The DARPA Phoenix Spacecraft Servicing Program: Overview and Plans for Risk Reduction

Carl Glen Henshaw

Naval Center for Space Technology
U.S. Naval Research Laboratory
Washington, DC 20375

Abstract

The Defense Advanced Research Projects Agency (DARPA) and the US Naval Research Laboratory (NRL) are developing robotic satellite servicing, repair, and assembly technologies with the goal of a demonstration mission at GEO. In preparation for such missions, NRL is conducting a campaign to develop and demonstrate technology for a variety of robotic satellite servicing tasks. This program, called Phoenix, plans to demonstrate the technologies needed to perform a wide variety of robotic satellite servicing tasks in the Naval Research Laboratory's Space Robotics Lab at TRL 7 by the end of 2015.

This paper describes the test campaign and the robotic components being developed for them, including the robotic arms, sensors, end effectors, and automation software.

1 Introduction

The Defense Advanced Research Projects Agency (DARPA) and the US Naval Research Laboratory (NRL) are developing robotic satellite servicing, repair, and assembly technologies with the goal of a demonstration mission at GEO. Generically, robotic satellite servicing tasks that may be of interest to satellite owner/operators include the repositioning of resident space objects (RSOs), e.g. for debris mitigation; repair of disabled satellites, such as repairing damaged thermal blanketing or freeing a stuck deployable [1]; upgrading existing satellites by, for instance installing an external Attitude Control System (ACS) package as was suggested for the Hubble Space Telescope robotic servicing mission; inspection, e.g. for micrometeorite or debris impact damage; and on–orbit assembly. In addition, DARPA is developing an innovative concept called Payload Orbital Delivery Systems or PODs that would allow spare parts, assemble–able components, and special–purpose end effectors to reach orbit rapidly and inexpensively using hosted payload compartments on commercial communications satellites [16]. This system would be used to supply a robotic satellite servicer with components to be used for a variety of satellite servicing operations and would greatly increase the capability and flexibility of such a system.

In preparation for such missions, NRL is conducting a campaign to develop and demonstrate technology for a variety of robotic satellite servicing tasks. This program, called Phoenix, plans to demonstrate the technologies needed to perform a wide variety of robotic satellite servicing tasks in the Naval Research Laboratory's Space Robotics Lab at TRL 7 by the end of 2015. This paper describes the technology being developed by NRL and its partners and discusses the current laboratory–based campaign where these technologies would be demonstrated.

2 Technology Development and Demonstration plans

Currently, the robotics development program, known as the Phoenix program, is organized around a series of laboratory demonstrations of various tasks which would be required for a robotic servicing mission. These demonstrations, called "Phoenix Technology Demonstrations" or PTDs, are planned to be carried out in NRL's Space Robotics Laboratory.

Under the DARPA FREND program, NRL demonstrated autonomous grappling of an RSO at TRL 7 in 2008 [9]. The robotic arms used for these demonstrations were the Alliance Spacesystems FREND arms, 7–DOF 2.5 meter arms designed for operation in geosynchronous orbit, which are discussed in more detail below. Testing was performed in NRL's Proximity Operations Testbed (Figure **??**) and Contact Dynamics Testbed (Figure **??**). The design of these arms is detailed in [2].

The Proximity Operations Testbed (POT) is a dual gantry, 12 degree–of–freedom computer controlled orbital simulation system. It allows very high accuracy hardware–in–the–loop testing of robotic rendezvous and grapple maneuvers and of subsequent servicing applications. The contact dynamics of grappling large RSOs and smaller PODs spacecraft with FREND arms was tested in NRL's Contact Dynamics Testbed (CDT), which as of 2010 consists of a 15 by 20 foot grade AAA granite surface plate on which mass targets are floated using air bearings. The CDT's design allows it to simulate nearly frictionless contact dynamics in a planar setting, and thus complements the POT which cannot accurately model contact forces. Together, these testbeds allows the Phoenix team to conduct very high fidelity robotic operations testing up to TRL 7.

