One of the goals of the Mars Science Laboratory (MSL) mission is to collect powderized samples from the interior of rocks in order to deliver these samples to onboard science instruments. This paper describes the algorithms and software used to control the drill, which is the component of the sample collection and delivery system that directly interacts with rocks to create and acquire powderized samples from their interior. This is the first time that autonomous drilling of rocks has ever been performed on another planet. One of the most important components of the algorithm used for drilling is a force feedback control system used to regulate the force applied to the rock during drilling. This algorithm and all of the other algorithms and software used to enable the process of robustly, efficiently, and autonomously drilling into rocks with a priori unknown and widely varying properties are described in detail in this paper. Results are shown from drilling rocks using the drill software on testbed hardware on Earth as part of the software development process. Results are also shown from the first holes drilled with the flight vehicle on Mars, thus successfully demonstrating the first extraterrestrial autonomous drilling of a rock.

1. INTRODUCTION

The Mars Science Laboratory (MSL) Curiosity rover landed on Mars on August 5, 2012. A major component of the ~900 kg rover that was landed MSL Launch Press Kit (2011) (shown in Figure 1) is the sample acquisition/sample processing and handling (SA/SPaH) system, which consists of a five degree-of-freedom robotic arm with a turret end-effector (see Figure 2). The turret consists of a rotary percussive drill (shown in Figures 3, 4, and 5), a sample processing tool known as CHIMRA (Collection and Handling for In situ Martian Rock Analysis), a dust removal tool (DRT), and two contact science instruments: APXS (Alpha-Particle X-ray Spectrometer) and MAHLI (Mars Hand Lens Imager). A detailed description of the SA/SPaH subsystem can be found in Jandura (2010).

The drill is designed to create and acquire powderized samples from the interior of rocks up to a depth of 50 mm. This sample is then sieved, portioned, and delivered by the SA/SPaH system to onboard science instruments for analysis. This paper describes the algorithms and software that are used to control the drill hardware to produce a system that can robustly, efficiently, and autonomously drill into rocks with a priori unknown and widely varying properties. The algorithms that are described here apply to the sample acquisition process between the time that the arm preloads the drill stabilizers onto a rock and the time that the arm unloads the stabilizers from the rock with the powderized sample contained in the interior of the drill. There are significantly more processes involved in target selection, rover approach, arm motion, preloading of the drill onto the rock, sieving, portioning, and delivering the sample after it is collected that are beyond the scope of this paper. The explicit goal of this development effort was to enable the autonomous collection of powder from the interior of rocks on Mars from up to a depth of 50 mm.

The MSL drill is the first extraterrestrial autonomous drill to have the capability of drilling into rock. The first extraterrestrial autonomous drill was sent to the moon in 1970 on the Soviet Luna 16 mission. It autonomously collected 101 g of lunar soil sample from a depth of 35 cm and returned this sample to Earth (Vinogradov, 1971). Two subsequent Luna missions (20 and 24) used the same system to successfully return more samples from the moon (Johnson, 1979). In 1981, the Soviet Venera 13 and 14 missions each collected a single soil sample from the surface of Venus (Surkov et al., 1984). There have been no other autonomous drills since then that have functioned on another planet or moon other than the MSL drill. The Mars Exploration Rovers had a rock abrasion tool (RAT) that produced a hole 45 mm in diameter and up to 5 mm deep, but it did not collect any sample (Gorevan et al., 2003). The Beagle 2 Mars lander had a rock corer grinder (RCG) that was designed to collect cores of up to 10 mm depth (Richter et al., 2003).
Unfortunately, the Beagle 2 did not land successfully. The Philae lander on the Rosetta mission, which is slated to land on the comet 67P/Churyumov-Gerasimenko in November 2014, has a sampler, drill, and distribution (SD) subsystem onboard. The SD is designed to collect soil samples from various depths up to 230 mm and distribute them to various science instruments with a minimum mission plan to collect two samples from a single hole but with the capability of collecting samples from multiple holes (Finzi et al., 2007; Magnani et al., 1998).

There have been many drills designed and tested on Earth with the intentions of extraterrestrial operation. A good summary of these can be found in Chapter 6 of Bar-Cohen & Zacny (2009). The most relevant and recent extraterrestrial sample acquisition system is that developed for the proposed Mars Sample Return (MSR) mission: the Integrated Mars Sample Acquisition and Handling (IMSAH) (Backes et al., 2010, 2012). This effort has designed and demonstrated a technology readiness level (TRL) 4 end-to-end sample acquisition system capable of autonomously collecting and caching rock cores of up to 7 cm in length. Another example is the Exomars drill, which will be designed to drill into Martian soil up to 2 m deep (van Winnendael et al., 2005). Prototype models of this drill have been developed and tested in laboratory environments.

Section 2 contains a brief description of the drill hardware and the avionics used to drive the hardware. A more in-depth discussion of the drill hardware can be found in Okon (2010). Section 3 discusses the design of the software that controls the drill hardware and implements the behaviors and algorithms. Section 4 describes the low-level algorithms that are used by the basic and advanced behaviors described in the subsequent sections. Section 5 contains an in-depth discussion of the basic behaviors designed to perform simple tasks in nominal conditions and to provide flexibility in off-nominal conditions. Section 6 contains an in-depth discussion of the advanced behaviors designed to perform the sample acquisition process. Section 7 discusses the venues that were used to test the drill software and algorithms. Section 8 shows the drilling results from the development and test program and from the first holes drilled with the flight vehicle on Mars, thus demonstrating the achievement of the stated goal of enabling the autonomous collection of powder from the interior of rocks on Mars from up to a depth of 50 mm. And finally, Sections 9 and 10 discuss the lessons learned and conclusions that were drawn from this work.
Figure 3. MSL drill (several stitched Mastcam images) (NASA/JPL-Caltech/Malin Space Science Systems).

Figure 4. External elements of the MSL drill (some elements removed for clarity).

Figure 5. Internal elements of the MSL drill (some elements removed for clarity).
2. HARDWARE AND AVIONICS

This section briefly describes the drill hardware and avionics. Figures 3–5 show different views of the drill hardware. The drill has four actuators that provide the following motions: bit (spindle) rotation, linear feed of the bit, percussion via a voice coil, and chucking. For the remainder of this paper, these four actuators will be referred to as spindle, feed, voice coil, and chuck, respectively. Three of the actuators (spindle, feed, and chuck) consist of a brushless motor and a planetary gear train. The fourth actuator is a voice coil that consists of an electromagnetic coil and a permanent magnet that powers a spring-mounted hammer. Figure 6 is a section view of the voice coil that illustrates the interaction of the various elements of the mechanism. The feed and chuck motors also have brakes that prevent motion of the motor when power is not applied. The spindle actuator rotates the fluted bit inside a powder collection tube. The two main functions of this actuation are to distribute the percussive impacts from the voice coil actuator across the entire surface area of the bottom of the hole [due to the flattened bit geometry (see Figures 4 and 5), each impact occurs along a single diametrical line] and to extract the powder from the bottom of the hole with the flutes. The nominal rotation speed used for drilling operations is ∼100 rpm. The feed actuator extends and retracts the bit with a ball screw at a maximum rate of ∼1 mm/s (but is nominally limited to a maximum rate of 0.25 mm/s during drilling operations). The voice coil actuator drives the hammer into a stationary anvil that is held in contact with the back of the bit. The anvil transfers the impact energy from the hammer into the rock, resulting in rock fracture and a powdered sample. The voice coil is capable of a range of commandable impact energies from 0.05 to 0.8 J. The chuck actuator drives a ball-lock mechanism that is used for releasing an old bit and engaging a new bit.

The drill also has several sensors, some of which provide critical feedback during the drilling process. A force sensor, consisting of redundant strain gages, measures the force in a single axis along the same axis of motion as the feed actuator. Sections 4.2 and 4.3 describe in detail how the force sensor signal is processed and used for drilling. An encoder on each of the three motors measures the rotational position. Current sensors measure the current of each of the three motors and the voice coil. Two redundant contact switches sense when the drill stabilizers are in contact with a rock. Reed switches are used to monitor the voice coil position (but are not used for closed-loop control of the voice coil). Fifteen platinum resistance thermometers (PRTs) located throughout the drill are used for thermal fault protection and as part of the voice coil thermal model described in Section 5.2.

