'Tip-over threshold', 'LineWidth', 1.5);
fill([CoM_x_sweep(1)*1000, CoM_x_sweep(end)*1000, ...
      CoM_x_sweep(end)*1000, CoM_x_sweep(1)*1000], ...
     [0, 0, -50, -50], 'r', 'FaceAlpha', 0.1, 'EdgeColor', 'none');
xlabel('CoM X Position (mm) — Forward/Back shift');
ylabel('Stability Margin (mm)');
title('Stability Margin vs CoM Position During Gait');
legend('Tripod A stance', 'Tripod B stance', 'Location', 'best');
grid on;

% Figure 3: Slope stability summary
figure('Name', 'Slope Stability', 'NumberTitle', 'off');
slopes     = [slope_fwd, slope_rear, slope_lat];
slope_lbls = {'Forward', 'Rearward', 'Lateral'};
bar(slopes, 'FaceColor', [0.2 0.5 0.8]);
hold on;
yline(15, 'r--', 'Typical terrain target (15 deg)', 'LineWidth', 2);
set(gca, 'XTickLabel', slope_lbls);
ylabel('Maximum Stable Slope (degrees)');
title('Slope Stability by Direction');
grid on;
%% ============================================================
%  NASA GBS Crawler – Dynamic Analysis
%
%  Computes:
%    1. Joint torques during motion (M*q" + C*q' + G = tau)
%    2. Ground reaction forces during stance
%    3. Impact force at foot touchdown
%    4. Required motor torque vs speed across gait cycle
%    5. Power consumption per joint
%
%  Leg geometry:
%    L1 = L2 = 9 in = 0.2286 m
%  Robot mass: 35 kg
```

```
% ========================================================

clc; clear; close all;

%% ── ROBOT PARAMETERS ──────────────────────────────────────────────
m_total   = 35.0;        % total robot mass (kg)
g         = 9.81;        % gravity (m/s²)
n_stance  = 3;           % stance legs at any time (tripod gait)
FOS       = 2.0;         % factor of safety

% Leg geometry
L1 = 9 * 0.0254;         % upper leg length (m)
L2 = 9 * 0.0254;         % lower leg length (m)

% Leg mass estimates — update when you weigh components
% Assume leg mass is ~4% of total per leg (6 legs = 24%, body = 76%)
m_leg    = m_total * 0.04;   % ~1.4 kg per leg
m_upper  = m_leg * 0.50;     % upper segment mass (kg)
m_lower  = m_leg * 0.35;     % lower segment mass (kg)
m_foot   = m_leg * 0.15;     % foot + servo mass (kg)

% Segment inertia (treating each as a uniform rod: I = 1/12 * m * L^2)
I_upper = (1/12) * m_upper * L1^2;
I_lower = (1/12) * m_lower * L2^2;

% Gait parameters
v_walk      = 0.10;        % target walking speed (m/s)
step_length = 0.24;        % step length (m) — from IK analysis
step_time   = step_length / v_walk;   % time per step (s)
swing_frac  = 0.50;        % fraction of step time in swing
swing_time  = step_time * swing_frac;
stance_time = step_time * (1 - swing_frac);

fprintf('=== Dynamic Analysis ===\n\n');
fprintf('Gait Parameters:\n');
fprintf('  Walking speed:  %.2f m/s\n', v_walk);
fprintf('  Step length:    %.0f mm\n', step_length*1000);
fprintf('  Step time:      %.2f s\n', step_time);
fprintf('  Swing time:     %.2f s\n', swing_time);
fprintf('  Stance time:    %.2f s\n\n', stance_time);

%% ── JOINT VELOCITY AND ACCELERATION ──────────────────────────────
% Estimate peak joint velocities during swing phase
% Hip rotation sweeps ~60 deg in swing_time
% Hip lift goes up and back down in swing_time
% Knee bends and extends in swing_time

theta1_swing = deg2rad(60);   % total hip rotation sweep (rad)
theta2_swing = deg2rad(75);   % total hip lift sweep (rad)
theta3_swing = deg2rad(90);   % total knee sweep (rad)
```

```
% Using trapezoidal velocity profile: peak velocity = 2 * total_angle / time
% (accelerate for half, decelerate for half)
omega1_peak = 2 * theta1_swing / swing_time;   % rad/s
omega2_peak = 2 * theta2_swing / swing_time;
omega3_peak = 2 * theta3_swing / swing_time;

% Peak angular acceleration (ramp up in quarter of swing time)
alpha1_peak = omega1_peak / (swing_time / 4);
alpha2_peak = omega2_peak / (swing_time / 4);
alpha3_peak = omega3_peak / (swing_time / 4);

fprintf('Joint Velocities & Accelerations (swing phase):\n');
fprintf('  Hip Rotation: omega=%.2f rad/s, alpha=%.2f rad/s²\n', omega1_peak, alpha1_peak);
fprintf('  Hip Lift:     omega=%.2f rad/s, alpha=%.2f rad/s²\n', omega2_peak, alpha2_peak);
fprintf('  Knee:         omega=%.2f rad/s, alpha=%.2f rad/s²\n\n', omega3_peak, alpha3_peak);

