dVRK-Si:
The Next Generation da Vinci Research Kit

Keshuai Xu, Jie Ying Wu, Anton Deguet, and Peter Kazanzides

***Abstract*—The da Vinci Research Kit (dVRK) is an open-source and open-hardware surgical robotics research platform that provides complete access to all levels of control on decommissioned da Vinci Surgical Systems. The original dVRK, released to the research community in 2012 and now installed at more than 40 institutions worldwide, is based on the first-generation da Vinci system released in 2000. In this work, we present dVRK-Si, an update that extends support to decommissioned second/third generation da Vinci S/Si patient-side robots. We describe extensions to the system architecture that maintain compatibility and interoperability with the original dVRK system, thereby enabling a heterogeneous mix of dVRK and dVRK-Si arms in the same system. Additionally, we describe the following components specific to the dVRK-Si: (1) a new controller that features 10 channels of PWM motor drivers with digital current loops; (2) alternative (closed-source) firmwares for embedded devices in the robot arm that stream serialized sensor data using an open protocol; and (3) electronics to support the passive Setup Joints included in full systems. All electronics designs, software, and firmware (except as noted above) are provided open source to the community.**

I. Introduction

The da Vinci Surgical System [1] follows a telesurgery paradigm, where the surgeon sits at a console and controls multiple dexterous robotic instruments inside the patient through minimally invasive incisions. The clinical success of this system makes it attractive to medical robotics researchers because it is a well-understood design and provides a familiar platform for surgeons to evaluate research prototypes. However, the clinical da Vinci systems are proprietary systems that do not readily support research, other than through a limited "read-only" interface [2].

The need for an open research version of the da Vinci system motivated the development of the da Vinci Research Kit (dVRK) [3], an open-source mechatronics research platform that repurposes the first-generation da Vinci robots (also called da Vinci Standard or da Vinci Classic). The software is free and open-source. The electronics are open-source and fully-assembled controllers are available at cost through a company. Intuitive Surgical and the Intuitive Foundation have been donating the mechanical hardware through a proposal selection process. Since the dVRK's release in 2012, more than 40 institutions have installed the system and an active surgical robotics research community has formed, as evidenced by the 253 publications surveyed in 2021 [4].

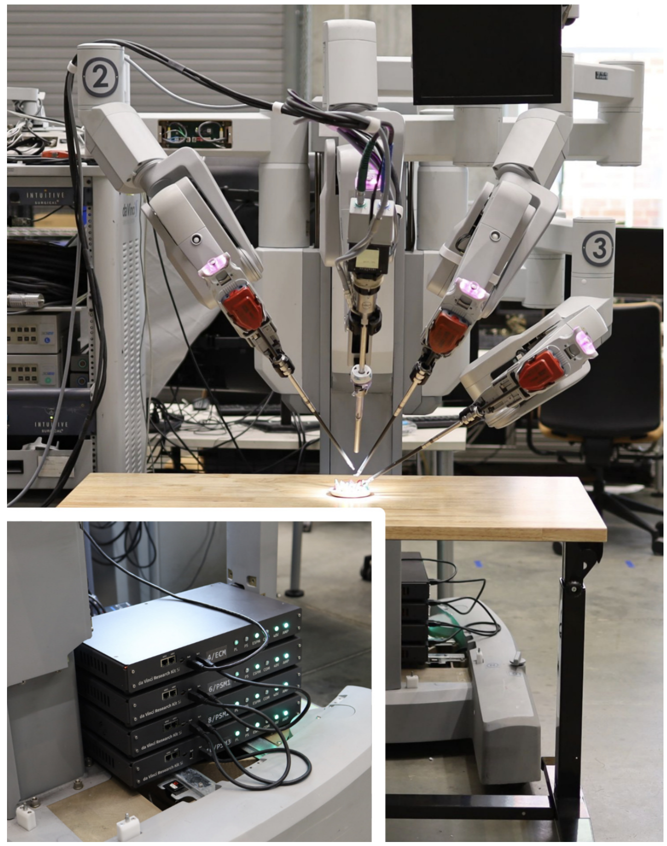

Fig. 1. A full dVRK-Si patient cart with SUJ, 3 PSMs, and 1 ECM. The original electronics in the patient cart have been replaced by 4 dVRK-Si controllers (bottom left inset).