There were three identical copies of the drill hardware fabricated and assembled: the flight model (FM), the engineering model (EM), and the qualification model (QM). The FM was integrated with the flight vehicle now on Mars and saw limited functional testing (testing of the FM drill was limited to testing in air/free space or against ground test fixtures for cleanliness reasons). The EM was integrated onto several testbeds and was the main hardware used for testing of the drill flight software. The QM was operated with a variant of the drill flight software and was used to evaluate the performance of the hardware in a wide variety of environmental conditions and with a wide variety of rock types. The test venues are discussed in more detail in Section 7.

The two main components of the avionics that are relevant to the operation of the drill are the motor control assembly (MCA) and the rover compute element (RCE). The MCA has a Sparc processor board that runs low-level software that is dedicated to motor control and sensing. It has independent boards dedicated to conditioning sensor signals, driving the motors, driving the brakes, and conditioning the power into the MCA. The RCE has a 200 MHz RAD750 processor that runs the VxWorks real-time operating system.

3. SOFTWARE DESIGN

The VxWorks operating system running on the RCE manages over 100 real-time tasks, of which the drill software described in this paper is one. The overall flight software (FSW) architecture used by the rover was based heavily on the Mars Exploration Rover (MER) software architecture (Reeves, 2005).

3.1. Commands

The rover is commanded by operators on the ground using command sequences that instruct the rover what to do for an entire day on Mars (a sol). Once a command sequence is received by the rover, the onboard software sequentially dispatches each command to the appropriate software module. Figure 7 shows the architecture for the
subset of software modules that controls the surface sampling and science (SSS) behaviors. This architecture enables the functionality of all of the SA/SPaH hardware (including the drill) and a small subset of the sampling science instrument functionality (Chemin and SAM) required for sample drop off.

Commands that are specific to SA/SPaH software modules (including drill) are sent from the command module to the sample processing and acquisition manager (SPAM) module. This module then performs functions that are common to all SA/SPaH modules, such as checking activity constraints, requesting resources needed by the current command, and making requests to initialize necessary hardware such as the MCA and the inertial measurement unit (IMU). Once the command passes all of the SPAM checks, it is then sent to the drill module. A typical sequence diagram for a single drill motion command is shown in Figure 8. Once the command reaches the drill module, the drill hierarchical state machine (HSM), shown in Figure 9, is used to process the command (Samek, 2002).

Before transitioning to the EXECUTING_CMD state, the drill module performs several functions. It determines whether the command requires actuation, and it initializes data structures based on command arguments, relevant parameters, and the state of the rover. For commands that require actuation or sensing, it then registers a callback with the software module that controls the motors (MOT). This callback is then called at 64 Hz, and this is where all of the real-time functionality of the command exists, as described in Section 3.3.
The drill module has a total of 35 commands that control the entire range of functionality of the drill. Some of these commands are used only for testing or off-nominal scenarios. The nominal drilling sequence uses 22 of these commands. Each of these commands has a set of arguments that is designed to handle the range of operational scenarios that is expected might change on a sol-to-sol basis. The 35 commands can be classified into two high-level categories based on whether the command requires actuation or sensing. If a command requires actuation or sensing, it is a motion command; if it does not, it is a nonmotion command. Within the motion category are subcategories that indicate the complexity of the command: basic behaviors (described in Section 5) and advanced behaviors (described in Section 6).

### 3.2. Parameters

Parameters that are settable by command are an additional software mechanism that greatly increases the robustness of FSW to future unknowns. Parameters are intended to be changed less often than command arguments. The parameter system was general-purpose software designed so that any FSW module could use it. The drill FSW module made more extensive use of the parameter system than any other FSW module. The drill FSW has over 1400 individual parameters that allow operational control over virtually every aspect of the drill FSW behaviors. While a detailed discussion of all of these parameters is beyond the scope of this paper, they can be categorized into roughly eight groups: force sensor, voice coil, filter, monitor, fault protection, telemetry control, basic behaviors, and advanced behaviors. A few examples are given here that indicate the breadth and depth of these parameters. There are parameters that specify the voice coil impact energy for each of the waveform levels (see Section 5.2). There are parameters that allow arbitrary FIR filters to be used for various filtering needs (see Section 4.1). And there are parameters that represent a look-up table for the disturbance forces described in Section 4.2. When discussing “nominal” scenarios in the descriptions of the rest of the paper, this usually refers to the use of a default set of parameters, and “off-nominal” scenarios are handled by setting these parameters to nondefault values as deemed necessary by ground operations. In addition to providing operational flexibility, these parameters were important to enable FSW development in parallel with testing of drilling
algorithms, and they reduced the risk of needing additional design iterations.

3.3. Real-Time Design and Implementation

There are 26 motion commands and 9 nonmotion commands. The 9 nonmotion commands do not have any real-time aspects and simply perform the function of the command and reply with the result. The 26 motion commands perform real-time execution of the behavior at 64 Hz within the context of the MOT module during the EXECUTING_CMD state. It is within this context that all motion is monitored and controlled. There are 15 separate callbacks that perform the real-time functionality of the motion commands. Each of these callbacks is controlled by at least one unique state machine. There are several common state machines shared by many of these callbacks (for example, all commands that use the voice coil will use the voice coil state machine shown in Figure 15). Sections 5 and 6 discuss the basic and advanced behaviors that result from the most significant of these state machines.

3.4. Fault Detection and Recovery

There are 40 separate drill FSW faults. They can be classified into two classes: goal error faults and motion error faults. Goal error faults simply indicate that the goal of the command was not achieved. Motion error faults are more serious, and indicate that there may be a hardware safety concern. These 40 faults cover all possible drill anomalies that can occur during operation. Examples of various categories of these faults include motor faults (overcurrent, stall, soft stop violation, etc.), timing faults (timeout, time of day, etc.), thermal faults, and excessive weight-on-bit (WOB) faults. All fault responses include stopping any current motion, dumping anomaly data products, and setting activity constraint manager (ACM) error flags. In most cases, these error flags will prevent any additional commands in the sequence from executing (they will fail with an ACM fault), and the vehicle will wait for ground operator intervention. In some situations, it is less than ideal to respond to a fault by simply stopping and waiting for ground operators. A ground response will not occur until the next day at the earliest. It is undesirable to leave the drill bit in a hole over a diurnal cycle as the thermal expansion and contraction of the rover can lead to substantial shear forces on the drill bit. In this case, an autonomous fault response may be sequenced that attempts to immediately retract the bit in the event of a fault. The distinction between goal and motion error faults becomes important for this type of activity. In general, goal error faults may be safely ignored when executing recovery actions. Motion error faults, on the other hand, are usually not ignored, as continued mechanism actuation may exacerbate the problem that led to the original fault.

4. Low-Level Algorithms

The low-level algorithms described in this section contribute functionality that is used in many parts of the basic and advanced behaviors described in Sections 5 and 6.

4.1. Signal Filtering

Finite impulse response (FIR) filters are used in many places during drill operation. They are used to filter noisy signals so that control loops can be closed or decisions can be made on the filtered signals. Table I shows a list of filter applications and the specific filter design that is used for each application. FIR1 is a 10th order filter with a 10 Hz cutoff frequency. FIR2 is a 20th order filter with a 1 Hz cutoff frequency. Both filters were designed with the window method using a Hamming window (Oppenheim and Schafer, 1999) and both operate at sample rates of 64 Hz.

The most important use of these filters is the force signal filtering done during the drilling process (see Figure 14). This use is important for two reasons:

1) There is a very large amount of process noise caused by the percussive impacts against the rock that are reacted through the force sensor.
2) Having a clean force sensor measurement enables the force control loop to be closed, which is fundamental to enabling drilling through rocks with \textit{a priori} unknown properties.