%% ── EQUATIONS OF MOTION: M*q'' + C*q' + G = tau ──────────────
% Simplified 2-link planar model for the hip lift + knee
% (hip rotation handled separately as it's decoupled in swing)
%
% For a 2-link arm with links (m1,L1) and (m2,L2):
%
% M matrix elements:
%   M11 = I_upper + I_lower + m_lower*L1^2 + (m_upper+m_lower)*L1^2/4
%   M12 = I_lower + m_lower*L1*L2/2*cos(theta3)
%   M22 = I_lower + m_lower*(L2/2)^2
%
% G vector (gravity torques):
%   G1 = (m_upper*L1/2 + m_lower*L1)*g*cos(theta2) +
%        m_lower*L2/2*g*cos(theta2+theta3)
%   G2 =  m_lower*L2/2*g*cos(theta2+theta3)

% Evaluate at worst-case stance position (leg extended, max load)
theta2_stance = deg2rad(-30);   % hip slightly depressed
theta3_stance = deg2rad(60);    % knee bent at 60 deg

% Mass matrix M
M11 = I_upper + I_lower + m_lower*L1^2 + (m_upper+m_lower)*(L1/2)^2;
M12 = I_lower + m_lower*(L1)*(L2/2)*cos(theta3_stance);
M22 = I_lower + m_lower*(L2/2)^2;
M_mat = [M11, M12; M12, M22];

% Coriolis matrix C (at peak swing velocity)
C11 = -m_lower*L1*(L2/2)*sin(theta3_stance)*omega3_peak;
C12 = -m_lower*L1*(L2/2)*sin(theta3_stance)*(omega2_peak + omega3_peak);
C21 =  m_lower*L1*(L2/2)*sin(theta3_stance)*omega2_peak;
C22 =  0;
C_mat = [C11, C12; C21, C22];

% Gravity vector G
G1 = (m_upper*(L1/2) + m_lower*L1)*g*cos(theta2_stance) + ...
```

```
    m_lower*(L2/2)*g*cos(theta2_stance + theta3_stance);
G2 = m_lower*(L2/2)*g*cos(theta2_stance + theta3_stance);
G_vec = [G1; G2];

% Joint accelerations during swing
q_ddot = [alpha2_peak; alpha3_peak];
q_dot  = [omega2_peak; omega3_peak];

% Required joint torques: tau = M*q" + C*q' + G
tau_dynamic = M_mat * q_ddot + C_mat * q_dot + G_vec;

fprintf('Equations of Motion — Required Joint Torques:\n');
fprintf('  (at peak swing acceleration, worst-case stance geometry)\n');
fprintf('  M matrix:\n    [%.4f  %.4f]\n    [%.4f  %.4f]\n\n', ...
      M_mat(1,1), M_mat(1,2), M_mat(2,1), M_mat(2,2));
fprintf('  G vector: [%.4f, %.4f] N·m\n\n', G_vec(1), G_vec(2));
fprintf('  tau_hip_lift = %.3f N·m\n', tau_dynamic(1));
fprintf('  tau_knee     = %.3f N·m\n\n', tau_dynamic(2));

% With FOS
tau_hip_lift_design = tau_dynamic(1) * FOS;
tau_knee_design     = tau_dynamic(2) * FOS;
fprintf('  With %.1fx FOS:\n', FOS);
fprintf('  tau_hip_lift = %.3f N·m (%.0f oz·in)\n', ...
      tau_hip_lift_design, tau_hip_lift_design*141.6);
fprintf('  tau_knee     = %.3f N·m (%.0f oz·in)\n\n', ...
      tau_knee_design, tau_knee_design*141.6);

%% ── STANCE PHASE TORQUES ──────────────────────────────────────────────
% During stance: static + body acceleration loads
a_body = v_walk / stance_time;   % body acceleration estimate (m/s²)

F_vertical   = (m_total * g) / n_stance;
F_horizontal = (m_total * a_body) / n_stance;
F_total_leg  = sqrt(F_vertical^2 + F_horizontal^2);

% Torque at hip and knee from ground reaction force
tau_hip_stance  = F_total_leg * (L1 + L2) * FOS;
tau_knee_stance = F_total_leg * L2 * FOS;

fprintf('Stance Phase Torques (GRF + body acceleration):\n');
fprintf('  Body acceleration:    %.3f m/s²\n', a_body);
fprintf('  Vertical GRF/leg:     %.2f N\n', F_vertical);
fprintf('  Horizontal GRF/leg:   %.2f N\n', F_horizontal);
fprintf('  Total force/leg:      %.2f N\n\n', F_total_leg);
fprintf('  tau_hip (with FOS):   %.2f N·m (%.0f oz·in)\n', ...
      tau_hip_stance, tau_hip_stance*141.6);
fprintf('  tau_knee (with FOS):  %.2f N·m (%.0f oz·in)\n\n', ...
      tau_knee_stance, tau_knee_stance*141.6);