Despite the substantial research enabled by the dVRK, it is based on the first-generation system whereas the fourth-generation da Vinci Xi is most prevalent in hospitals today and the fifth-generation da Vinci 5 has recently entered the market. Thus, younger surgeons have not had clinical experience with the first-generation system and it is also becoming more difficult for researchers to acquire the older mechanical hardware. This motivated the expansion of the dVRK to support parts of the second and third-generation da Vinci systems, known respectively as the da Vinci S and da Vinci Si. Specifically, we introduce the dVRK-Si (Figures 1

K. Xu and P. Kazanzides are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA. K. Xu, A. Deguet, and P. Kazanzides are with the Laboratory for Computational Sensing and Robotics (LCSR), Johns Hopkins University, Baltimore, MD 21218, USA. J.Y. Wu is with the Department of Computer Science and the Vanderbilt Institute for Surgery and Engineering, Vanderbilt University, Nashville, TN 37235, USA. P. Kazanzides can be reached at pkaz@jhu.edu

TABLE I
DA VINCI GENERATIONS AND ASSOCIATED CONTROLLERS

Generation/ Maintenance	**Arm**	**Controller**
Gen 1	MTM	dVRK
Ended 12/2012	PSM/ECM	dVRK
Gen 2 (S)	MTM	dVRK
Ended 12/2017	PSM/ECM	dVRK-Si (some)
Gen 3 (Si)	MTM	N/A
Ended 12/2024	PSM/ECM	dVRK-Si
Gen 4 (X/Xi)	MTM	N/A
Active	USM	N/A
Gen 5	MTM	N/A
Active	USM	N/A

and 2), which supports the Patient Side Manipulator (PSM) and Endoscopic Camera Manipulator (ECM) of the da Vinci Si (and some from the da Vinci S). Table I indicates which controllers support which generations of da Vinci systems, where dVRK indicates the original controller [3], dVRK-Si is the controller reported here, and N/A indicates that no open-source controller is available. Note that in the da Vinci X/Xi and 5, the Universal Surgical Manipulator (USM) can hold either an instrument or an endoscope.

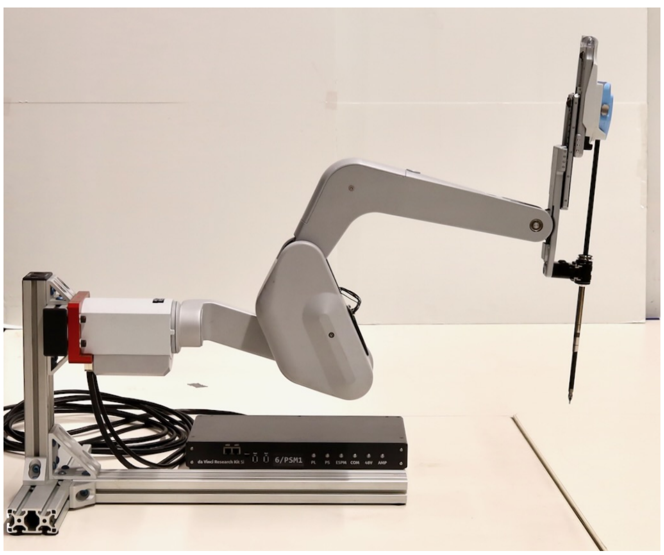

Fig. 2. A portable Si PSM mounted on an aluminum extrusion frame with its dVRK-Si controller.

Our primary contribution is the release of a new open-source research platform to the surgical robotics community, and the goal of this paper is to present a concise description of this platform as a reference for researchers.

II. Background

The da Vinci Si PSM and ECM had substantial mechanical and electrical changes from their first-generation counterparts. Apart from the drastic change in appearance, the remote center of motion mechanism was redesigned with a steel belt, and the translational joint became telescopic.

Electrically, the da Vinci Si and first-generation PSMs have mostly the same sensors and actuators. They both have seven joints (kinematically 6 degrees of freedom + gripper), each driven by a brushed DC motor, and position sensed by an absolute position sensor and an encoder. The first-generation PSMs have analog potentiometers as absolute position sensors, whereas the Si PSMs have magnetic angular position sensors. For both generations, the absolute position sensors of the last four joints are installed on the proximal side of the cables adjacent to the motors.

However, the sensor connections differ. In the first-generation PSM/ECM, all motor wires and signals are directly connected to the controller via a 156-pin connector. The analog potentiometer lines are directly connected, and the raw encoder signals are differentially buffered. The only relevant embedded electronics is the DS2505 memory chip inside the instruments. In the da Vinci Si PSM/ECM, the robot arm connects to the rest of the system with two substantially smaller D-sub connectors. The motor wires are still directly exposed via an external connector, but all sensors, including the encoders and magnetic rotary position sensors, are serialized by the Embedded Serializer for Patient-side Manipulator (ESPM), an FPGA-based circuit board inside the PSM/ECM, and sent to the controller via a Low Voltage Differential Signaling (LVDS) connection with four differential pairs, two in each direction. Figure 3 shows the ESPM with sensors connected inside the PSM/ECM.