PTD 1 would occur in the last quarter of 2014 and, using the FREND arms and the Contact Dynamics Testbed, would demonstrate the autonomous capture of a PODs, placing the PODs on a work bench hard point, and the teleoperated unpacking of the PODs. Autonomous capture operations would be carried out in a fully autonomous fashion as detailed above, but the subsequent PODs manipulation tasks would be performed using a combination of scripting, partial autonomy, and teleoperation. The PODs capture would be simulated using NRL's Contact Dynamics Testbed, and would exercise the visual servoing system, the compliance control system, and the inverse kinematics and obstacle avoidance system. The subsequent teleoperated tasks would exercise the IRW, which must be developed to the point that it allows the operator to easy switch control modes, specify virtual camera views, and track the location of all the objects being manipulated.

In order to complete PTD 1, the FREND arms are being upgraded, with improved force/torque sensing capability, improved joint angle sensing capability, and the addition of a tool changer. Completion of the FREND arm upgrades, successful operation of the tool changer, and the demonstration of basic tool functionality are required in order to complete this demonstration successfully.

PTD 2 would occur in the first quarter of 2015 and would demonstrate robotic assembly of an external payload and the placement of the payload on an RSO mockup. This would primarily be a teleoperated emplacement task, and as such would exercise the ability of the IRW to provide sufficient environmental awareness cues to the operator and the ability of the teleoperator to prcisely control the arm in the presence of realistic communications delay and orbital lighting. The robotic emplacement of an external payload would be challenging to simulate due to the fact that the payload box would almost certainly exceed the FREND arm's payload capacity; it may therefore require gravity offloading techniques. NRL is pursuing the design of a gravity offloader sufficient for this purpose.

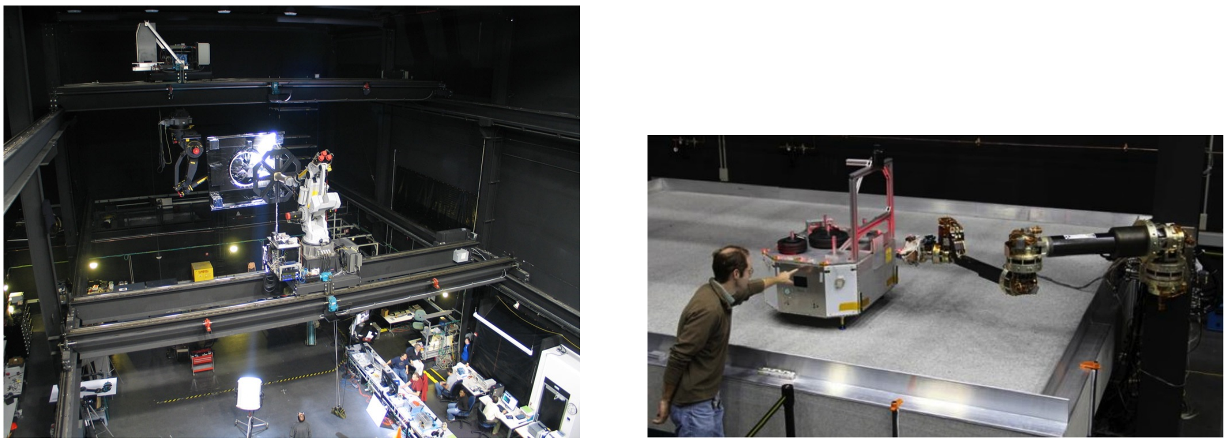
Finally, PTD 3 would occur in the third quarter of 2015 and would demonstrate dexterous robotic manipulation tasks such as freeing a stuck deployable or repairing torn thermal blanketing. The precise goals of this PTD are not yet fixed.

3 Major Phoenix Technology Components

To date, NRL has been commissioned to upgrade the existing FREND arms, to design, manufacture, and begin testing a tool changer for the FREND arms, to begin the design and manufacture of a dedicated inspection arm to aid in teleoperation of the FREND arms, to procure a number of robotic tools, and to develop and validate robotic algorithms, a robotic operator ground station, and a Concepts of Operations (ConOps) for candidate robotic servicing missions.