%% ── IMPACT FORCE AT TOUCHDOWN ─────────────────────────────────────────
```

```
% Foot velocity at end of swing phase
v_foot_x = step_length / swing_time;     % horizontal (m/s)
v_foot_z = 2 * 0.08 / swing_time;        % vertical (m/s) — from arc height
v_foot   = sqrt(v_foot_x^2 + v_foot_z^2);

% Impact duration (assume 20ms for a moderately compliant foot)
dt_impact = 0.020;   % seconds

F_impact = (m_leg * v_foot) / dt_impact;
F_impact_FOS = F_impact * FOS;

fprintf('Impact Force at Foot Touchdown:\n');
fprintf('  Foot velocity at touchdown: %.3f m/s\n', v_foot);
fprintf('  Impact duration:            %.0f ms\n', dt_impact*1000);
fprintf('  Peak impact force:          %.2f N\n', F_impact);
fprintf('  With %.1fx FOS:             %.2f N\n\n', FOS, F_impact_FOS);

% Impact torque at hip
tau_hip_impact = F_impact_FOS * (L1 + L2);
fprintf('  Impact torque at hip:       %.2f N·m (%.0f oz·in)\n\n', ...
        tau_hip_impact, tau_hip_impact*141.6);

%% ── POWER CONSUMPTION ──────────────────────────────────────────
% P = tau * omega for each joint

P_hip_rot  = abs(tau_dynamic(1)) * omega1_peak;  % approx
P_hip_lift = abs(tau_dynamic(1)) * omega2_peak;
P_knee     = abs(tau_dynamic(2)) * omega3_peak;
P_per_leg  = P_hip_rot + P_hip_lift + P_knee;
P_all_legs = P_per_leg * 6;

% Add body electronics estimate
P_electronics = 20;   % W (Pi, Arduino, sensors)
P_total = P_all_legs + P_electronics;

fprintf('Power Consumption Estimate:\n');
fprintf('  Hip rotation power (per leg): %.2f W\n', P_hip_rot);
fprintf('  Hip lift power    (per leg): %.2f W\n', P_hip_lift);
fprintf('  Knee power        (per leg): %.2f W\n', P_knee);
fprintf('  Total per leg:               %.2f W\n', P_per_leg);
fprintf('  All 6 legs:                  %.2f W\n', P_all_legs);
fprintf('  Electronics:                 %.2f W\n', P_electronics);
fprintf('  TOTAL POWER:                 %.2f W\n\n', P_total);

%% ── TORQUE-SPEED CURVE ─────────────────────────────────────────
% Plot required torque vs joint speed across gait cycle
% Compare against motor capability

n_pts     = 100;
t_cycle   = linspace(0, step_time, n_pts);
tau_cycle = zeros(n_pts, 3);   % hip_rot, hip_lift, knee
```

```
omega_cycle = zeros(n_pts, 3);

for i = 1:n_pts
  t = t_cycle(i);
  phase = t / step_time;

  if phase <= swing_frac
    % Swing phase — inertial + gravity torques dominate
    scale = sin(pi * phase / swing_frac);  % sinusoidal profile
    tau_cycle(i,1) = abs(tau_dynamic(1)) * scale;
    tau_cycle(i,2) = abs(tau_dynamic(1)) * scale;
    tau_cycle(i,3) = abs(tau_dynamic(2)) * scale;
    omega_cycle(i,1) = omega1_peak * scale;
    omega_cycle(i,2) = omega2_peak * scale;
    omega_cycle(i,3) = omega3_peak * scale;
  else
    % Stance phase — gravity + GRF torques, low speed
    stance_scale = 0.3;
    tau_cycle(i,1) = tau_hip_stance * stance_scale / FOS;
    tau_cycle(i,2) = tau_hip_stance * stance_scale / FOS;
    tau_cycle(i,3) = tau_knee_stance * stance_scale / FOS;
    omega_cycle(i,1) = omega1_peak * 0.1;
    omega_cycle(i,2) = omega2_peak * 0.1;
    omega_cycle(i,3) = omega3_peak * 0.1;
  end
end

