Visual Odometry Thinking While Driving for the Curiosity Mars Rover's Three-Year Test Campaign: Impact of Evolving Constraints on Verification and Validation

Mark Maimone, Nikunj Patel, Anna Sabel, Alexandra Holloway, and Arturo Rankin
Jet Propulsion Laboratory, California Institute of Technology 4800 Oak Grove Dr.
Pasadena, CA 91109
Mark.Maimone@jpl.nasa.gov

Abstract—**Over the first 9 years of the Mars Science Laboratory (MSL) Curiosity rover's surface mission, more than 87% of its driving was performed using Visual Odometry (VO). The benefits of using VO during driving are that it minimizes rover position uncertainty and can be used to monitor wheel slip, halting a drive if excessive wheel slip is occurring. The VO implementation onboard Curiosity acquires and processes VO images in between drive steps while the rover is stationary.**

A VO Thinking While Driving (VTWD) flight software capability has been developed to enable the processing of VO images during rover driving, increasing the distance Curiosity can drive using VO during a given time period up to as much as 1.75x total distance. Verification and Validation (V&V) of this capability has been challenging due to impacts from the COVID-19 pandemic and unavailability of the JPL Mars Yard outdoor test site. The VTWD V&V test procedures were modified to use a small indoor space with Mars-like terrain. This paper describes the 3 year V&V effort under challenging conditions to approve the VTWD capability for use on the Curiosity rover.

TABLE OF CONTENTS

1. INTRODUCTION

The Mars Science Laboratory (MSL) Curiosity rover is typically driven using a Visual Odometry (VO) capability to minimize position uncertainty and monitor slip onboard, enhancing mission safety. The current VO implementation acquires and fully processes VO images while stopped between drive steps [1]. The time required for that processing more than halves the drive rate achievable without VO, significantly impacting how far Curiosity can drive each sol. Noting that the CPU had significant idle time while in motion during drive steps, a new VO Thinking While Driving (VTWD) capability was developed to better optimize CPU usage, performing most of the VO processing while the rover is in motion. The new capability therefore reduces idle CPU time while driving, which allows drives to complete more quickly while using less power overall.

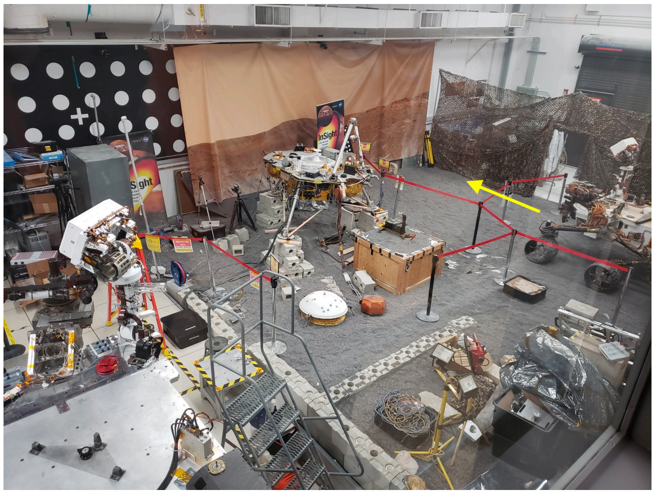

Figure 1. View of the 3m x 9m edge of the In Situ Instrument Lab (ISIL) test area available for mobility testing. The yellow arrow indicates the limited drive motion that was possible, after removing the stanchions in front of the rover. Annotation added to existing photograph [12].

The VTWD capability runs the existing processing ("thinking") step in parallel with vehicle motion, increasing VO-enabled drive distance by up to 1.75x over a given planning period. VTWD was developed initially for MSL as a software patch, and was later integrated into a complete flight software (FSW) monolithic image. Verification and Validation (V&V) consisted of formal software and hardware testing processes to demonstrate and verify the functions of the new VTWD capabilities, and to ensure that no other MSL capabilities would be adversely affected by this enhanced capability.

The V&V campaign began by verifying the VTWD Hot Patch implementation in JPL's Mars Yard, a spacious 2,000 m^2 area, resembling Martian terrain and containing many features useful for VO. However, more than halfway through the V&V campaign, major repairs to the Mars Yard pushed the rover testbed into a small indoor space with Mars-like terrain called the In-Situ Instrument Laboratory (ISIL). The new ISIL space, shown in Figure 1 from [12], is about 3 m by 9 m, allowing up to about 4 m of motion to be commanded. Combined with a typical 1 m waypoint tolerance value, that

978-1-6654-3760-8/22/$31.00 ©2022 IEEE

meant that most ISIL-based drives terminated after just 3 m of motion; enough to confirm VTWD was working, but not enough to develop useful drive rate statistics.

MSL project management also decided to integrate what had been a patch into a new flight software build. These changes required that the team restart V&V from the beginning, due to software changes necessary to integrate the new capability following nominal Development-phase coding standards. Space and terrain constraints in ISIL prevented exercising the mobility subsystem using existing test cases with VTWD, and led to the development of new, specialized procedures to accommodate VTWD test scenarios. These include using VO during a precision approach to a proximity science target, on featureless terrain, and while driving on sloped terrain. Testing was also impacted by evolving responses to the 2020 COVID-19 pandemic and remote work, resulting in constraints in accessing and staffing the testbed for VTWD shifts. Finally, one of the two testbed cameras used for VO became unavailable mid-VTWD V&V campaign, leading to an urgent workaround.

This paper describes the nominal usage of VO during past mission operations, the new VTWD capability and its expected operations impact, and the 3-year V&V campaign which began when patch development was complete and includes the test-procedure pivot to accommodate driving in a tiny space. The new software is expected to be deployed on Curiosity in mid-2022.