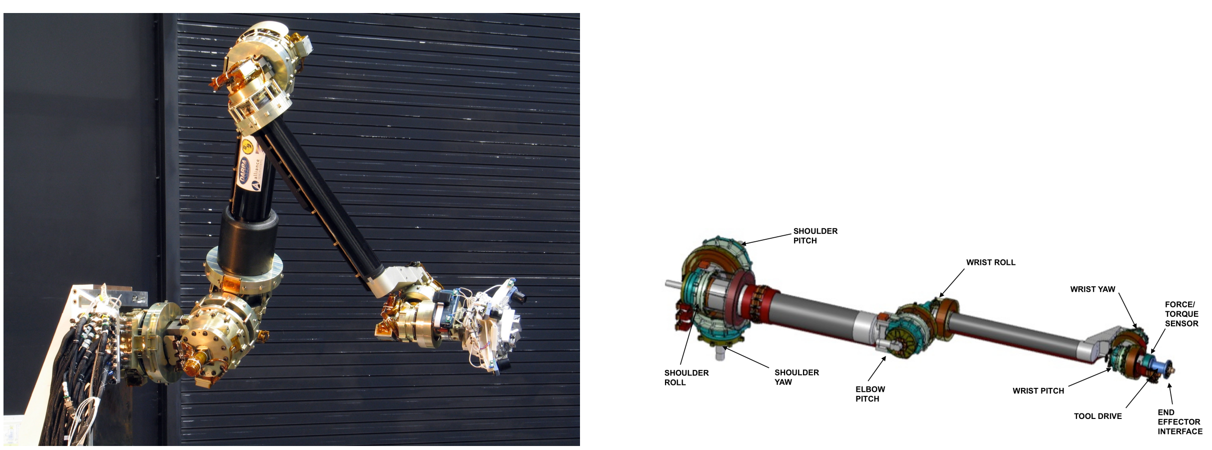
The FREND arms (Figure 2) were designed to perform autonomous grapple and subsequent repositioning of large RSOs such as disabled communications satellites or intact spent rocket bodies. As such, they are designed to be able to grapple and hold a relatively massive disabled client spacecraft during attitude control and orbital maneuvering operations. This requirement drive them to be stiff, as the rule of thumb for flexible spacecraft modes requires, in this case, that the first flexible mode for the arms should ideally be at least ten times higher than the bandwidth of the bus ACS system. The FREND arms are 78 kg and approximately 2.5 m in length; the mass is primarily driven by the stiffness requirement and by NRL's desire for arms that could be tested in 1 G and the length by the need to grapple a wide variety of spacecraft launch vehicle adapter interfaces. They are seven degree–of–freedom RPRPRPR arms, a requirement driven by the fact that the arms are intended to be used in close proximity to spacecraft with a wide variety of geometries that constrain the arm's workspace, such as upper stage kick motor nozzles, radiators, and so on. The redundant degree of freedom in the arms allows us to complete dexterous manipulation tasks while simultaneously avoiding obstacles by fully specifying the position and orientation of the end effector while retaining independent control of the elbow. The major components of the FREND arm are shown in Figure 2(b), and a heat map visualization of the dexterous workspace is show in Figure 3.

There are a number of upgrades currently being carried out to prepare the FREND arms for the successful completion of the PTDs. These include the addition of a tool changer, an upgraded force–torque sensor, and upgraded absolute joint angle sensors. In the implementation of the original FREND engineering development unit arm, the force/torque sensor was found to be extremely sensitive to temperature, and the upgraded force–torque sensor uses a different strain gauge technology in order to reduce this sensitivity. The absolute joint angle sensors were found to have been incorrectly designed, an a corrected design is being implemented. Finally, although the FREND arm was originally scarred to accept a tool changer, the original FREND program did not call for one and it was thus not manufactured. The Phoenix program requires several different tools, and the tool changer is thus a requirement for the new program. The tool changer is being manufactured by Oceaneering Space Systems and is based on Oceaneering's micro-conical interface [4],



(a) (b)

Figure 1. : (a) NRL's Proximity Operations Testbed (POT), and (b) NRL's Contact Dynamics Testbed.



(a) (b)

Figure 2. : (a) FREND robotic arm, and (b) FREND robotic arm major components.