Figures 10 and 11 show the impulse, step, and frequency responses for the FIR1 and FIR2 filters. Figure 12 shows an example of raw force sensor data during drilling and the resulting filtered data. The voice coil impacts are at a constant frequency of $\sim 30$ Hz and the low-pass filter

\begin{table}[h]
\centering
\caption{Filter applications.}
\begin{tabular}{lll}
\hline
Purpose & Signal & Filter Design \\
\hline
determining when actuators have stopped & motor rates & FIR1 \\
force feedback for force application & force sensors and reference voltages & FIR1 \\
force feedback for drilling & force sensors and reference voltages & FIR2 \\
voice coil current limiting & voice coil current & FIR1 \\
spindle current feedback & spindle current & FIR1 \\
\hline
\end{tabular}
\end{table}
is able to reject this process noise so that the force sensor measurement can be used to robustly close the loop while drilling (see Section 4.3.2 for a description of the drilling force control algorithm). It was determined experimentally that the FIR2 filter was necessary during drilling because it significantly reduced the noise in the signal and improved the performance of the controller relative to the FIR1 filter.

4.2. Force Sensor Signal Processing

This section describes the conversion of the raw force sensor signal into a measurement that can reliably be used to close the loop on the force between the bit and the rock, which is called weight-on-bit (WOB). The drill force sensor consists of two independent strain gages in Wheatstone bridges. A reference voltage (nominally 10 V) is applied to the Wheatstone bridge, and the voltage across each strain gage is measured at 1024 Hz by an A/D on the MCA after passing through a low-pass hardware filter. There is \( \sim 10 \) m of flexprint cabling between each force sensor strain gage and the MCA on the flight vehicle. The nominal output of the force sensor is \( <0.1 \) mV/N and this same flexprint cable also had to carry high-current, switched motor driver wires, so much care had to be taken with respect to minimization of electromagnetic interference (EMI).

The drill software is provided with two 64 Hz signals (one for each strain gage) that are averages of the 16 samples taken at 1024 Hz by the MCA. Each of the force sensor signals goes through the conversion pipeline described below in parallel and they are combined (by averaging) at the end in order to produce a single WOB estimate.

The first step is to filter the force sensor outputs and the voltage reference signals (as described in Section 4.1). In retrospect, a better design would have been to wait until after the step in Eq. (2) to do the filtering, which would have helped reject high-frequency common mode noise. Equation (1) is then applied to the voltage reference signal to compensate for the voltage drop across the length of the flexprint cable:

\[
V_{\text{ref}} = V_{\text{measfilt}} \frac{R_{fs}}{R_{fs} + R_{\text{harn}}},
\]

where \( V_{\text{ref}} \) is the effective reference voltage, \( V_{\text{measfilt}} \) is the filtered reference voltage, \( R_{fs} \) is the known resistance of the force sensor strain gage, and \( R_{\text{harn}} \) is the known resistance of the force sensor flexprint cable harness. This calculation had to be done because of the non-negligible effects of the voltage drop due to \( R_{\text{harn}} \) because of the \( \sim 10 \) m flexprint cable.
Range checking is done to check for out-of-range values for $V_{\text{ref}}$, and then Eq. (2) is applied to normalize the force sensor signal with the reference voltage:

$$V_{fs} = \frac{V_{\text{fsfilt}}}{V_{\text{ref}}},$$

(2)

where $V_{\text{fsfilt}}$ is the force sensor voltage provided by the MCA that has been processed by the FIR filter described in Section 4.1. The drill force sensor output, $V_{fs}$, has a temperature-dependent bias. A lookup table that uses the current force sensor temperature (as measured by a PRT that is located close to the strain gage) as input is then used to correct for the thermal offset in $V_{fs}$. This thermally compensated force sensor signal is then used as an input to another lookup table that removes any force sensor nonlinearities and converts the signal to a force (in Newtons). The drill force sensor output, $V_{fs}$, has a temperature-dependent bias. A lookup table that uses the current force sensor temperature as input is then used to correct for the thermal offset in $V_{fs}$. This thermally compensated force sensor signal is then used as an input to another lookup table that removes any force sensor nonlinearities and converts the signal to a force (in Newtons). The next step in the pipeline is to compensate this force for several different sources of external disturbance forces, $F_{\text{dist}}$. These disturbances are caused by three separate physical mechanisms that are all functions of the linear feed actuator position and so are combined into one lookup table that takes the feed position as input and outputs the disturbance force offset (as shown in Figure 13). The first mechanism is the spring force of metal bellows used to prevent dirt and debris from getting in between the stationary and extending parts of the drill. This spring force is linear across the range of motion of the feed actuator in between the hard stops. The second mechanism is a sinusoidal magnetic effect caused by interaction of the ball screw and the permanent magnet of the voice coil. The third mechanism is the effect of the feed actuator moving through the stiffness of each hard stop on both edges of the plot.

The final step of the WOB calculation is to remove the bias on the force sensor created by the effect of gravity on the mass that hangs from the force sensor, $F_{\text{grav}}$. This mass is 10 kg and the gravity bias is a function of the orientation of the drill with respect to gravity. This bias can be estimated in two ways:

1) Use the known orientation of the drill (as calculated from the IMU on the rover and the arm joint angles), the known gravity on Mars, and the known hanging mass of the drill.
2) Use the measured WOB estimate when the bit is known not to be touching anything.

The second method is used most often because it is more accurate. The final two force sensor processing steps described above are performed using

$$WOB_{\text{meas}} = F_{\text{meas}} - F_{\text{dist}} - F_{\text{grav}}.$$  (3)

By convention, a positive force indicates compression of the force sensor, which is what occurs during nominal drilling.

Once the disturbance force and the gravity bias are removed from the signal, this becomes the final WOB estimate for each of the strain gage signals. A check is then performed on the difference between the two WOB estimates to make sure that they are within certain acceptable bounds, and then the two values are averaged in order to come up with
Figure 14. Force control block diagram.

the final WOB estimate to be used by all of the force control algorithms described below.

4.3. Force Control Algorithms

There are two separate force control algorithms used at various points during the drilling process, both of which can be represented by the block diagram shown in Figure 14. The first is the apply WOB algorithm that is used to command a desired WOB, and is used by the surface seek (Section 6.1), hole start (Section 6.2), and hardness test (Section 6.3) behaviors. The second is the drilling force control algorithm, which is used by the drilling behavior (Section 6.4) to maintain the WOB while the drilling process is being performed. Both of these algorithms are described in detail below. These force control algorithms are the core of the first force feedback control system ever used on Mars.

4.3.1. Apply WOB

The apply WOB algorithm is a simple proportional force feedback controller that uses the filtered force sensor signal (WOB\text{meas}) (Sections 4.1 and 4.2) as feedback, a commanded force input (WOB\text{cmd}) as the reference signal, and the feed actuator velocity (V_{\text{feed}}) as the control signal:

\[ WOB_{\text{error}} = (WOB_{\text{cmd}} - WOB_{\text{meas}}), \]
\[ V_{\text{feed}} = K_p WOB_{\text{error}}. \]

Equation (5) is applied at 64 Hz by measuring the WOB and commanding the feed actuator velocity until the nominal exit condition

\[ WOB_{\text{error}} < WOB_{\text{eps}} \]

is met, or any of a number of fault conditions occurs (where WOB_{\text{eps}} is a small force, nominally 2 N). Safety checks are done in this inner loop to make sure that the WOB and the feed position are within acceptable bounds. Additionally, the control signal (V_{\text{feed}}) is limited to stay with the physical capabilities of the actuator. This algorithm is used by the seek surface (Section 6.1), hole start (Section 6.2), and hardness test (Section 6.3) advanced behaviors.

4.3.2. Force Control while Drilling

The force control algorithm used during drilling is a PI (proportional and integral) force feedback controller that uses a heavily filtered force sensor signal (WOB\text{meas}) (Sections 4.1 and 4.2) as feedback, a commanded force input (WOB\text{cmd}) as the reference signal, and the feed actuator velocity (V_{\text{feed}}) as the control signal. It uses the same error equation as the apply WOB algorithm [Eq. (4)], and calculates the control signal using

\[ V_{\text{feed}} = K_p WOB_{\text{error}} + K_i WOB_{\text{interr}}, \]

where WOB_{\text{interr}} is the integrated error over time:

\[ WOB_{\text{interr}}(i + 1) = WOB_{\text{error}}(i) \Delta t + WOB_{\text{interr}}(i). \]

An integrator wind-up limit is then applied that saturates the WOB_{\text{interr}} at a specified threshold. Equation (7) is applied at 64 Hz by measuring the WOB and commanding the feed actuator velocity. This algorithm is used by the drilling advanced behavior (Section 6.4).