2. DRIVING ON MARS AND VISUAL ODOMETRY

Curiosity and NASA's others Mars Rovers are operated by teams of scientists and engineers [4]. The Rover Planner (RP) team is charged with creating the specific sequences of driving commands used to move the rover. RPs decide what mode of driving should be used based on the appearance of the nearby terrain in rover-acquired images, orbital terrain images, input from the science team regarding how far to go and where to stop the rover, and engineering tradeoffs between the various modes. Those trades include power, data volume, duration, and choosing which vehicle safety systems to enable.

The vehicle is capable of three basic drive modes, summarized in Table 1. Directed drives are the fastest and follow RP commands without using vision to understand the terrain, VO drives run at less than half the Directed drive speed and measure rover position at each step and provide slip numbers useful for fault monitoring, and Autonav drives can be 4-10 times slower than Directed drives and enable the rover to choose its own path to drive safely around hitherto unseen obstacles to a series of waypoint goals. See [3] for specific performance numbers.

VO can be commanded in several modes, either all the time or in a Slip Check mode. Driving aluminum wheels on rocky and sandy terrain often results in some amount of slippage, and Curiosity has no sensors able to measure that positional slip other than by processing images acquired by its mast-mounted Navigation Cameras (NavCams) using VO flight software. VO re-localizes the rover position with a precision better than 2% of the distance driven, and a typical step distance between acquiring VO images is 1 m. Knowing the rover's precise location enables the flight software to calculate slip fractions, and use them as fault detection thresholds.

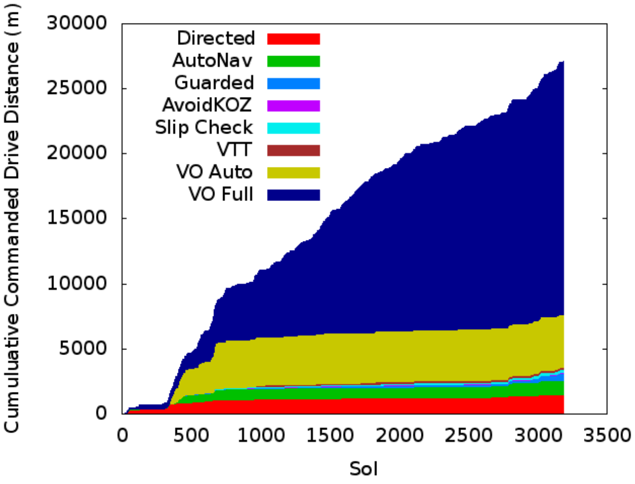

Figure 2. Curiosity cumulative commanded odometry by drive mode. This data spans nearly the first nine years of the mission, August 6, 2012 through July 20, 2021

The Slip Check mode was created following the Opportunity Rover's sol 446 embedding into Purgatory Ripple. On that sol, Opportunity commanded over 50 m of motion, but only moved 2 m in reality [6]. At the end of of all that driving its wheels were 80% embedded in Martian sand. Ever since then, Mars Rover operations have been subject to a driving constraint that we must not command more than 20 m without checking at least once to measure whether forward progress is still being made.

Table 1 summarizes the three VO modes available for use. VO Full mode uses VO processing throughout an entire drive segment. VO Slip Check will drive up to 19 m without using VO, but will enable it for the final meter. That guarantees that we will command no more than 20 m of motion without confirming that the rover is not getting embedded into sand or otherwise slowing down. VO Auto alternates between VO Full and VO Slip Check modes autonomously; the change from Slip Check to Full mode occurs if any of four triggers suggests the rover might be slowing down [3].

RP's typically command long drive segments between waypoints using the MOB_GO_TO command. That command drives long distances by moving in short steps (nominally 1 m long when running VO), and always selects the best path forward *given its current position knowledge*. When VO is active, the command waits for VO processing to complete before selecting the next step. That enables it to take any slip that occurred during that step into account, steering autonomously back toward the goal if needed. The more slip the rover encounters, the more corrections will be needed using arcs with higher curvatures; and changing the steer angles will slow down the overall drive rate somewhat due to the time needed to steer and the slightly slower overall drive rate during tighter arcs. See Figure 3 for an illustration of how the rover position knowledge changes during VO drives with slip.

For most of the mission, RP's have chosen to enable the VO safety feature, especially since noticing the wheel skin was getting torn by the local terrain around sol 455 [2], [5]. That, and an initial embedding on sol 672, led to the great reliance on VO over other driving modes. Figure 2 shows that VO has been used for 87% of all drives during the first nine years of

Table 1. Explanation of most common drive modes in Figure 2.

Mode	Sub-mode	Relative speed	Description
Directed drive		1	Fastest drive mode; no active vision sensing onboard
Autonav		4–10x slower	Good for safe driving; rover chooses its own path based on geometric hazards, keepout zones, and waypoint goals
Visual Odometry			Good for slip and positional fault monitoring
	VO Full	2x slower	Uses VO processing for entire drive segment
	Slip Check	1.1x slower	Uses VO to periodically check for wheel embedding
	VO Auto	1.1–2x slower	Switches autonomously between **Slip Check** and **VO Full**