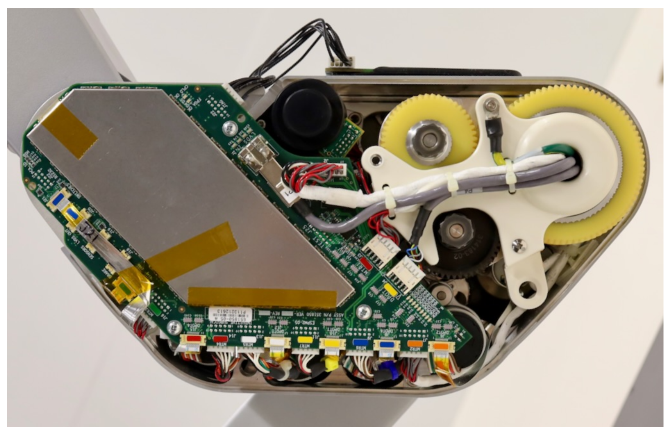

Fig. 3. ESPM (green PCB) in da Vinci Si PSM/ECM.

Both da Vinci PSM generations use brushed DC motors on all joints. However, the da Vinci Si PSMs feature a mechanical design less balanced by counterweights and therefore they require higher motor power on the first three joints. The da Vinci Si PSM motors are rated for higher nominal voltage (48V instead of 24V), but have low DC resistance, which makes them operate at far less than the nominal voltage under static conditions. For example, the Maxon motors in the first two joints have $1.13\,\Omega$ terminal resistance. When they hold position at 2.0 A, the voltage across the terminals is 2.3 V. This operating condition would thermally stress the linear amplifiers used in the first-generation dVRK controller because the linear amplifier would have to drop 45.7 V and dissipate more than 90 W of heat for each axis. It would also require a powerful AC/DC power supply. This prompted the redesign of the dVRK-Si motor drivers to adopt a pulse width modulation (PWM) architecture. Additionally, the first three joints on the da Vinci Si PSMs have electromechanical brakes. Therefore, a total of 10 channels is needed to drive all the motors and brakes, which would exceed the 8 channels available in the first-generation dVRK controller.

III. System Design

The primary dVRK-Si system design goals were:

1) Allow researchers to access and modify all levels of control, as with the original dVRK.
2) Interoperate with dVRK controllers, especially since first-generation MTMs will be used with Si PSM/ECM.
3) Maximize reuse of both electronics and software between the dVRK and dVRK-Si controllers.
4) Streamline the conversion process from the clinical da Vinci S or Si system to the open-source dVRK-Si research platform.
5) Support the passive Setup Joints (SUJ) from day one, rather than a few years later as with the original dVRK.

To achieve the second goal, we adopted the dVRK system architecture [5], where the PC and all controllers are connected via a fieldbus, using either IEEE 1394 (FireWire), or Ethernet, or a combination of the two (PC to first controller via Ethernet, then bridging to a Firewire subnetwork connecting all controllers [6]). We defined a new unified fieldbus protocol for the dVRK and dVRK-Si controllers. This allows researchers to reconfigure their systems to create arbitrary combinations of robot arms.

We were able to maximize both the electronics and code reuse between the two generations of controllers (third goal) by adopting the custom FPGA board originally developed for the dVRK. For the first-generation dVRK, this custom FPGA board was mated with a custom motor driver board, the Quad Linear Amplifier (QLA). Each dVRK controller contained two FPGA/QLA board sets to provide up to 8 axes of motor control. For the dVRK-Si controller, we designed a custom motor driver board (dRAC) to replace the QLA. In a parallel effort (outside the scope of this paper), the custom FPGA board was redesigned (FPGA V3) to replace the Xilinx Spartan 6 FPGA with a Xilinx Zynq System on Chip (SoC) and to add a second Ethernet port. Regardless of the FPGA board version or companion board type (QLA or dRAC), the interface to much of the functionality, including to write the motor current setpoint, read the motor current, and turn on and off the power, is consistent between dVRK and dVRK-Si, despite the differences in motor driver architecture and controls, to which the PC software is agnostic.

