Software Framework for Research in Semi-Autonomous Teleoperation

Co-PIs: Peter Kazanzides, Russell Taylor, Greg Fischer, Blake Hannaford

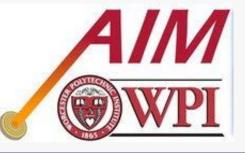
NRI 1637789

Johns Hopkins University



NRI 1637759

Worcester Polytechnic Institute



NRI 1637444

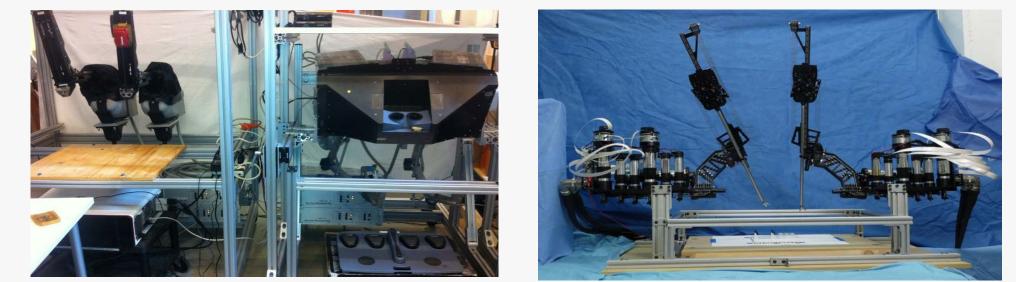
University of Washington

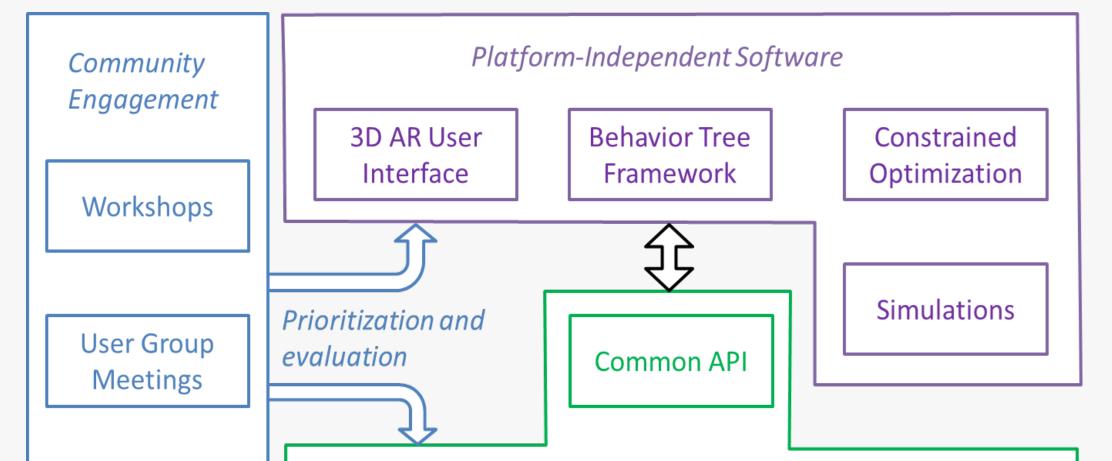


ELECTRICAL ENGINEERING UNIVERSITY of WASHINGTON

Project Overview

- **Community Engagement:** \bullet Organized a CRTK Tutorial at IROS 2018, Madrid. Organized a Workshop at ICRA 2018, Brisbane.
- Platform-specific Enhancements: \bullet Initial implementation of a ROS CRTK API to Raven II and dVRK.
- New Platform-independent Software: Prototyped vision pipelines and worked on robot simulations in Gazebo/rviz. Initial implementation of CRTK Python client API





da Vinci Research Kit (dVRK)

Raven II

Investigator Meetings	da Vinci Research Kit	Raven II	Other Systems		
	Platform-Specific Enhancements				

JHU Highlights

dVRK Software Release 1.6.0:

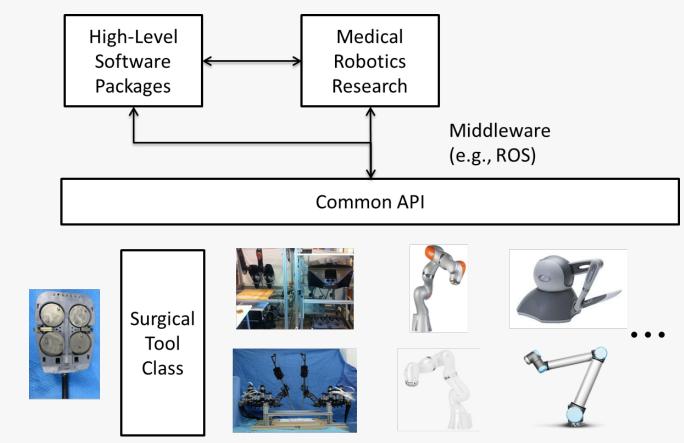
- **New Features**
 - **Experimental ROS CRTK interface**
 - Added ROS tf2 broadcaster
 - Camera manipulator (ECM) tele-operation
 - Software and hardware support for da Vinci:
 - Endoscope focus controller
 - Operator present head sensor
- Improvements and bug fixes
 - Improved velocity estimation on FPGA
 - Added audio feedback for console events
 - Better performance for ROS publishers/subscribers
 - Factorized and added new Qt Widgets