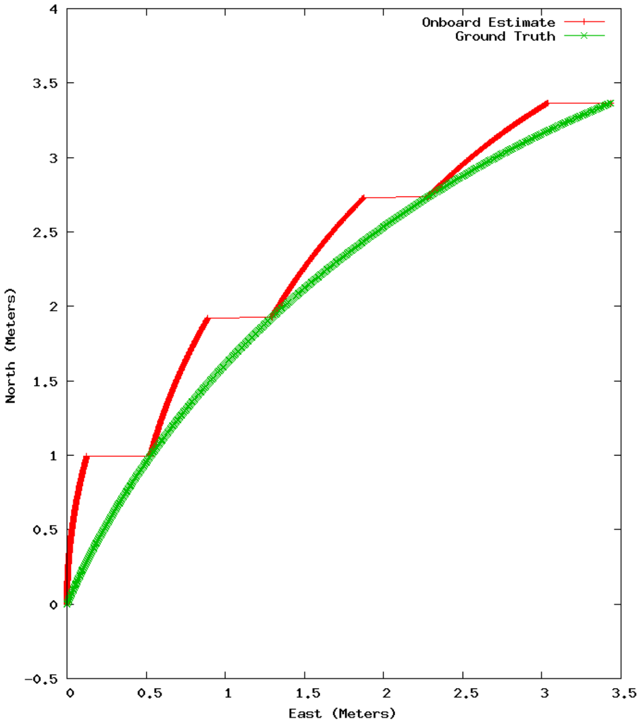

Figure 3. Illustration of the impact of **nominal VO** on the onboard position estimate. In this simulated plot, the rover starts at (0,0) facing north (up), and is commanded to move along an ARC 4 m forward with 1 radian of heading change (57°) to the right, under conditions that result in 40% slip to the East at each step. The green path indicates the actual path the rover would take in these conditions (including the 40% slip at each step), the red path shows the expected onboard position estimate. In this plot, the VO processing occurs at the end of each 1 m step, which has been the nominal mode for the first nine years of operations.

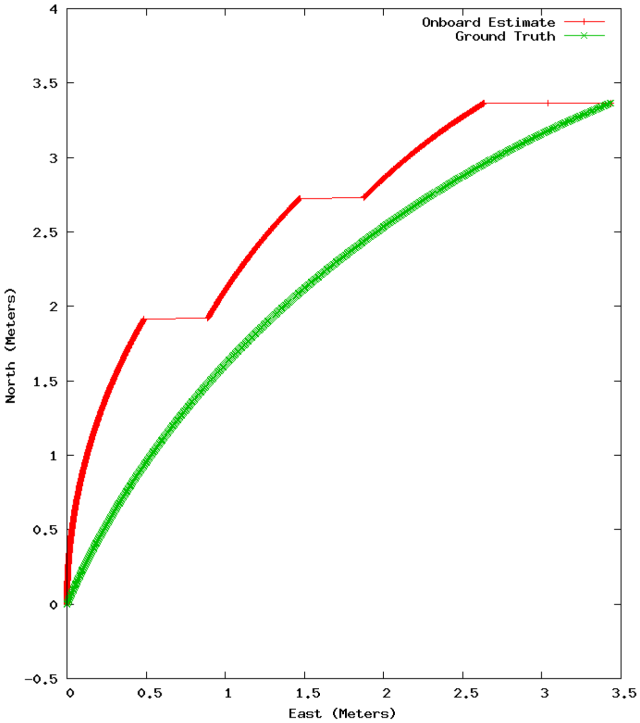

Figure 4. Impact of having **VTWD enabled** on the onboard position estimate. In this simulated plot, the rover starts at (0,0) facing north (up), and like Figure 3 is commanded to move along an ARC 4 m forward with 1 radian of heading change (57°) to the right, with 40% slip to the East. The green path shows the actual path, the red path shows the onboard position estimate. In this plot, the VO processing is presumed to finish after driving 100% of each 1 m step, corresponding to the CPU plot in Figure 6 where the purple NAV computation time extends beyond the dark blue MOT activity.

operation, primarily in VO Full mode and sometimes in VO Auto mode. By far, the majority of the drive distance (over 91%) has been in VO Full mode ever since sol 672 (i.e., near the end of the second year of operations). The average speed in this mode has been approximately 46 m/hr (see "Go to: VO_Full off off DIRECTED off" entries in Figures 20, 21 in [3]).

Software Implementation and System Impact

VO processing is implemented in Curiosity's mobility flight software. From landing in 2012 through calendar year 2021, the strategy for VO processing has been: Stop the rover; Collect a stereo pair of images; Compare this pair to one captured earlier (usually 1 m before); Calculate the 6-DOF position and pose change between the two stereo pairs; Apply only the position update to the onboard position estimate; Plan the next drive step using the latest information; Drive another step; Repeat [6]. The CPU usage pattern associated with this strategy is illustrated in Figure 5; dark blue indicates use of the drive motors for each 1 m step (the MOT flight software task); purple indicates the time spent comparing the pairs of stereo images and generating the motion estimate ("thinking" in the NAV flight software task); and the yellow indicates idle CPU time while the rover is driving.