The fourth goal was challenged by the existence of embedded proprietary electronics inside the da Vinci S and Si robot arms, which had the potential to significantly increase the complexity of the conversion to the dVRK-Si, compared to the first generation system where all motor and sensor lines were directly available via the 156-pin connector. It was impractical to rewire the robot to bring all signals to larger connectors at the robot base, so we determined that it was necessary to reuse the existing LVDS serial connection. This led to two design alternatives: (1) removing the proprietary embedded ESPM and replacing it with custom open-source embedded electronics that reuses the LVDS connection, or (2) reprogramming the proprietary embedded ESPM with alternative firmware that supports an open protocol over the LVDS connection. We adopted the second approach, which was realized by one of the authors engaging in a summer internship at Intuitive Surgical to obtain access to the proprietary information required to implement the alternative firmware. Because this firmware relays the sensor data to the dVRK-Si controller with minimal processing, its closed-source nature does not hinder the user's full control and all other components of the dVRK-Si controller are open-source, thereby satisfying the first goal above.

The fifth goal (SUJ support) was added when it became clear that full da Vinci S or Si systems would soon become available for conversion to dVRK-Si.

IV. Hardware and Embedded Design

Figure 4 depicts a block diagram of the dVRK-Si system, with the dVRK-Si controller on the left and the PSM/ECM arm, with embedded electronics, on the right. Sections IV-A and IV-B respectively describe the dVRK-Si controller electronics and supporting firmware. This is followed by Sections IV-C and IV-D, which describe the alternative firmware for the ESPM embedded in the Si PSM/ECM, and the mechanism by which this firmware is programmed during system powerup. Finally, Section IV-E describes the additional electronics that enable each dVRK-Si controller to also handle the passive Setup Joint (SUJ) associated with that PSM or ECM.

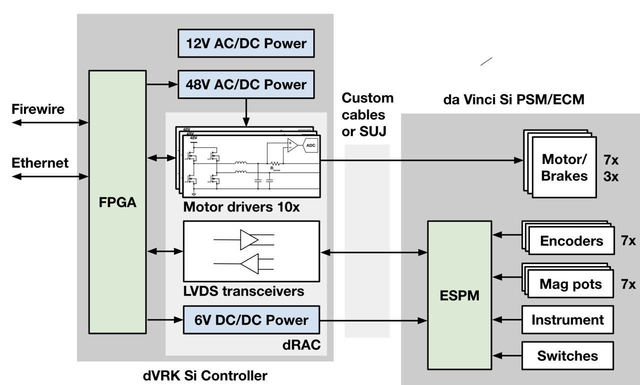

Fig. 4. Overview of the dVRK-Si controller. Mag pots = magnetic angular position sensors. The ECM has 4 joints instead of 7 in PSM, thus although the ESPM hardware remains the same, only 4 motors, 4 encoders, 4 magnetic position sensors, and 3 brakes are populated in the robot.

A. Controller Electronics

We designed the dVRK-Si controllers (Figure 5) in a 1.5U rack mount form factor. The controller can be rack mounted, stacked on a table or rack shelf, or stacked inside the da Vinci patient cart. The core of the dVRK-Si controller consists of the FPGA and dRAC boards. The FPGA board

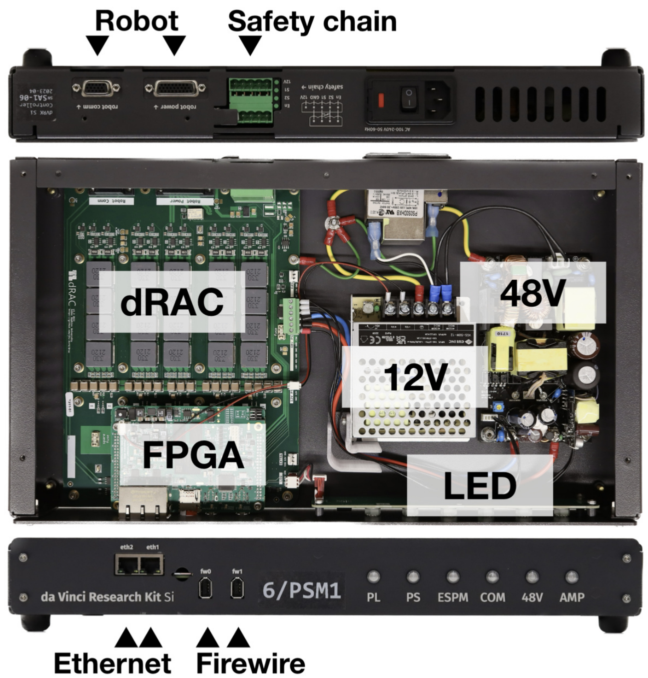

Fig. 5. dVRK-Si Controller. Rear, top internal, and front views.

handles Firewire and Ethernet communication. The dRAC board handles motor current control, power control, robot communication, and the safety chain. The name dRAC indicates the dVRK equivalent of the proprietary Remote Arm Controller (RAC) inside the da Vinci Si patient cart. Each RAC or dRAC drives one arm.