%% ── SUMMARY TABLE ──────────────────────────────────────────────────────
fprintf('=== Torque Summary (with %.1fx FOS) ===\n\n', FOS);
fprintf('  %-20s | %-15s | %-15s\n', 'Condition', 'Hip Lift (N·m)', 'Knee (N·m)');
fprintf('  %s\n', repmat('-',1,55));
fprintf('  %-20s | %-15.2f | %-15.2f\n', 'Dynamic (swing)',   tau_hip_lift_design, tau_knee_design);
fprintf('  %-20s | %-15.2f | %-15.2f\n', 'Static (stance)',   tau_hip_stance, tau_knee_stance);
fprintf('  %-20s | %-15.2f | %-15.2f\n', 'Impact (touchdown)',tau_hip_impact, tau_hip_impact/2);
fprintf('  %-20s | %-15.2f | %-15.2f\n', 'DESIGN MAXIMUM', ...
        max([tau_hip_lift_design, tau_hip_stance, tau_hip_impact]), ...
        max([tau_knee_design, tau_knee_stance, tau_hip_impact/2]));

%% ── PLOTS ──────────────────────────────────────────────────────────────

% Figure 1: Torque and speed across gait cycle
figure('Name', 'Dynamic Joint Loads', 'NumberTitle', 'off');
subplot(3,2,1);
plot(t_cycle*1000, tau_cycle(:,1), 'b-', 'LineWidth', 2);
xline(swing_time*1000, 'k--', 'Swing|Stance');
ylabel('Torque (N·m)'); title('Hip Rotation Torque'); grid on;

subplot(3,2,3);
plot(t_cycle*1000, tau_cycle(:,2), 'r-', 'LineWidth', 2);
xline(swing_time*1000, 'k--', 'Swing|Stance');
ylabel('Torque (N·m)'); title('Hip Lift Torque'); grid on;
```

```
subplot(3,2,5);
plot(t_cycle*1000, tau_cycle(:,3), 'g-', 'LineWidth', 2);
xline(swing_time*1000, 'k--', 'Swing|Stance');
ylabel('Torque (N·m)'); xlabel('Time (ms)'); title('Knee Torque'); grid on;

subplot(3,2,2);
plot(t_cycle*1000, rad2deg(omega_cycle(:,1)), 'b-', 'LineWidth', 2);
xline(swing_time*1000, 'k--', 'Swing|Stance');
ylabel('Speed (deg/s)'); title('Hip Rotation Speed'); grid on;

subplot(3,2,4);
plot(t_cycle*1000, rad2deg(omega_cycle(:,2)), 'r-', 'LineWidth', 2);
xline(swing_time*1000, 'k--', 'Swing|Stance');
ylabel('Speed (deg/s)'); title('Hip Lift Speed'); grid on;

subplot(3,2,6);
plot(t_cycle*1000, rad2deg(omega_cycle(:,3)), 'g-', 'LineWidth', 2);
xline(swing_time*1000, 'k--', 'Swing|Stance');
ylabel('Speed (deg/s)'); xlabel('Time (ms)'); title('Knee Speed'); grid on;

sgtitle('Joint Torque and Speed Across One Gait Cycle');

% Figure 2: Power consumption breakdown
figure('Name', 'Power Consumption', 'NumberTitle', 'off');
subplot(1,2,1);
pie([P_hip_rot*6, P_hip_lift*6, P_knee*6, P_electronics], ...
   {'Hip Rotation', 'Hip Lift', 'Knee', 'Electronics'});
title('Power Distribution');

subplot(1,2,2);
components = [P_hip_rot, P_hip_lift, P_knee, P_per_leg, P_all_legs, P_total];
labels = {'Hip Rot\n(per leg)', 'Hip Lift\n(per leg)', 'Knee\n(per leg)', ...
         'Total\n(per leg)', 'All Legs', 'Total\n+Electronics'};
bar(components, 'FaceColor', [0.2 0.5 0.8]);
set(gca, 'XTickLabel', {'Hip Rot', 'Hip Lift', 'Knee', 'Per Leg', 'All Legs', 'Total'});
ylabel('Power (W)'); title('Power by Component'); grid on;

% Figure 3: Ground reaction forces
figure('Name', 'Ground Reaction Forces', 'NumberTitle', 'off');
t_stance_vec = linspace(0, stance_time, 50);
GRF_v = ones(50,1) * F_vertical;
GRF_h = (m_total * a_body / n_stance) * sin(pi * (0:49)'/49);

subplot(2,1,1);
plot(t_stance_vec*1000, GRF_v, 'b-', 'LineWidth', 2);
ylabel('Force (N)'); title('Vertical GRF per Stance Leg'); grid on;
yline(F_vertical*FOS, 'r--', sprintf('Design load (%.0fN)', F_vertical*FOS));

subplot(2,1,2);
plot(t_stance_vec*1000, GRF_h, 'r-', 'LineWidth', 2);
```

```
xlabel('Time in stance (ms)'); ylabel('Force (N)');
title('Horizontal GRF per Stance Leg (Propulsive)'); grid on;

sgtitle('Ground Reaction Forces During Stance Phase');
```