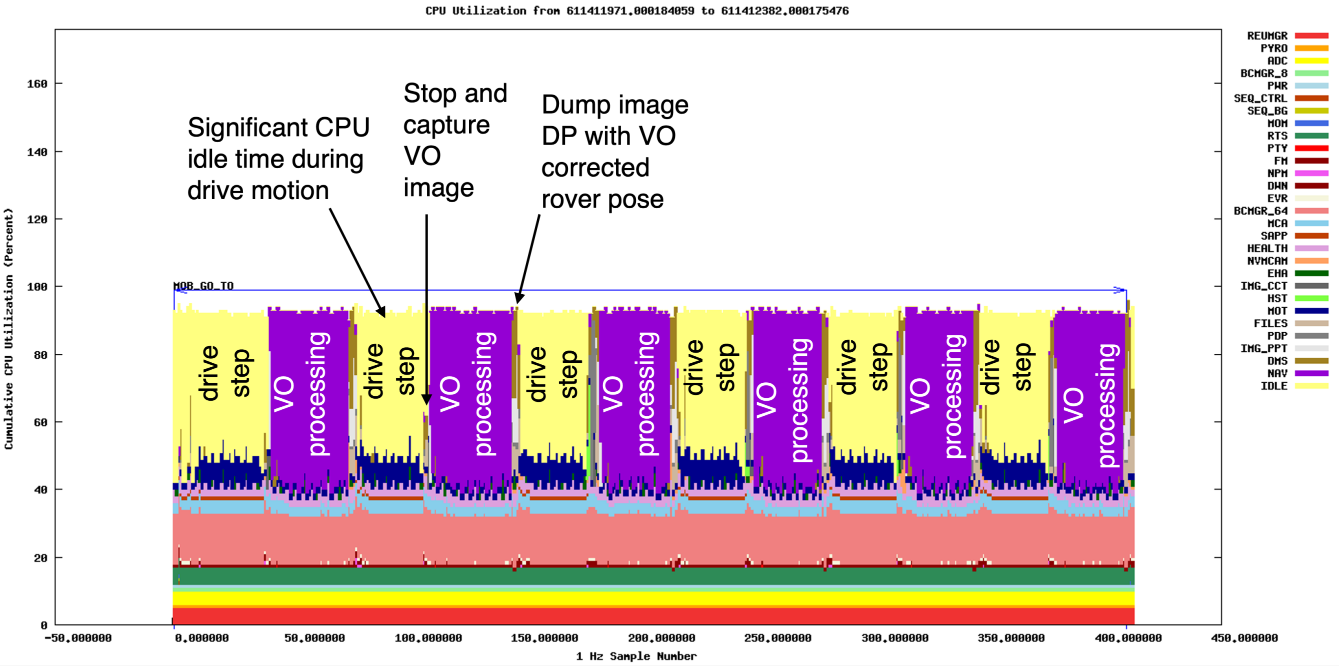

Figure 5. CPU Usage vs Time with **Nominal VO** for VO Full – 6 m drive, 52.9 m/hr drive rate. The plot shows the percentage of CPU time spent by dozens of VxWorks FSW tasks (the vertical scale goes up to 100% cumulative CPU Usage), with CPU Usage percentages calculated over 1 second intervals. **Light yellow on top indicates CPU idle time**, other colors correspond to other FSW tasks; only tasks with at least 0.5% CPU Usage appear in the legend. The vertical ordering of task colors in the plot is the reverse of the the task colors given in the legend (e.g., the task named at the top of the legend (REUMGR) corresponds to the bottommost red line in the plot). The plot was generated using our CPU Utilization Statistics Plotting tool, a.k.a. CUSP [7].

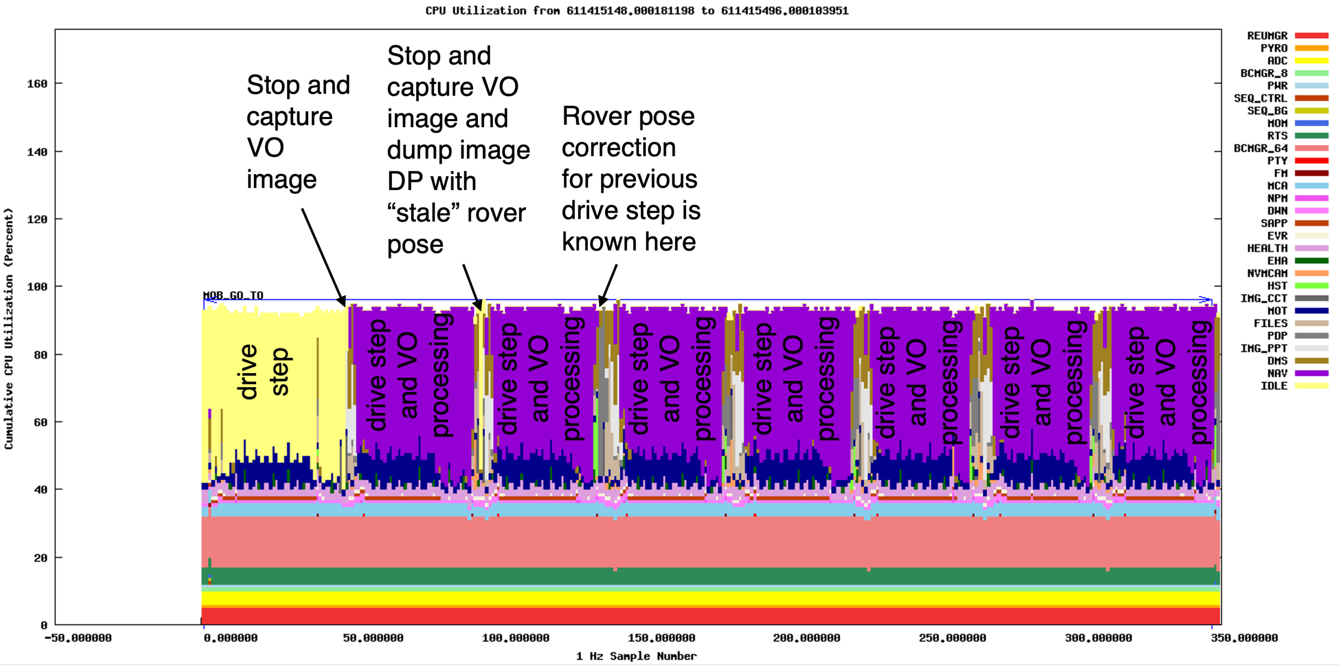

Figure 6. CPU Usage vs Time with **VTWD enabled** for VO Full: 8 m drive, 83.2 m/hr drive rate (about a 1.6x improvement over the rate in Figure 5). Yellow CPU idle time goes away after the first step. The plot was generated using our CPU Utilization Statistics Plotting tool, a.k.a. CUSP [7].