10 individual motor driver channels (Figure 6) handle bidirectional current control for 7 brush DC motors and 3 brakes. We drive the brakes the same way as the motors. Each channel has an H bridge that drives the load through an LC filter. A resistor between the LC filter and the motor senses the current. Its differential voltage is amplified by a bidirectional current sense amplifier before being digitized by an analog-to-digital converter (ADC). Identical circuits are used in all channels with minor variations in the current sense range, which is a tradeoff between current sense resolution and the maximum controllable output current. The first two joints have ± 6.827 A, and the others have ± 2.048 A because the former uses more powerful motors. The maximum PWM frequency is 500 kHz, limited by both the switching speed of the H bridge driver IC and the sample rate of the ADC.

A 6V DC/DC supply provides logic power to the ESPM and a set of LVDS transceivers converts the FPGA single-

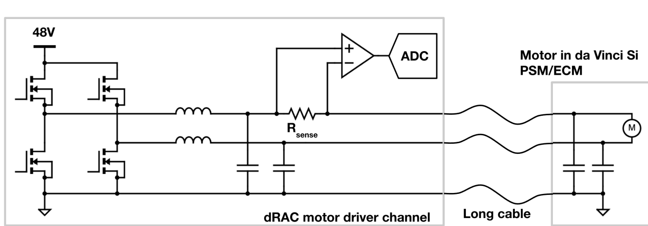

Fig. 6. A simplified schematic of a motor driver channel in dRAC.

ended signals to the differential LVDS signals that connect to the ESPM. To simplify the controller wiring, we included the safety chain circuit, identical to the first-generation dVRK, on the dRAC. It provides a hardware interlock that inhibits the 48V motor power supply.

B. Controller Firmware

The motor current is controlled by varying the duty cycle on the H bridge in response to the current measurement from the ADC. The PWM waveform is generated by thresholding an up-down counter shared by all channels. The center of both on-time and off-time are aligned between channels. We sample the current at the center of the off-time plus a 90-degree shift to account for the phase shift caused by the output LC filter. Sampling at the center of recirculation time captures the average current and avoids sampling during the noisy switching. The PWM waveform was set to 72 kHz with a 10-bit resolution in either direction. We control the current with a proportional-integral (PI) loop implemented in the FPGA. The PI loop runs every PWM cycle. Figure 7 shows the sequence and timing.

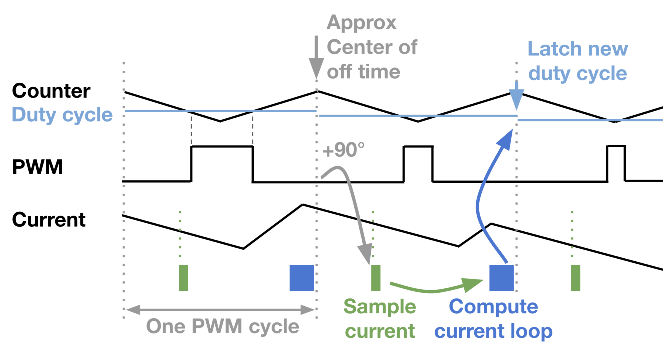

Fig. 7. Timing of PWM, current sampling, and current loop computation. Note that current waveform has 90-degree phase shift due to output filter.

The motor driver channel goes into fault shutdown when an overcurrent or overtemperature condition is detected by the H bridge IC, the current sense ADC saturates, or the actual current deviates from the setpoint. Interlocks force all motor drivers off when any of the following unsafe conditions are detected: (1) motor power supply voltage too low, (2) watchdog timeout, (3) ESPM communication error, (4) robot arm internal communication error, and (5) encoder not preloaded in ESPM (see next section).

The dVRK and dVRK-Si firmwares share the FireWire and Ethernet implementations, as well as an updated packet protocol for real-time data. The old protocol only supported 4 motors and 4 encoders per FPGA, which was insufficient for the dVRK-Si. The new protocol supports arbitrary numbers of motors and encoders.

C. ESPM Firmware

The ESPM is a module inside the robot arm that reads all sensors and communicates with the rest of the system via a full duplex LVDS link. Through collaboration with

Intuitive Surgical, we developed an alternative firmware for the dVRK-Si. The firmware reads encoders, magnetic angular position sensors, instrument memory, and various buttons and switches. It also controls the LED lights on the robot. Most sensor readings are transparently forwarded, except that we added hardware encoder velocity estimation [7]. The data are streamed to the controller via a custom open protocol at approximately 35 kHz. The same protocol is also used for transmitting data (i.e., LED control) from the controller to the ESPM.