4.3.3. Gain Tuning

The gains used by the apply WOB (K_p) and the force control while drilling algorithms (K_p and K_i) were tuned using standard iterative tuning methods in simulation (see Section 7.1) to create acceptable controller performance characteristics such as rise time, settle time, elimination of steady-state errors, and stability. This tuning was performed with the filters described in Section 4.1 to assure that the performance and stability characteristics of the control algorithms were accurately represented in simulation. It is interesting to note that the gains that were tuned in simulation were then used as-is in all hardware venues (EM, QM, and FM) and no additional tuning was necessary.

5. Basic Behaviors

This section describes basic behaviors that are used in nominal command sequences to perform simple tasks between the advanced behaviors described in Section 6. They can also be used in off-nominal conditions to provide flexibility for diagnostics or unforeseen required functionality.

5.1. Spindle, Feed, and Chuck Actuation

The spindle, feed, and chuck actuators all have motors that are driven by the MCA, just like the other 28 motors on the rover (not including payload instrument motors). Common software (MOT) is used by all modules that control the behavior of these motors. The drill software interacts with MOT to control these three drill actuators by command. They can be operated in position, velocity, or current control mode and can be commanded individually or simultaneously. Each command specifies the desired goal (position, velocity, current), a voltage limit, a current limit, and, depending on the mode, either a speed ratio (percentage of maximum speed) or a time limit. The nominal stop condition for each command is either goal position achieved (in position mode) or time limit expired (in velocity and current modes). When commanded simultaneously, the drill software synchronizes the motion of all of the commanded motors such that they all start and stop motion at the same time.

5.2. Voice Coil Actuation

There are three main functions of the voice coil behavior:

1) Robustly and accurately control the voice coil actuator to create a commandable impact energy at a constant frequency. This is used by several advanced behaviors to create impacts that fracture and powderize the rock: hardness test (Section 6.3), hole start (Section 6.2), and drilling (Section 6.4). This mode of operation is also used to create a dynamic environment in the drill bit assembly that makes it possible to move the powder acquired by the drill to the sample processing hardware.

2) Apply a range of constant voltages across the voice coil for a single impact with known energy. This is used by the hardness test (Section 6.3) advanced behavior.

3) Apply a range of constant voltages across the voice coil, which is used for two purposes: to measure the resulting currents for self-calibration of resistance (see Section 5.2.1) and to perform a health check of the mechanism by retracting it slowly and exciting the reed switches.

   See Figure 6 for a section view of the voice coil mechanism, and Okon (2010) for a more detailed functional description of the voice coil hardware. The voice coil actuation is the most complicated of the basic drill behaviors. This is mainly due to two facts:

1) There is only one voice coil onboard the rover and, therefore, nearly all of the software required to operate the voice coil is in the drill FSW module (as opposed to the other actuators, which rely heavily on MOT software to control the actuator).

2) Voice coil operation is inherently complicated because it is operated in an open-loop fashion [i.e., the position feedback sensors (reed switches) are not used for control] and, therefore, it must use several models to allow it to robustly operate over a wide range of conditions (most importantly, a wide range of temperatures due to the cold temperatures on Mars and the significant self-heating effects of the voice coil).

   Figure 15 shows the statechart for the operation of the voice coil. This state machine is enabled during any command that requires voice coil actuation. The WAIT_FOR_ZERO_PROFILE_LOADED and WAIT_FOR_ALL_MOTORS states are used to synchronize the start of percussion with the motion of other actuators. This is important for several reasons, the most significant being to avoid undue bit wear that would be caused by rotating the bit against the rock without percussion. The THERMAL_UPDATE state is entered when the temperature estimate of the voice coil has change by more than a specified amount. The calculations used for the thermal update are described later in this section. The other significant feature of the voice coil state machine is the autonomous cool down if the voice coil temperature exceeds a specified value. The advanced behaviors that use the voice coil state machine monitor for transitions into this state so that they also pause to wait for the voice coil to cool down.

   The MOT interface used by the drill software to drive the voice coil is a simple voltage waveform consisting of 17 discrete voltages that are applied to the voice coil by the MCA for intervals of 2 ms for each entry. The voice coil input specifications in the drill FSW are upstroke force, downstroke force, and impact velocity. So the drill FSW must convert these input specifications to the voltage waveform required by MOT. This conversion process can be separated into two parts: feed-forward model conversions and real-time sensor feedback updates.

   The feed-forward model conversions take the three waveform input parameters (upstroke/downstroke force and impact velocity) and convert them to a current waveform and a back EMF waveform using Eqs. (9)–(17).

   First, the maximum hammer displacement ($x_{\text{max}}$) is estimated:

   $x_{\text{max}} = n_{\text{down}} \cdot \text{dwell} \cdot v_{\text{impact}},$ \hspace{1cm} (9)

   where $n_{\text{down}}$ is the number of downstroke entries in the waveform, $\text{dwell}$ is the time spent at each entry, and $v_{\text{impact}}$ is the desired impact velocity.

   Then, the position ($\bar{x}_{\text{up}}$) and velocity ($\bar{v}_{\text{up}}$) profiles that describe the motion of the voice coil hammer during upstroke are calculated:

   $\bar{x}_{\text{up}} = -x_{\text{max}} \sin \left( \frac{\pi}{2} \frac{t}{n_{\text{up}}} \right) - x_{0}.$ \hspace{1cm} (10)

   $\bar{v}_{\text{up}} = v_{\text{impact}} C_{t} \cos \left( \frac{\pi}{2} \frac{t}{n_{\text{up}}} \right), \hspace{1cm} (11)
where $i$ is the index of the waveform, $x_0$ is coil position at impact, $v_{\text{impact}}$ is the desired impact velocity, $n_{\text{up}}$ is the number of upstroke entries in the waveform (nominally set to 10), and $C_r$ is an estimate of the coefficient of restitution between the hammer and the anvil.

Next, the position ($\vec{x}_{\text{down}}$) and velocity ($\vec{v}_{\text{down}}$) profiles that describe the motion of the voice coil hammer during downstroke are calculated:

$$
\vec{x}_{\text{down}} = -x_{\text{max}} \cos \left( \frac{\pi}{2} \left( i - n_{\text{up}} \right)/n_{\text{down}} \right) - x_0, 
\tag{12}
$$

$$
\vec{v}_{\text{down}} = v_{\text{impact}} - \sin \left( \frac{\pi}{2} \left( i - n_{\text{up}} \right)/n_{\text{down}} \right). 
\tag{13}
$$

$\vec{x}_{\text{up}}$, $\vec{x}_{\text{down}}$, $\vec{v}_{\text{up}}$, and $\vec{v}_{\text{down}}$ are then concatenated to form $\vec{x}_{\text{prof}}$ and $\vec{v}_{\text{prof}}$ respectively. The “up” profiles nominally have

**Figure 15.** Voice coil statechart.
10 elements and the “down” profiles have 7 elements, so the final concatenated profiles nominally have 17 elements. Figure 16 shows example $\vec{x}_{\text{prof}}$ and $\vec{v}_{\text{backemf}}$ profiles.

Then, a coefficient array $\vec{K}_f$ that empirically models the relationship between the hammer position and the force generation capability of the coil (Newtons provided per ampere drawn) is calculated:

$$\vec{K}_f = C_0 + C_1 \vec{x}_{\text{prof}} + C_2 \vec{x}_{\text{prof}}^2 + C_3 \vec{x}_{\text{prof}}^3 + C_4 \vec{x}_{\text{prof}}^4. \quad (14)$$

Next the upstroke and downstroke current waveform ($\vec{I}_{\text{up}}$ and $\vec{I}_{\text{down}}$) can be generated using

$$\vec{I}_{\text{up}} = F_{\text{up}} / \vec{K}_f, \quad (15)$$
$$\vec{I}_{\text{down}} = F_{\text{down}} / \vec{K}_f, \quad (16)$$

where $F_{\text{up}}$ and $F_{\text{down}}$ are, respectively, the desired upstroke and downstroke forces. $\vec{I}_{\text{up}}$ and $\vec{I}_{\text{down}}$ are then concatenated for the total current waveform $\vec{I}_{\text{prof}}$ of nominally 17 elements. The vector division notation in Eqs. (15) and (16) refers to element-by-element division.