This strategy ensures that the rover has the best position knowledge after each 1 m step. Any slip event will be detected immediately after the 1 m step in which it occurred. All steering choices will be made based on the actual rover position. However, this knowledge comes at the cost of leaving the CPU idle much of the time; see the yellow background areas in Figure 5. That idle time, combined with the time needed to process the VO image pairs, reduces the overall VO drive rate from over 100 m/hr in Directed drives to just 40-50 m/hr in VO drives. The gain in safety has justified its use for years, but several issues prompted us to consider speeding up the implementation.

Drives require a significant amount of power for the CPU, the Rover Inertial Measurement Unit (RIMU), and wheel and steering motors. If the drive duration could be shortened, CPU and RIMU power needs would be similarly reduced. As Curiosity's mission continues well beyond the primary goal of 1 Martian year (about 1.9 Earth years), diminishing power generation will impact how often, how long and how far future drives will be able to go. For example, the Spirit RP team often had to trade saving power against using its VO capability as they sought a "winter haven" location during its final year of operation. So an update that shortens drive duration while preserving mission safety will be very beneficial in Curiosity's future extended missions.

Also, the Mars 2020 Perseverance rover mission was already planning to improve its onboard autonomy architecture by enabling autonomous image processing and path selection without having to stop, a.k.a. "Thinking While Driving" [8]. Both projects agreed that even a limited demonstration of the capability on Curiosity would benefit Mars 2020 as well. That led us to develop a new, faster capability called *VO Thinking While Driving* for Curiosity.

3. VO THINKING WHILE DRIVING

Our strategy for speeding up the overall drive rate is to start driving *immediately* after collecting each new image pair. That allows the VO processing to take place while the wheels are in motion, making better use of the CPU during a long drive segment and virtually eliminating the CPU idle time. This is illustrated in Figure 6, which demonstrates far less yellow idle time than the original method in Figure 5. We also chose to restrict this change to the execution of the `MOB_GO_TO` command, which is the primary command used to drive the long straight segments between waypoints. Most of the long range driving employs this command, and restricting flight software changes to one command meant it would be simpler to implement this capability as a flight software patch, since fewer source code changes would be needed.

This change from the original strategy will impact the accuracy of `MOB_GO_TO` drives. The flight software will not be able to detect slip events as quickly while using VTWD; if large slip occurs, it will not be noticed until after **two** 1 m steps have completed. But we felt this was a reasonable compromise, especially since our Slip Check rule allows as much as 20 m of driving between VO Slip Checks. It will also impact the onboard position estimate itself during `MOB_GO_TO` commands. Instead of having perfect position knowledge after each step, each end-of-step position estimate will fail to include the last step's (as yet unmeasured) slip. Ultimately, once the rover reaches its goal, the final VO update will be processed before the command completes, thus ensuring that subsequent commands and computations have the best positional knowledge, as in the non-VTWD mode. So the only impact to mobility knowledge is the additional position uncertainty during execution of the `MOB_GO_TO` command.

The change is illustrated in Figures 3 and 4. Both plots illustrate the motion of the rover origin (located midway between the center pair of wheels) during an ARC motion of 4 m forward and 1 radian (57°) heading change to the right in the presence of 40% slip to the east. This is a circumstance the rover might encounter, e.g., while driving cross-slope with significant positive roll. The green dots show the rover origin in their simulated "ground truth" position, including the 40% eastward slip. The red dots show the position of the rover origin as it will be estimated onboard. VO updates appear as thin lines "jumping" from one red dot to another. Figure 3 shows the effect of non-VTWD VO on the onboard position estimate; position knowledge does not reflect slip updates during each 1 m step, but the position *is immediately corrected to match the simulated "ground truth" before driving the next step*. In contrast, Figure 4 shows the behavior expected during VTWD. In VTWD, the red and green dots will only align at the start of the drive (before any motion has occurred) and at the end (when all VO processing has completed); the red dots never align with the green dots during the drive, illustrating the additional uncertainty in its position estimate throughout the 4 m drive segment.

Software Implementation and System Impact

The software changes needed to implement VTWD were initially implemented and unit tested in a Linux environment. Parameters were added to control when VTWD would be invoked, and whether to allow it during Autonav drives. Temporary storage was found for the image data, so that they could be cached for later processing after receiving a message from a different flight software task (the Mobility Manager, or "MOM" task) confirming that driving had begun. And these changes were made in a manner to minimize differences from the existing flight software, in the expectation of implementing this capability using a Hot Patch rather than a full FSW build [9]. Validating a patch is in general easier than generating, validating and preparing a full flight software build for upload.

However, deployment of this capability was overtaken by events, as we describe in the next section.

4. V&V CAMPAIGN: PATCH TO INTEGRATION

Initial Software Patch Development

Development of the VTWD flight software capability began in 2015. But during the first test of the capability on the flight-like engineering model rover in 2015, a software bug caused an exception when attempting to run a basic VTWD-enabled `MOB_GO_TO` drive command. The exact problem was difficult to diagnose because it had only occurred on the full system testbed, and it was not repeatable in the single-threaded development test environment. Fixing it became a low priority task, overridden by various anomalies, issues and concerns.

Development stalled until 2019, when the software team was given access to a new, multithreaded linux process test framework, the Surface System Development (SSDev) simulation environment [11]. The bug, which was only manifest in a multithreaded framework, was then quickly replicated and resolved, and development testing continued.

Patch development testing occurred using the Vehicle System Testbed (VSTB) in the Mars Yard from March of 2019

Figure 7. VSTB in the Mars Yard with Total Station Tracking System

through August of 2019. Figure 9 shows our VSTB rover in the Mars Yard. The VSTB is an engineering model of the Curiosity rover with flight-like mobility components, avionics and cameras, making it our highest-fidelity testbed for mobility operations.