Because the encoders are counted on the ESPM, power cycling the ESPM causes the counters to reset, which may create an erroneous large step in the encoder count. To avoid this error, the ESPM has an encoder preloaded flag that is cleared at startup. When the PC preloads the encoder, this flag is set. If the ESPM is reset, this flag gets cleared and subsequently triggers a safety interlock in the controller, as described in the previous section.

In the ECM, the encoders and magnetic angular position sensors for the last two joints are not connected to the ESPM. Instead, a separate module interfaces with them and serializes the data. The ESPM firmware automatically handles this difference. Due to the low update rate, hardware encoder velocity estimation is not available for these two joints.

D. ESPM Reprogramming

The ESPM normally loads the original firmware from its flash memory. We leave an ESPM Programmer (Figure 8) attached to the robot to program it with the alternative firmware every time it powers up. This device contains a microcontroller that connects to, and receives power from, the JTAG port on the ESPM. When the ESPM powers up, it powers the microcontroller, which reprograms the ESPM over JTAG. The microcontroller program is an XSVF player that plays back the ESPM firmware XSVF file from an on-board micro SD card (XSVF is a Xilinx binary representation of the standard Serial Vector Format, which can be used to program non-Xilinx devices such as the Altera FPGA in the ESPM). This approach simplifies the firmware update process because it only involves replacing a file on the micro SD card, which avoids the risk of an interrupted firmware update or a bad firmware image rendering the ESPM inoperable. In the future, we may instead use the ESPM Programmer to permanently update the ESPM firmware, in which case it would only be used once per device and would no longer need to be mounted on the robot.

E. Setup Joints and Mounting Options

The dVRK-Si offers versatile mounting options. The user may mount the PSM/ECM alone on a user-designed aluminum extrusion frame with an Intuitive Surgical-provided mounting rail (Figure 2), or on a full system, leave the robot mounted on the Setup Joints (SUJ), as shown in Figure 1. The former option directly connects the robot and the controller with two custom D-sub cables. The latter option requires a dVRK-Si SUJ Interface Board (dSIB-Si, Figure 9), a custom

Fig. 8. ESPM Programmer (green PCB) programs the ESPM

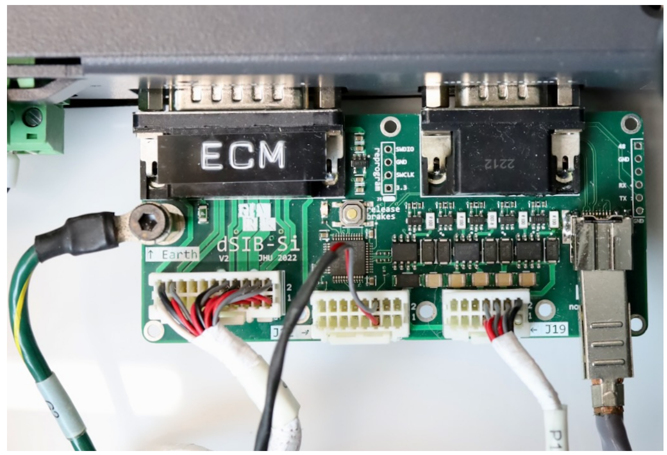

Fig. 9. The SUJ cables connect to the controller via dSIB-Si (green PCB)

circuit board, instead of the D-sub cables to connect to the internal wiring in the patient cart.

The SUJ is the structure on the da Vinci patient cart with passive arms that hold the PSMs and ECM near the patient. Each SUJ arm has one vertical translational joint and 3 or 4 rotational joints. A brake holds the position of each joint and two potentiometers redundantly measure its position. In the original (clinical) system, the RAC drives the brakes, and the Embedded Serializer for Setup-Joint (ESSJ), a module in the SUJ arm with an FPGA, ADCs, and LVDS transceivers, reads the potentiometers and inserts the data in the LVDS data stream. Another module at the bottom of the patient cart reads the potentiometers on the Z-axis joints. Figure 10 shows the locations of the major components.

The dSIB-Si replicates the internal connectors on the RAC and adapts them to the robot cable connectors on the dVRK-Si controller. It contains 5 brake drivers with current sensors. A microcontroller releases the brakes upon a digital signal from the dVRK-Si controller.