<u>Support for full da Vinci (including Setup Joints):</u>

- Fixed last issues with controller design (JHU)
- Finalizing orders for first batch, ~15 groups (WPI)



Common API (JHU, WPI, UW)



Guiding Principles:

- As simple as possible
- Guided by uses cases
- Logical and consistent naming conventions
- Short enough to type into interpreter
- Publish/subscribe (ROS topics) and client/server (ROS services)

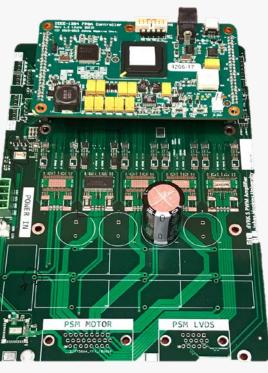
<u>dVRK-S/Si</u>





da Vinci S controller (2xQLA + FPGA + custom boards)

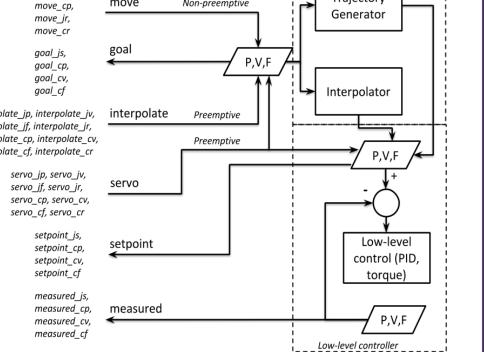




da Vinci S controller (PWM + FPGA)

- Initial prototype interface to da Vinci S/Si PSM
- Working prototype using 2 x QLA + FPGA, but gets hot •
- Designed new PWM based controller, needs testing

Control level	<pre>servo : direct real-time stream (pre-emptive) interpolate : interpolated stream (pre-emptive) move : plan trajectory to goal (pre-emptive), monitor with is_moving</pre>	move_cp, move_jr, move_cr goal_js, goal_cp,
Feedback	<pre>measured : sensor feedback measuredN : redundant sensor feedback (N=2, 3) setpoint : current setpoint to low-level controller goal : most recent interpolate or move goal</pre>	goal_cv, goal_cf interpolate_jp, interpolate_jv interpolate_jf, interpolate_jr, interpolate_cp, interpolate_c interpolate_cf, interpolate_cr
Space	j : joint c : cartesian	servo_jp, servo_jv, servo_jf, servo_jr, servo_cp, servo_cv, servo_cf, servo_cr
Туре	 p : position r : relative v : velocity or twist f : generalized force (effort and wrench) s : state for joint feedback (includes position, velocity and effort) 	setpoint_js, setpoint_cp, setpoint_cv, setpoint_cf measured_js, measured_cp, measured_cv, measured_cf



Results:

- Initial CRTK implementation for *dVRK*, *Novint Falcon* and *Sensable Omni*
- Python CRTK Client API prototype

Falcon00/Falcon00	base 4.465	00:03:25	<pre># create a new goal starting with current position 1.087 KHz 99.6 % 1088 samples 0.005 /2.043 ms 0.4 /221.4% 191 > period 01:26:12 Status #1: Falcon00: is right handed or symmetrical # create a new goal starting with current position start_cp.p = Self.measured_cp().p start_cp.M = self.measured_cp().M goal.p = Self.measured_cp().M goal.p = Self.measured_cp().M amplitude = 0.01 # 2 centimeters</pre>
Axis X Axis Y Axis Z Vector 5.531 Gripper	0		<pre></pre>
	Freeze		/force_dimension/Falcon00/RightSelf.move_cp(goal)
Grav	ty compensation		/force_dimension/Falcon00/Top self.is_moving_wait(20) /force_dimension/Falcon00/device_state /force_dimension/Falcon00/gripper/measured_js /force_dimension/Falcon00/is_moving /force_dimension/Falcon00/measured_cf /force_dimension/Falcon00/measured_cp

https://github.com/collaborative-robotics