The last step in the feed-forward model conversions is to calculate the back EMF waveform using

$$\vec{v}_{\text{backemf}} = \vec{K}_f \vec{v}_{\text{prof}}. \quad (17)$$

Figure 16 shows example $\vec{I}_{\text{prof}}$ and $\vec{v}_{\text{backemf}}$ waveforms.

The next part of the waveform generation, the real-time sensor feedback updates, uses an iterative two-node thermal model to estimate the temperature and resistance of the voice coil:

$$P_{\text{in}} = I_{\text{VC}}^2 R_{\text{VC}}, \quad (18)$$

where $P_{\text{in}}$ is the electrical power provided to the voice coil by the current, $I_{\text{VC}}$, $R_{\text{VC}}$ is the estimate for the voice coil resistance [which is estimated using Eq. (22)]. The thermal power lost to the environment, $P_{\text{ef}}$, is then calculated using

$$P_{\text{ef}} = T_{\text{VCpre}} - T_{\text{PRT}} / R_0, \quad (19)$$

where $T_{\text{VCpre}}$ is the previous iteration’s voice coil temperature estimate, $T_{\text{PRT}}$ is the current PRT measurement value, and $R_0$ is the thermal resistance of the voice coil mechanism. $\Delta T_{\text{VC}}$, the change in voice coil temperature over the last iteration, can then be calculated using

$$\Delta T_{\text{VC}} = P_{\text{in}} - P_{\text{ef}} \Delta t / C_{\text{VC}}. \quad (20)$$

Next, the voice coil resistance $R_{\text{VC}}$ is estimated using

$$R_{\text{VC}} = R_{\text{RT}}(1 + \alpha T_{\text{VC}}), \quad (22)$$

where $R_{\text{RT}}$ is the resistance of copper (the voice coil winding material) at 0 °C and $\alpha$ is the temperature coefficient of resistance of copper. Next, the round-trip resistance of the voice coil circuit, $R_{\text{RT}}$, can be calculated using

$$R_{\text{RT}} = R_{\text{VC}} + R_{\text{harn}}, \quad (23)$$

where $R_{\text{harn}}$ is the harness cabling resistance. Both $R_{\text{harn}}$ and $R_{\text{VC}}$ are updated periodically using the procedure described in Section 5.2.1. $R_{\text{harn}}$ uses this method because it is the only way to estimate that value (because there are no PRTs available to measure the temperature of the cabling) and $R_{\text{VC}}$ uses this method periodically during voice coil operation to correct for thermal model drift.

The voltage waveform that the MCA uses to apply voltage to the voice coil can then be calculated using

$$\vec{v}_{\text{prof}} = \vec{I}_{\text{prof}} \ast R_{\text{RT}} + \vec{v}_{\text{backemf}}. \quad (24)$$
Finally, the first and last entries of the voltage waveform, $V_{\text{prof}}$, are zeroed to allow the hammer to coast before and after the impact. This is a key aspect of the open-loops drive of the voice coil mechanism, which provides tolerance to impact timing variation and minimizes losses due to dynamic braking in the presence of process variations and uncertainties.

These calculations result in waveforms that can be commanded to have a wide range of impact energies and can operate robustly over a wide range of temperatures. The voice coil is parametrized to have six discrete “levels” each with increasing impact velocities and upstroke/downstroke forces. These levels can be commanded by the ground to fit the needs of the current behavior. As is described in Section 6.4.2, the drilling behavior has an additional algorithm that can autonomously determine the optimal level, which may change over the depth of the hole due to the nonhomogeneity of the rock.

### 5.2.1 Voice Coil Resistance Calibration

As mentioned in the previous section, $R_{\text{harn}}$ and $R_{\text{VC}}$ need to be updated periodically using a voice coil resistance calibration algorithm. The algorithm applies several incrementally increasing voltages across the coil and measures the current at each voltage. It then uses Ohm’s law to estimate the round-trip resistance $R_{\text{RT}}$. Then $R_{\text{harn}}$ or $R_{\text{VC}}$ is calculated using Eq. (23) using the known value for the other resistance. $R_{\text{harn}}$ is estimated before operating the voice coil when the coil temperature (and thus its resistance) is known from the PRT measurement. $R_{\text{VC}}$ is estimated periodically during the advance behaviors that use the voice coil, when $R_{\text{harn}}$ is known, in order to correct for thermal model drift.

### 5.3 Rover State Monitoring

Rover state monitoring provides the capability of monitoring the state of the rover simultaneously with any other drill behavior. This capability is used to ensure that the state of the rover is within safe bounds during drilling operations. For example, this behavior can be used to detect if the arm preload or the rover attitude changes by more than a safe amount while drilling.

There are 23 independent channels that can be monitored. Each of these channels falls into one of the following categories: drill force sensors, arm force sensors, arm resolvers, mobility resolvers, and rover tilt. Each channel has associated parameters for minimum value, maximum value, and persistence. The monitor behavior has three different modes: monitor, fault, and fault only when not percussing. Any channel that is in the monitor mode will be measured and collected, but will not cause a fault if the value is outside of the specified range. Any channel that is in the fault mode will cause a fault if the value is outside of the specified range for more than the specified persistence. Any channel that is in the fault only when not percussing mode will behave like the fault mode when the voice coil is not being actuated and like the monitor mode when the voice coil is being actuated. This model is useful for channels that are significantly affected by voice coil operation, such as the arm and drill force sensors and the arm resolvers. Additionally, there are two classes of channels: absolute and delta. The absolute channels use the current measurement of that sensor to do the range checking. The delta use the difference between the current measurement of that sensor and an initial measurement that is taken by command to do the range checking. As an example, the arm force sensor delta channels are useful for making sure that any motion of the rock, arm (due to slippage of the drill stabilizers on the rock), or rover (due to slippage of the wheels) does not cause excessive changes in preload force during the drilling process.

### 5.4 Resistance Calibration

Resistance calibration for the non-voice-coil actuators of the drill is performed by MOT, and thus the details of the algorithm are beyond the scope of this paper. It is used to estimate the resistance of the actuator harness and windings so that the current loop of the motor controller can accurately estimate the motor current, which is important for efficient operation and hardware safety.

Before every move command, if the temperature or time since the last resistance calibration has changed more than a specified amount, resistance calibration for the actuators required for the commanded motion is performed. Additionally, resistance calibration can be performed during the hole start and drilling advanced behaviors because these behaviors can take a long time to complete and thus can span wide operating temperatures.

### 6. ADVANCED BEHAVIORS

The advanced behaviors of the drill are the nominal commands used to autonomously collect a sample from the interior of rocks. These drilling behaviors are all performed between the time the arm preloads the drill stabilizers onto a rock and the time the arm unloads the contact points from the rock, after drilling has been performed, with the powdered sample contained in the interior of the drill.

The constraints on the initial state of the rover when these behaviors are started include the following:

1. Rover tilt is less than $20^\circ$.
2. The arm has preloaded the drill stabilizers against the surface of the rock with at least 300 N (which is greater than $2\times$ margin on the maximum WOB during all drilling behaviors).
3. The drill actuators have been preheated to acceptable temperatures for operation.
6.1. Surface Seek

Because the topology of the rock cannot be remotely measured with enough accuracy and resolution to safely start interacting with the rock (the stereo error in the workspace of the arm is on the order of ±1 cm), the surface of the rock must be located by the drill. Additionally, this precisely determined surface location is used for depth calculations as described below. This behavior is typically preceded by a commanded move of the feed actuator to a position of ~8 mm, which guarantees that the hole start behavior (see Section 6.2) has enough room for retraction. This feed move also measures the force sensor to ensure that contact with the rock is not made [which should never happen because of the large margin between remote topology knowledge and the distance along the drill bit axis between the stabilizer contact points and the bit at a feed position of 8 mm (see Figure 5)]. The surface seek uses the low-level apply WOB algorithm described in Section 4.3 to extend the feed actuator until the bit comes in contact with the surface of the rock and the WOB reaches the commanded value. At this point, the surface location for a particular bit rotation position is known very precisely (to within ±0.02 mm). The position uncertainty is related to the force uncertainty and the stiffness of the feed mechanism (because the spring stiffness is the linear mapping from force uncertainty to position uncertainty). This surface location is then used to define the depth, \( d \), for all of the subsequent behaviors using

\[
d = x_{\text{feed}} - x_{\text{surf}},
\]

where \( x_{\text{feed}} \) is the current feed position and \( x_{\text{surf}} \) is the surface location.