As an initial solution to read the SUJ joint positions, we replaced ESSJ with a custom circuit board (dESSJ) that passes through the LVDS communication and incorporates a microcontroller to read the SUJ potentiometer values and transmit them over Bluetooth Low Energy (BLE). Reliability and complexity concerns of BLE discovered in the field led us to develop a closed-source alternative firmware for the

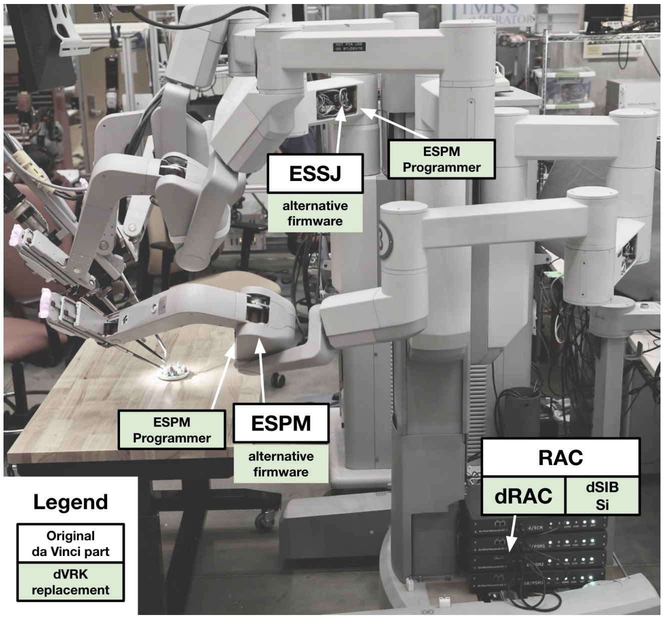

Fig. 10. Major electronic modules in da Vinci Si patient cart and their dVRK counterparts. The patient cart has been converted to dVRK-Si.

original ESSJ using an approach similar to the one used for the ESPM alternative firmware. The ESPM Programmer is reused to program ESSJ. The Z-axis joints will be read by an extension board (dSIB-Z-Si) connected to dSIB-Si. In this design, a fully converted da Vinci Si system with 3 PSMs and 1 ECM needs 4 dSIB-Si, 1 dSIB-Z-Si, 4 dVRK-Si controllers, and up to 8 ESPM Programmers (only 1 if the ESPM/ESSJ alternative firmware is permanently programmed).

V. Software

The majority of the differences between dVRK and dVRK-Si are encapsulated in the FPGA firmware level, thanks to the unified real-time packet protocol. Some dVRK-Si-specific features, such as reading the serial number and controlling the LED on PSM/ECM, are handled in the mechatronics-software low-level library to communicate with the motor driver boards in dVRK and dVRK-Si. The high-level software, e.g., kinematics, position control, teleoperation, and CRTK [8] compatible ROS 1 or ROS 2 interfaces, are reused with new configuration files and minimal code changes. The magnetic angular position sensor readings are mapped differently to account for nonlinearity and overflow, and we developed a calibration procedure that relies on the mechanical joint limits to find the zero positions.

VI. Results

The dVRK-Si system has undergone functional testing of hardware components and system-level testing with the full Si patient cart (Figure 1) connected to a dVRK MTM. For example, Figure 11 shows the step response of the digital current control for the first PSM joint. The rise time is less than 1 ms.

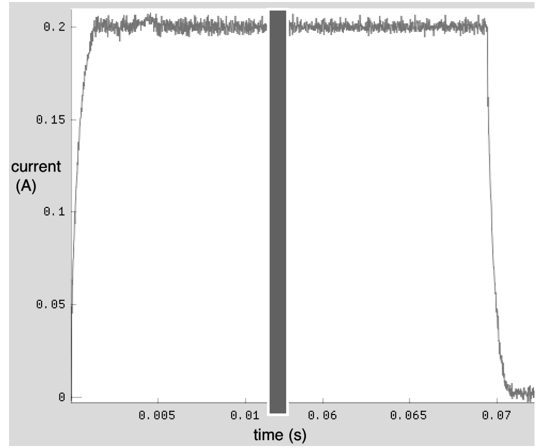

Fig. 11. Step response of digital current control on the first joint of PSM

The system-level testing indicated that it could support all FireWire IOs as well as PID control for all arms (MTMs, PSMs, ECM and SUJ) at the same default rate as the dVRK, i.e., 1.5 kHz (6.67 ms) with a standard deviation of 0.01 ms.

Two production runs of dVRK-Si controllers have been completed and delivered to the dVRK community.