In September 2019, the VTWD capability was officially proposed and approved as a Hot Patch to the R12.0.3 version of Flight Software currently running onboard the Curiosity rover. Approval was conditional on additional parallelism testing to confirm that normal rover activities could still be run in parallel with a drive.

V&V Campaign Overview and Test Facility Challenges

In order to approve a flight software change to the mobility subsystem, even a patch, the mobility team testing has to meet certain requirements to ensure optimal and safe operation of the Curiosity rover on Mars.

These requirements range from testing off-nominal cases of a drive to conducting basic drive commanding. Such regression tests help ensure that any new capability has not inadvertently impacted other parts of the system. We are mindful of other missions that have been negatively impacted (even lost) due to such changes [10].

For example, off-nominal cases to test include potentially harmful interactions with terrain that exercise the vehicle's autonomous capabilities to protect itself. To test one such off-nominal scenario, the team would drive the rover over a sandbag and induce a fault response by lowering the safe suspension limit criteria for the rover, making it extremely sensitive to the slightest suspension change. Hence, when the rover drives over the sandbag, its suspension would go higher above the set limit and the flight software would stop the drive. Figure 7 illustrates how the sandbags are placed for such tests.

For approval of new mobility subsystem flight software code, all the off-nominal cases for a drive are required to be retested. In addition, all nominal drive modes are required to be retested so as to ensure that the new code did not break any existing capabilities.

An example of a nominal drive mode would be to test the VO capability of the rover. As VO measures the distance moved during a drive command, testing it would require an external

Figure 8. A sandbag is placed in front of each front wheel prior to a forward drive to induce a suspension fault response.

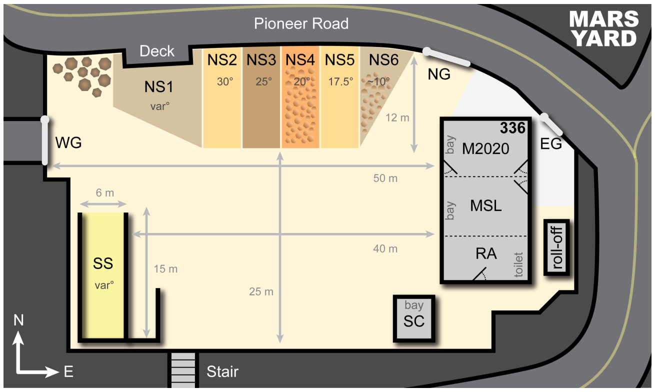

Figure 9. Schematic overview of JPL's Mars Yard test area.

position measurement device to ensure that the drive distance reported is accurate.

The test team uses a Total Station Tracking System to track the position of the rover and compare that data with the values reported by VO processing. These data have been collected but have not yet been analyzed. Figure 9 illustrates the tracking system. The red prism mounted on the rover gets tracked by the Total Station that is operated by a tester.

The Mars Yard was originally developed to create an environment for conducting extensive mobility testing for rovers and other vehicles. The 2,000 m^2 outdoor space has a large flat surface for driving, and slopes designed with different terrain surfaces for testing rover capabilities at different tilts on different terrains; see Figure 8 for a graphical view.

Due to remodeling of the area for the development of a new rover (Perseverance), the VSTB was moved to an indoor testing location known as the In-Situ Instrument Laboratory (ISIL) in December of 2019. In that facility, the test plan's total commanded drive distance had to be reduced to fit within the narrow confines of the shared ISIL test area, and a new camera pointing strategy was adopted. Figure 10 shows the VSTB parked in ISIL next to a camouflage tarp.

To process VO properly, sets of clear and distinguishable features must be visible between two sets of images of the terrain near the rover. The environment in ISIL could not support that due to multiple factors: reflections from viewing glass areas, mostly-white walls, people and/or objects changing location between image updates (in contrast to the Martian environment which is very static, i.e. unchanging except for shadow and Sun-incidence angle effects over time), repeated patterns in the doorway and wall-mounted camera calibration features, and specular reflections from other projects' metallic equipment in the shared laboratory space. Therefore, in order to enable proper VO convergence, the team installed the camouflage tarp and commanded the VSTB rover to point its cameras in that direction while driving.

V&V Campaign Roadblocks, and the Evolution of the Test Plan

As the V&V campaign encountered various delays, changes to the original test plan were made to ensure that latest flight software code changes as well as facility changes were considered.

VTWD testing began on the VSTB in ISIL in late January 2020 but was put on hold after just a few tests due to higher priority testing, and later, due to mandatory lab-wide telework as a response to the COVID-19 pandemic. In March 2020, all MSL testing was halted until further notice.

During this time, the team was able to make changes to the flight software code in order to improve the VTWD behavior. Earlier testing had discovered that some of the lesser-used VO modes did not work with the VTWD changes (VO Slip Check and VO Auto modes). That initially led us to create new flight rules, guidance to the operations team to be aware of this problem during mission operations. Curiosity's operation team tracks multiple flight rules to ensure that the rover operations are conducted with safety as its highest priority. But the delayed testing gave us extra development time and allowed us to correct the issue; so the test plan was edited to include the new capabilities and we were able to remove those additional flight rules.

VTWD was originally developed as a flight software Hot Patch that would be installed on the current flight software, and the team could turn on and off the new capability whenever needed. As it was separate code that was being installed on the currently active version of rover flight software, the mobility team would have been required to conduct an extensive test campaign to make sure that the active flight software was not altered in any harmful manner.