The bit is not rotating during this seek behavior in order to prevent any rough and uneven topology of the rock from putting undesirable side loads on the bit. The downside of not rotating during this behavior is that the surface location is only precisely known at a single bit rotation angle (the bit geometry is such that contact with the rock occurs along a diametrical line). Due to variations in the local topography, the surface location can vary significantly across the range of bit rotation angles. This variation is dealt with by the hole start behavior described in Section 6.2.

6.2. Hole Start

Again, because the topology of the rock cannot be known with good accuracy, it must be assumed that the surface height as a function of bit rotation angle can vary significantly. This, in combination with a bit that is not radially symmetric, contributes to the necessity of an algorithm designed specifically to start the hole. This behavior, although significantly slower than the drilling behavior described in Section 6.4, is necessary to ensure that the bit does not encounter significant side loading caused by asymmetric interaction forces with the rock. These initial loads could potentially cause the hole to be drilled in an off-axis direction, which would result in failure of the drilling process deeper in the hole due to ever increasing side loads.

The hole start behavior safely cuts away any surface topology variation, creates a smooth surface at the bottom of a shallow hole, and creates the beginning of the hole that acts as a guide to keep the bit on-axis so that the nominal drilling behavior (which is much more efficient at making progress through the rock) can start safely and effectively.

A statechart of this behavior is shown in Figure 17.

This behavior starts by percussing for a short period to cut away the rock at the current bit rotation angle. It then retracts a distance that is large enough to guarantee that a subsequent bit rotation is safe (by retracting beyond any possible local variation of the surface location). It then rotates the bit a small amount and enters the APPLYING_WOB state that will stop at either the desired WOB (using the algorithm described in Section 4.3.1) or the current highest surface location, whichever it reaches first. If it reaches the desired WOB first, then a new highest surface location has been found. The behavior then repeats the preceding steps for one-half of a revolution of the bit (which is when the entire hole surface has been cut away to the current highest surface location). Then a clear-out motion is done, which consists of rotating and percussing lightly for a short period of time to clear out any cuttings from the hole. The clear-out is done at the highest surface location of each cycle. The behavior then repeats all of the preceding steps until a depth has been reached that guarantees that all surface variation has been removed, the bottom of the hole is flat, and the hole is deep enough that the sides of the drill bit are constrained by the rock. This condition is necessary for the effective operation of the drilling behavior described in Section 6.4. Results of the hole start behavior from the testbed and Mars are shown in Section 8.

6.2.1. Rate of Penetration During Hole Start

The rate of penetration (ROP) (the change in hole depth over time) is measured during the hole start behavior by comparing subsequent highest surface locations from each cycle. Thresholds are specified for the minimum and maximum acceptable ROP during hole start to ensure that the rock is neither too hard nor too soft at any point during this behavior. Although, nominally, only the minimum acceptable ROP threshold is used.

6.3. Hardness Test

The hardness test algorithm consists of three techniques that are used to determine the hardness of the rock. It is typically performed after hole start. The first technique is to apply a high WOB on the rock and measure the depth of the resulting indentation using the surface seek technique described in Section 6.1. The second technique is a tap test.
that creates a single impact with the voice coil (as described in Section 5.2) and measures the depth of the indentation in the rock created by this impact. The third technique is to actuate the voice coil with a waveform for a brief period of time (without rotating the bit) and to measure the depth of the indentation in the rock created by this actuation. All three of these techniques are used to estimate the hardness of the rock. Results of the hardness test behavior from the testbed and Mars are shown in Section 8.

6.4. Drilling

After hole start has completed successfully, the drilling behavior is started. The core of this state machine is the DRILLING state, which employs the force control algorithm described in Section 4.3.2. This algorithm runs while simultaneously rotating the spindle at a constant rate and actuating the voice coil. As described in Section 2, the combination of rotation and percussion distributes the
percussive impacts over the entire area of the hole and moves the powder that is created up the flutes and into the interior of the drill. The force control algorithm maintains a constant WOB, which is required in order for the impacts from the bit to fracture the rock, and it provides the rate of penetration that is dictated by the mechanical properties of the rock at any specific depth.

The DRILLING state also monitors for faults (Section 3.4), operates the voice coil (Section 5.2), monitors for voice coil cool down periods (Section 5.2), monitors telemetry buffer usage, controls the duty cycle timing, monitors the rate of penetration (Section 6.4.1), performs resistance calibrations when necessary (Sections 5.4 and 5.2.1), performs voice coil level control (Section 6.4.2), and monitors for the stop conditions. A statechart of this behavior is shown in Figure 18.

This behavior is completed successfully if the desired depth is reached or one of a subset of more benign faults occurs after a minimum depth has been reached. After the successfully completion of this behavior, the WOB is reduced and the spindle is rotated for a specified time to ensure that all of the powder is transported up the bit flutes and into the drill. Then the retraction behavior (Section 6.5) is performed. Results of the drilling behavior from the testbed and Mars are shown in Section 8.

6.4.1. Rate of Penetration During Drilling
ROP is measured during the drilling behavior periodically while actively drilling (i.e., in the DRILLING state). The measurement period is a parameter, but is nominally set to 10 s. The ROP is determined by comparing the depth at the beginning of the period to the depth at the end of the period. If this measured ROP falls outside specified minimum and maximum thresholds for more than a specified number of measurements, then an ROP fault will occur. This protects against rocks that are too hard or too soft at any point during the drilling behavior. For example, if a rock is hard enough to cause the ROP to fall below the minimum threshold, then it is desirable to stop drilling to prevent potential excessive wear to the bit.

6.4.2. Voice Coil Level Control
Voice coil level control (VCLC) is a mode of operation of the drilling behavior that can autonomously determine the optimal voice coil level. The main motivation for this algorithm is to make the voice coil usage more efficient from both an energy and a mechanism life standpoint. When enabled, this algorithm has four main functions:

1) It uses a minimum ROP threshold to determine when to increase the level of the voice coil.
2) It uses a minimum WOB threshold to determine when to decrease the level of the voice coil.
3) It maintains a cost estimate to determine the effectiveness of the voice coil level at the current depth.
4) It periodically increases and decreases the voice coil level to search for the optimal voice coil level.
The first function, using a minimum ROP threshold to increase the level, ensures that a minimum ROP is being met, which is important for drill life and operational efficiency. If the minimum ROP is not being met and the voice coil level has been increased to its maximum, then a fault occurs, and the drilling operation is aborted.

The second function, using a minimum WOB threshold to decrease the level, takes advantage of a feed rate saturation effect in the force control algorithm. If the WOB cannot be maintained even at the maximum feed rate, then the voice coil level is decreased.

The cost estimate, the third function, is determined using

$$C(L) = W(L)/\text{ROP},$$

where $C(L)$ is the cost as a function of voice coil level, $L$, and $W(L)$ is a weight factor for each level. If $W(L)$ is set to be equal for all levels, which is the nominal setting, then the optimization will just be on ROP (i.e., drill as fast as possible within the maximum ROP threshold). The weight factors leave open the possibility to create cost functions based on other factors, for example, potential nonlinear voice coil mechanism life effects.

The main motivation for function four, the search function, is to avoid getting trapped in local minima caused by the nonhomogeneity of the rock. For example, if the VCLC behavior is not enabled, and a brief hard spot in the rock is encountered at the top of the hole, then it is possible that the voice coil level would unnecessarily stay at the highest level for the entire hole. Conversely, if a brief soft spot is encountered at the top of the hole, then it is possible that ROP would stay at an unnecessarily low rate for the entire hole. By periodically searching up and down a level, the algorithm can detect if either of these levels is more optimal than the current level for the mechanical properties of the rock at the current depth. This function is not nominally enabled, and the testbed and Mars results with this fourth function are disabled. Results of the VCLC behavior from the testbed and Mars are discussed in Section 8.