VII. Conclusions

We presented a new development of the dVRK system to support the da Vinci Si PSM and ECM. We designed new controller electronics for dVRK-Si with and without SUJ: dRAC to drive the motors and read the sensors; dSIB-Si and dSIB-Z-Si to interface with SUJ; and ESPM Programmer to load modified firmware into the existing embedded devices inside the robot arms. The dRAC features PWM motor drivers with digital current loops. In addition, we created a closed-source open-protocol firmware for the ESPM and ESSJ to allow reading of the robot and SUJ sensors. The new system maintains compatibility and interoperability with previous-generation dVRK systems.

The design files and source code are available in the following repositories:

- dRAC: PCB [9], firmware [10], front panel PCB [11]
- dSIB-Si: PCB [12], firmware [13]
- ESPM Programmer: PCB [14], firmware [15]
- Low-level software (mechatronics-software): [16]
- dVRK software (sawIntuitiveResearchKit): [17]

Acknowledgments

We thank Intuitive Surgical and the Intuitive Foundation for their support and for the assistance from many employees, including Simon DiMaio, Dale Bergman, Alessandro Gozzi, Chris Nourse, Kollin Tierling, Kyle Ettinger, Vince Pearson, Ted Walker, and Gerry Labonville. In addition, Hao (Simon) Yang contributed to design discussions, initially at JHU and currently at Vanderbilt University. We acknowledge Russell H. Taylor and Greg Fischer for their continuous support and assistance with the dVRK effort, and we thank the dVRK community and the dVRK Consortium for their encouragement and support.

REFERENCES

[1] G. Guthart and J. Salisbury, "The Intuitive^{TM} telesurgery system: Overview and application," in *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2000, pp. 618–621.

[2] S. DiMaio and C. Hasser, "The da Vinci research interface," in *MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions*, Midas Journal: http://hdl.handle.net/10380/1464, July 2008.

[3] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci Surgical System," in *IEEE Intl. Conf. on Robotics and Auto. (ICRA)*, Hong Kong, China, 2014, pp. 6434–6439.

[4] C. D'Ettorre, A. Mariani, A. Stilli, F. R. y Baena, P. Valdastri, A. Deguet, P. Kazanzides, R. H. Taylor, G. S. Fischer, S. P. DiMaio, *et al.*, "Accelerating surgical robotics research: A review of 10 years with the da Vinci Research Kit," *IEEE Robotics & Automation Magazine*, vol. 28, no. 4, pp. 56–78, 2021.

[5] Z. Chen, A. Deguet, R. H. Taylor, and P. Kazanzides, "Software architecture of the da Vinci Research Kit," in *IEEE Intl. Conf. on Robotic Computing*, Taichung, Taiwan, April 2017.

[6] L. Qian, Z. Chen, and P. Kazanzides, "An Ethernet to FireWire bridge for real-time control of the da Vinci Research Kit (dVRK)," in *IEEE Conf. on Emerging Technologies and Factory Automation (ETFA)*, Luxembourg, Sep 2015, pp. 1–7.

[7] J. Y. Wu, Z. Chen, A. Deguet, and P. Kazanzides, "FPGA-based velocity estimation for control of robots with low-resolution encoders," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6384–6389.

[8] Y.-H. Su, A. Munawar, A. Deguet, A. Lewis, K. Lindgren, Y. Li, R. H. Taylor, G. S. Fischer, B. Hannaford, and P. Kazanzides, "Collaborative Robotics Toolkit (CRTK): Open software framework for surgical robotics research," in *IEEE Intl. Conf. on Robotic Computing*, Nov. 2020, pp. 48–55.

[9] "dRAC." [Online]. Available: https://github.com/jhu-dvrk/dRAC

[10] "mechatronics-firmware." [Online]. Available: https://github.com/jhu-cisst/mechatronics-firmware

[11] "dvrk-si-front-panel-led." [Online]. Available: https://github.com/jhu-dvrk/dvrk-si-front-panel-led

[12] "dSIB-Si-pcba." [Online]. Available: https://github.com/jhu-dvrk/dSIB-Si-pcba

[13] "dSIB-Si-firmware." [Online]. Available: https://github.com/jhu-dvrk/dSIB-Si-firmware

[14] "espm-programmer-pcba." [Online]. Available: https://github.com/jhu-dvrk/espm-programmer-pcba

[15] "espm-programmer-firmware." [Online]. Available: https://github.com/jhu-dvrk/espm-programmer-firmware

[16] "mechatronics-software." [Online]. Available: https://github.com/jhu-cisst/mechatronics-software

[17] "sawIntuitiveResearchKit." [Online]. Available: https://github.com/jhu-dvrk/sawIntuitiveResearchKit