During the mandatory telework period imposed by COVID restrictions, Curiosity's operations team decided to include the VTWD Capability in R13, the next major MSL flight software release in active development. This was driven by the understanding that it would become safer and easier to patch R13 in the future if all spacecraft patches, both existing and in development, were incorporated into the monolithic build [13]. It also eased the burden on the test team from having to validate both the patch version and the R13 integrated version of VTWD.

As VTWD would now be a part of a new, complete flight software release, the mobility team would be required to only test the commands that were affected by the VTWD code and leave the remaining mobility flight software validation testing to the systems regression team. Furthermore, parts of the old test campaign, such as testing installation of the patch, turning on and off of the patch, and so forth, were abandoned

Figure 10. VSTB in ISIL with camouflage tarp to ensure VO convergence

as they would no longer be needed.

The team also was required to alter the commands for testing previously used in the expansive Mars Yard due to the space restrictions in ISIL. Given that the driving area in the ISIL could only account for a drive distance of 4 m forward/backward, the team had to come up with ways to conduct most of its testing using that small drive distance.

Resuming the VTWD V&V campaign was additionally complicated during this period by testbed staffing requirements. VSTB testbed safety rules require that a minimum of two operators be present for any tests involving use of vehicle motors or actuators or complex test setups. That was challenging to accomplish with on-lab operations severely limited.

Two-person VSTB flight software test shifts resumed after management consideration and approval in the fall of 2020, with VTWD V&V testing resuming in late October. "Safe at Work" training was required for all personnel physically using JPL's facilities, and lab-wide safety measures including masking, social distancing, daily health attestation, and cleaning were required during all testbed operations.

This iteration of the test campaign continued until another bug was discovered in one of the VO modes, which necessitated further software development work to characterize and resolve.

As this work was ongoing, in April of 2021 the VSTB lost its stereo vision capabilities due to a hardware issue with one of the two mast-mounted NavCams.

Development work was finally completed and validated via Linux-based Unit Testing and VxWorks-based simulation using our Work Station Test Set (WSTS) environment. However, the VSTB Engineering Model test campaign had been formally on hold until the NavCam issue was resolved, only recently in October 2021.

5. RESULTS

VSTB testing has successfully demonstrated an effective speedup ranging from 29–75% in the overall traverse rate using the new VTWD capability. Table 2 summarizes some of the effective drive rates achieved during our individual test runs. Actual performance on Mars will likely fall into a similar range, though it is likely to vary with drive step size, VO processing duration, CPU load, and the variety of arc curvatures selected for driving.

A 50% drive rate speedup will enable us to drive 50% farther

Table 2. Test drive durations, with and without VTWD, as run on the VSTB in the Mars Yard test area. Only drives with distance 4 m or above and nominal step size of 1 m are shown (all ISIL drives were shorter, hence are not shown). Average drive rate without VTWD (total distance / total duration) is 51.5 m/hr, average drive rate with VTWD is 77.5 m/hr, indicating a speedup of 50% in the average rate. Durations vary due to different test configurations; some were deliberately trying to induce slip to show the drive would still complete successfully, which increased the duration when non-straight arcs were autonomously selected to compensate (requiring extra time to steer, and slowing down the forward progress rate).

Drive date	**Spacecraft Clock Start**	**Distance driven**	**Duration**	**Effective drive rate**	**CPU Usage**
Without VTWD					
March 2, 2019	607093209	4 m	294 s	49.0 m/hr	Figure 5
May 17, 2019	611411971	6 m	411 s	52.6 m/hr	
July 8, 2019	615885099	8 m	553 s	54.0 m/hr	
With VTWD					
March 2, 2019	607093643	9 m	377 s	86.0 m/hr	Figure 6
May 17, 2019	611414649	6 m	267 s	80.9 m/hr	
May 17, 2019	611415148	8 m	348 s	82.8 m/hr	
July 8, 2019	615888270	9 m	447 s	72.5 m/hr	
July 8, 2019	615890453	6 m	308 s	70.1 m/hr	
July 8, 2019	615893277	9 m	435 s	74.5 m/hr	

using the same drive duration. Conversely, it will also enable Curiosity to achieve the same distance in less time, with a savings of 33% in duration compared to the original drive rate, and a comparable savings in power consumed by the CPU, RIMU, and Rover Motor Controller Assembly (RMCA).

6. CONCLUSION

Visual Odometry Thinking While Driving (VTWD) is expected to speed up Curiosity's most-commonly-used drive mode by up to 75% (50% on average). The reduction in resources needed for VO drives will make it easier to schedule such drives and free up resources for other science and engineering activities on a day-to-day tactical basis in the near term. And it will enable continuing use of VO safety features in drives over the long term, even as available power diminishes after more than a decade of operation on Mars.

Always with an eye toward the future, the testbed team persevered through multiple bottlenecks to validate this capability. A challenging bug was resolved via an improved multithreaded simulation framework, tests were modified to work in a much smaller space while the Mars Yard was rebuilt for Perseverance rover development, the software and test plan were adapted for direct inclusion in the next official flight software release, the team learned how to deal with the challenges presented by COVID-19 impacts on lab operations, and the failure of testbed hardware led to further test refactoring into different venues.

As of November 2021, the NavCam hardware repairs have been completed, and the VSTB has just been made available for continued testing in the Mars Yard. Once the test V&V campaign has completed for VTWD, the other R13 capabilities and systems regression tests, we will be able to formally approve and deploy this software onto the Curiosity Mars Rover. This is currently expected to occur in early 2022.

ACKNOWLEDGMENTS

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The authors would like to thank the Mars Science Laboratory and Mars 2020 Programs for supporting this research.