6.5. Retraction

After the drilling behavior has completed nominally (or any of the drilling behaviors fault after the bit has entered the rock), the retraction behavior safely and robustly removes the bit from the rock. This behavior is designed to be more robust to off-nominal conditions (as compared to other drill behaviors) because it is used in all of the fault responses during drilling. It is undesirable to leave the bit in the rock overnight due to thermal expansion and contraction of the rover and the arm, which are capable of providing enough displacement to cause large binding forces between the bit and the rock.

Nominal retraction is achieved by moving the feed actuator to a position that is a specified distance above the rock surface location while rotating the spindle and monitoring the force sensor. The spindle is actuated for two purposes: to prevent any powder still in the flutes of the bit (there should be very little due to the final spindle rotation after drilling) from being lost, and to reduce the risk that the bit will get caught on the side of the hole. The force is monitored to prevent excess force from being applied to the force sensor itself if the bit snags during retraction. The force threshold is set just below hardware safety levels so that the retraction is as robust as possible. It is also possible, but not nominal, to use the voice coil during retraction. It is not nominal in order to prevent as much sample loss from the bit as possible. If a nominal retraction fails, then the voice coil and/or higher feed actuator current limits may be used to increase the chances of success for subsequent retraction attempts.

The retraction behavior is the only behavior with the capability of gracefully degrading depending on what actuators are available. All other commands immediately fault if all expected actuators are not available. The retraction behavior will not fault in cases in which the spindle and/or voice coil actuators are not available. Instead of faulting, this behavior will proceed with the functionality that is provided by the available actuators. This is another technique used to increase robustness of the retraction behavior.

7. SIMULATION AND TESTBEDS

This section describes the simulation and testbed venues that were used for algorithm development, parameter tuning, debugging, and verification and validation (V&V) testing of the drill FSW.

7.1. Simulation

The drill simulation is part of a much larger simulation environment used by all of MSL FSW known as the Work Station Test Set (WSTS). WSTS simulates the FSW/hardware interactions. The general interaction models that are relevant to the drill FSW include simulation of the MCA hardware, the actuators, and the IMU. Simulation models that were developed specifically for the drill FSW include the force sensor, the voice coil, and the rock.

The force sensor model calculates the WOB, $F_{\text{WOB}}$, using

$$F_{\text{WOB}} = (x_{\text{feed}} - x_{\text{surf}})K_{\text{feed}},$$

where $x_{\text{feed}}$ is the feed position, $x_{\text{surf}}$ is the rock surface location, and $K_{\text{feed}}$ is the stiffness of the feed actuator. This WOB is then converted to a simulated MCA measurement by essentially inverting the force sensor processing pipeline discussed in Section 4.2 and adding Gaussian noise to the measurement.

The voice coil model is a simple resistance and back EMF model that estimates the voice coil current, $I_{\text{VC}}$, given the RMS voltage of the waveform provided to the MCA simulation.
The rock model then calculates the change in the surface location of the rock, $\Delta s$, over the simulation period, $\Delta t$, using

$$\Delta s = E_{\text{drill}}(F_{\text{WOB}}W_1 + I_{\text{VC}}W_2)\Delta t/S_{\text{rock}}(d),$$  \hspace{1cm} (28)$$

where $S_{\text{rock}}(d)$ is the strength profile of the rock, which can be varied by depth, $d$, in order to simulate nonhomogeneous rocks. $W_1$ and $W_2$ are weight factors that estimate the relative contributions of $F_{\text{WOB}}$ and $I_{\text{VC}}$, and $E_{\text{drill}}$ is an estimate of the drilling “effectiveness,” which is an experimentally determined value based on known rate of penetrations that the drill achieves in rocks of known strength. This equation essentially approximates the sensitivity of the rate of penetration through the rock to the commanded WOB and voice coil impact energy.

This simulation environment was used for a wide variety of testing, including algorithm development, parameter tuning, FSW debugging, and V&V testing. Thousands of drilling simulations were run during the drill FSW development for these purposes. The simulation was particularly useful for algorithm comparisons. Comparing algorithm performance in real rock is problematic because the precise rock properties are nonrepeatable between holes. It is impossible to exactly repeat an experiment to compare two different algorithms in real rock. In simulation, however, the rock properties are precisely repeatable.

### 7.2. Testbeds

The EM drill was used for nearly all of the hardware testing of the drill FSW. There was limited testing on the FM drill before launch, the main goal of which was to verify that the FM drill behaved like the EM drill and to characterize any differences. Over the project life-cycle, the EM drill was integrated into several different test venues. The most significant of these venues are the Payload System Testbed (PSTB) (Figure 19) and the Vehicle System Testbed (VSTB) (Figure 20). Both of these venues use flightlike avionics and are capable of running exact copies of the FSW as is run on the flight vehicle. The PSTB consists of the SA/SPaH sub-system mounted on a Stewart platform that enabled testing in the laboratory over a wide range of vehicle attitudes. The VSTB is a full rover with complete mobility and SA/SPaH subsystems integrated onto a flightlike vehicle that enabled testing in the laboratory sandbox or outdoors in the JPL Mars yard.

There were two other test venues whose main purpose was not to evaluate the drill FSW, but they did use the same algorithms and shared a significant amount of code with the drill FSW, and therefore had some role in helping to verify these algorithms. The testbeds Qualification Model Dirty Testbed (QMDT) and “baby Q” drilled many more holes than the PSTB and VSTB venues in a wide range of rocks and in Mars-like environments (temperatures and pressures). These test venues were critical to qualify and
characterize the hardware, to build an understanding of the sample acquisition and processing behavior, and to gain confidence in the robustness of the algorithms across relevant conditions.

8. RESULTS

The goal of this section is not to show comprehensive results from the entire drill test and V&V program and Mars operations. Instead, the goal is to show representative results from the development testing of the drill FSW and initial results from operations on Mars that demonstrate the functionality of the software and algorithms described in this paper. It is also important to note that no conclusions can be drawn from the results shown in this paper regarding relative drilling performance, because the drill arguments, parameters, algorithms, and software were all continuously being modified throughout this test program.

8.1. Testbed Results

The testbeds described in Section 7.2 have been key venues for debugging and demonstrating the functionality of the drill software. The testbed is also important for designing and validating the specific software settings that ultimately get used to drill on Mars. Approximately 40 holes have been drilled using the EM drill over the course of several years of testing. The rocks that were used ranged in strength from kaolinite and limestone (compressive strengths of 25–50 and 50–100 MPa, respectively) to saddleback basalt (compressive strength of >100 MPa). The subsections that follow describe the results of testing the major parts of the drilling behaviors in the testbed.

8.1.1. Hole Start Results from Testbed

Figure 21 shows the results for the hole start behavior for many of these holes. The exploration and tuning of hole start algorithm settings are apparent in this data. During
early testbed testing, the hole start algorithm was run with a higher WOB. This resulted in faster progress (as measured by depth achieved during each cycle) as illustrated by the leftmost grouping of results. The start hole WOB setting was later reduced to more gradually start the hole and thus minimize the risk of the bit walking about the surface of the rock. For a period of time, the entire hole start process was done with this low WOB setting. This resulted in dramatically slower hole start progress, as illustrated by the rightmost grouping of results in Figure 21. The best solution was a combination of the two approaches. The first two cycles of the hole start algorithm are run with a lower WOB and the remaining cycles are run with a higher WOB. This balances the need to create a good starter hole with the operational time constraints levied on the drilling operation.

This process of exploring the hole start WOB setting is just one example of the many lessons that have been learned by operating the drill and its algorithms in the testbed. Other algorithm settings have also been varied over time, including percuss time per cut, clear out time per cycle, percuss level, and target depth. The algorithm has also gone through some revisions. Negative depths occurred during the early cycles of some of the holes because of the method used to define the surface location (as described in Section 6.1). The surface seek behavior would happen to find a low point of the rock topology, and the first cycle of the hole start behavior (as described in Section 6.2) would find a highest surface location that was higher than the surface location found by the surface seek. The final version of the hole start algorithm has logic to reset the surface location based on the highest point found during the first cycle, so more recent test results do not have any instances of the first cycle of hole start making negative progress.

The latest settings of the hole start algorithm result in the hole start process typically taking four to six cycles to achieve the target depth. The hole start process cannot take more than about nine cycles given the low ROP threshold enforced by software. Testing in very hard rocks has demonstrated that the behavior successfully limits the number of hole start cycles when the rock is not being cut.

8.1.2. Hardness Test Results from Testbed

Figure 22 shows the results for the hardness test behavior for many of these holes. There is significant scatter in the data, but rough grouping by rock types can be observed. For example, the hardness test tends to make less progress in saddleback, while larger divots are created in limestone and kaolinite.

8.1.3. Drilling Results from Testbed

Figure 23 shows a sequence of images taken from the first hole drilled in a rock in the testbed with the drill FSW. Figure 24 shows the depth versus time results for the drilling behavior for many holes. Drilling progress varies more widely with rock type than hole start performance. In soft rocks, the drilling behavior can reach the desired depth with less than 5 min of drill on time, as illustrated by the leftmost data on the plot. The straightness of the depth versus time relationship for these faster holes occurs because the ROP is saturated at the maximum feed rate allowed during drilling. The flat segment at the bottom of each hole signifies
Figure 23. Image sequence of the first testbed hole.

Figure 24. Drilling depth results.
that the drill achieved maximum depth. The software algorithm detects this condition (low ROP) and declares drilling a success as long as sufficient depth has been achieved. The variability in the depth profiles achieved by execution of the drilling algorithm is a testament to its robustness and adaptability.

Figure 25 shows the ROP versus drilling time results for the drilling behavior for many holes. As can be seen, the average ROPs vary significantly between holes. For example, holes 19 and 20 (limestone and kaolinite, respectively) both stay at the maximum ROP (0.25 mm/s) for the entire hole, and hole 15 (saddleback basalt) has an average ROP of ∼0.02 mm/s.

8.1.4. VCLC Results from Testbed

Figure 26 shows the results for the VCLC behavior from a soft rock (kaolinite). As can be seen in the figure, the algorithm responds to the initial low WOB by decreasing the voice coil level. The voice coil level then stays at the lowest level for a short period and then the algorithm responds to the low ROP by increasing the voice coil level.

8.2. Mars Results

The first drill sampling operation on Mars occurred during February 2013 (see Figure 27). Figure 28 shows the mini hole (right) and full hole (center) drilled in the “John Klein” target on sols 180 and 182, respectively. The diameter of each of the holes is about 1.6 cm. The mini hole was drilled to a depth of 20.0 mm to generate drill tailings for inspection without collecting any sample. The full hole was drilled to a depth of 63.9 mm to collect a sample that was later delivered to the science instruments inside the rover. Each of these operations was a complete success. The performance of the drill during the mini hole and full hole operations on Mars was consistent with the performance of the drill during Earth testing.

8.2.1. Hole Start Results from Mars

Figure 29 shows the depth penetrated by each cycle of the hole start algorithm. The results of the hole start algorithm were similar for the mini hole and full hole, which is not surprising given that the holes were in the same rock. For both holes, the hole start algorithm required five cycles to exceed 4.5 mm of depth. This is consistent with the number of cycles typically required when similar algorithm settings were using during Earth testing (as shown in Figure 21).

8.2.2. Hardness Test Results from Mars

Figure 30 shows the divot depth created by the hardness test sequence for the mini hole and the full hole. The hardness test creates a divot at the bottom of the hole after the hole start algorithm and before the drilling algorithm. As
with the hole start results, the hardness test results of the mini hole and the full hole were similar. Both hardness tests showed that little or no divot was created by simply pushing the bit into the rock with 130 N. Divots nearly 0.7 mm deep were generated once bursts of percussion were applied. This divot depth is in agreement with the results of hardness tests done during Earth testing. Although it is difficult to map the hardness test results to an absolute hardness, comparison with Earth testing (shown in Figure 22) suggests that the rock is probably softer than saddleback basalt and harder than kaolinite.

8.2.3. Drilling Results from Mars

The depth and rate of penetration of the drilling algorithm over time are shown in Figures 31 and 32. The depth plot illustrates that the drill made steady progress toward its target depth in both the mini hole and the full hole. The mini hole reached its target depth of 20.0 mm in 1.3 min and the full hole reached its maximum possible depth of 63.9 mm in 6.7 min. The drilling algorithm detected that the drill achieved maximum possible depth in the full hole after 30 s of very low rate of penetration. This feature is also visible in the rate of penetration plot (Figure 32). The rates of penetration achieved in this rock (0.13, 0.2 mm/s) are higher...
Figure 29. Hole start results from Mars.

Figure 30. Hardness test results from Mars.
Figure 31. Drilling depth results from Mars.

Figure 32. Drilling rate of penetration results from Mars.
than many of the hard rocks drilled during Earth testing, but still not as fast as the maximum drilling speed (0.25 mm/s) in softer rocks. The voice coil level control algorithm kept the drilling percussion level at level 4 throughout the full drill hole. If the rock had been softer, then the percussion level would have decreased during drilling. These results suggest a hardness relative to Earth analog test rocks that is consistent with the results of the hardness test. As was the case with start hole and hardness test results, the rate of penetration during the mini hole and the full hole were very similar. Rover operators examined the rate of penetration during the mini hole to gain confidence that an adequate rate of penetration could be achieved in the target before committing to drilling a full hole. After the sample was collected from the first full-depth hole, it was transferred to the CHIMRA scoop where it could be observed by the Mastcams (see Figure 33). Following this, the sample was then delivered to the onboard science instruments.

9. LESSONS LEARNED

In addition to the lessons learned described in the body of this paper, some additional lessons learned during this software and algorithm development effort are described here. First is the invaluable benefit of having a high-fidelity simulation for algorithm development. Limited access to hardware and the time-consuming nature of hardware testing would have significantly inhibited the iterative approach necessary to evaluate complex algorithms early in the design process. And very often, when a problem was found in systems level testing, it ended up being in an area of the software that had limited simulation fidelity associated with it. Second, is the difficulty of parallel development of high-level software with low-level software. Changing interfaces and low-level bugs discovered during high-level algorithm testing slows development down and requires immense regression testing efforts. Even with excellent functional decomposition design, there inevitably will be more complicated interactions than are initially expected between modules. Similarly, parallel development of hardware and software creates immense complexity in the development process and could possibly be improved by minimizing the barriers between the hardware and software development teams. Third, there is a significant cost in developing flexible software (which is manifested in the 1400 parameters). This cost is borne most significantly by the verification and validation test program. And last, even with an excellent high-fidelity simulation, when hardware interacts with the environment, unexpected things happen that can only be discovered through a robust “test as you fly/fly as you test” development effort.

10. CONCLUSIONS

This paper describes in detail the design of the algorithms and the FSW to enable the robust, efficient, and autonomous collection of powderized samples from the interior of rocks with a priori unknown and widely varying properties. The MSL drill is the first extraterrestrial autonomous drill to have the capability to drill into rock. One of the most important components of the algorithms used for drilling is a force feedback control system used to regulate the force applied to the rock during drilling. This algorithm and the rest of the algorithms and software described in this paper enable all of the functionality of the MSL drill.

Testbed results performed in the laboratory are shown from a variety of rocks as part of the development effort. These results show the robustness of the algorithms and software to widely varying rock properties. Results are also shown from the first holes by the FM rover on Mars on sols 180 and 182 in February 2013. These results demonstrate the autonomous collection of a sample from the interior of rocks on Mars with a priori unknown properties, reaching a depth of 63.9 mm. This achievement was the primary goal of this development effort. Demonstrating robustness to rocks with widely varying properties on Mars will require many more holes to be drilled, which, with luck, will be demonstrated in the upcoming months and years of this mission.
ACKNOWLEDGMENTS

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. There were thousands of people who contributed to the Mars Science Laboratory project. We would like to thank the entire team for its incredible effort to enable a successful mission. We would especially like to thank the FSW team that developed all of the software that the drill FSW relied upon to function, and the SA/SPaH and systems engineering team members involved in the development, design, and testing of the drill hardware and software.

REFERENCES