Thanks also to Todd Litwin for porting his SSDev multithreaded simulation framework from Mars 2020 into the MSL FSW environment, helping us to quickly discover and correct an early implementation bug; Evan Graser for Total Station and testbed activity leadership; Kimberly Rink and Freddy Wang for testbed support; Jennifer Herman for power usage discussions; and Neil Abcouwer for helpful code review comments.

REFERENCES

[1] A. E. Johnson, S. B. Goldberg, Yang Cheng and L. H. Matthies, ”Robust and Efficient Stereo Feature Tracking for Visual Odometry,” 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 39-46, doi: 10.1109/ROBOT.2008.4543184.

[2] O. Toupet, J. Biesiadecki, A. Rankin, A. Steffy, G. Meirion-Griffith, D. Levine, M. Schadegg, M. Maimone, “Terrain-adaptive Wheel Speed Control on the Curiosity Mars Rover: Algorithm and Flight Results,” Journal of Field Robotics, Volume 37, Issue 5, Special Issue on Space Robotics, August 2020, August 26, 2019, 699 - 728.

[3] A. Rankin, M. Maimone, J. Biesiadecki, N. Patel, D. Levine, O. Toupet, “Mars Curiosity Rover Mobility Trends During the First 7 Years,” Journal of Field Robotics, January 9, 2021, 1-42.

[4] M. Gildner, A. R. Allbaugh, A. Mishkin, S. Algermissen, M. Van Kirk, D. Ellison, C. Bridge, T. Stough, A. Stroupe, “Commanding Curiosity from the couch: MSL

remote operations, challenges, and path ahead”

[5] A. Rankin, N. Patel, E. Graser, J. Wang, K. Rink, “Assessing Mars Curiosity Rover Wheel Damage”, IEEE Aerospace Conference, Big Sky, March, 2022.

[6] M. Maimone, Y. Cheng, L. Matthies, “Two Years of Visual Odometry on the Mars Exploration Rovers.” Journal of Field Robotics, Volume 24, 2007, 169–186.

[7] M. Maimone, “Visualizing Multi-process CPU Utilization Using CUSP”, IEEE Aerospace Conference, Big Sky, March, 2022.

[8] M. McHenry, N. Abcouwer, J. Biesiadecki, J. Change, T. Del Sesto, A. Johnson, T. Litwin, M. Maimone, J. Morrison, R. Rieber, O. Toupet, P. Twu, “Mars 2020 Autonomous Rover Navigation,” Proceedings of the 43rd AAS Rocky Mountain Section guidance, Navigation and Control Conference, Breckenridge, Colorado, 1/30/2020, 867-880.

[9] E. Benowitz, M. Maimone, “Patching Flight Software on Mars” Workshop on Spacecraft Flight Software, Laurel, MD, USA, 27 October 2015.

[10] D. J. Mudgway, “Telecommunications and Data Acquisition Systems Support for the Viking 1975 Mission to Mars”, JPL Publication 82-107, May 15, 1983.

[11] S. Brooks, T. Litwin, J. Biesiadecki, N. Abcouwer, T. Del Sesto, M. McHenry, S. Myint, P. Twu, D. Wai, “Testing Mars 2020 Flight Software and Hardware in the Surface System Development Environment”, IEEE Aerospace Conference, Big Sky, March, 2022.

[12] D. Coil. “Report on 2020 Spacecraft-Associated Genome Analysis (SAGAn) workshop at NASA JPL”, Microbiology of the Built Environment network. February 18, 2020.

[13] M. Maimone, A. Holloway, S. Swieso, “Load and patch: Improving hot patch capabilities in Curiosity’s flight software”, Flight Software Workshop, Laurel, MD, USA, 7 February 2022.

BIOGRAPHY

Mark Maimone *is a Robotic Systems Engineer in the Robotic Mobility group at the Jet Propulsion Laboratory. Mark designed and implemented the GESTALT self-driving surface navigation Flight Software for MER and MSL missions; during MSL operations served as Deputy Lead Rover Planner, Lead Mobility Rover Planner and Flight Software Lead; developed downlink automation tools for MER and MSL; and is now Mars 2020 Rover Deputy Lead Rover Planner and a member of the FSW development team. He holds a Ph.D. in Computer Science from Carnegie Mellon University.*

Nikunj Patel *is a Tactical Downlink Lead for MSL, which follows the responsibilities of a flight director on the mission. Furthermore, he is also the Mobility Subject Matter Expert and VTWD V&V team lead for Curiosity Rover. He received his Masters in Aerospace Engineering from University of Central Florida (UCF) where he led multiple cubesatellite development missions through AIAA and UCF. In addition to that, Nikunj designed the first Ground Support Biobarrier (GSB) that would be utilized by NASA’s Planetary Protection (PP) team to meet stringent bioburden requirements for future life detection missions to other terrestrial bodies.*

Anna Sabel *is the V&V lead for R13 and a member of MSL’s FSW and Mobility & Mechanisms teams. She received her Bachelor of Science degree from Brown University. Her previous work at JPL has included the development of multi-agent cooperative behaviors for a maritime robotics application. She is currently training to be an MSL Tactical Downlink Lead.*

Alexandra Holloway *received her Ph.D. in Computer Science at the University of California, Santa Cruz, specializing in human-centered design and storage systems. She now leads the MSL Flight Software team in building new capabilities for the Curiosity rover. Previously, Alexandra designed user interfaces for the operators of the Deep Space Network.*

Arturo Rankin *received his Ph.D. in Mechanical Engineering at the University of Florida in 1997. He is currently a Robotic Systems Engineer in the Robot Operations group at the Jet Propulsion Laboratory, and the M2020 Robot Operations deputy lead. Prior to that, he was the MSL Mobility/Mechanisms team lead and the MSL Flight Software team lead.*