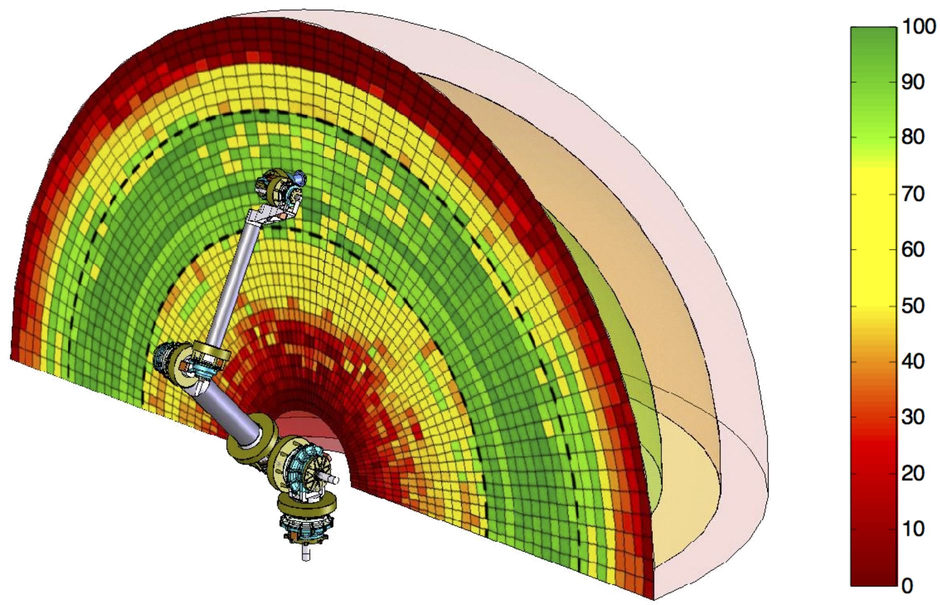


Figure 3. : FREND robotic arm heat map workspace visualization. Colors correspond to percent of end effector orientations at each position for which the inverse kinematics algorithm found a feasible solutions, with dark green corresponding to 100% and dark red corresponding to 0%. For details of analysis, see [2].

which has been used on the International Space Station as a standard robotic fitting on Orbital Replacement Units (ORUs).

Due to the complex manipulations required for orbital assembly, the majority of fine manipulation tasks will be tele–operated. It is well known that robotic tele–operation tasks can be performed more quickly and reliably when appropriate camera views are provided to the human operator [15], [8]. Obviously, this presents a challenge to a spacecraft design team: using multiple fixed cameras require few mechanisms, but entail considerable additional electronics, both to power the cameras and to multiplex the camera signals through the onboard signal processing circuitry and transmit the images to the ground. In addition fixed cameras rarely provide the precise camera views desired by an operator. Another option is to use a special–purpose "camera arm", typically a long robotic arm with a camera and lighting at its end effector. Camera arms have the advantage that they require fewer total cameras and offer much more flexibility in selecting camera views, at the cost of additional mechanisms and mass, and somewhat increased flight processor loading due to the need to control and command the additional arm. Also, strict reliance on a camera arm can lead to increased mission risk, both because the arm could theoretically fail, depriving the operator of the needed camera views, and because the arm constitutes an additional robotics operation collision risk.

The Phoenix architecture utilizes a hybrid approach. We are using a small number of fixed spacecraft deck cameras to provide overall environmental awareness to the operator, and are also designing a dedicated camera arm called the MeDUSA arm to provide camera views for precise teleoperation tasks such tool changes and external payload attachments. MeDUSA is anticipated to be a 3.5 meter, seven DOF arm that will have a mass of approximately 40 kgs. It will have fairly limited force and torque capabilities and will be sized primarily to be able to provide orthogonal camera views of a workspace to a teleoperator in a reasonable amount of time.

In addition to the FREND and MeDUSA arms, other major components will include an RPO sensor, tools, and a carefully designed spacecraft deck with fixed attachment points that would be used for assembly operations and tool fixturing. Tools would include, but are not limited to, Marman ring and separation bolt hole gripper tools which would be used to grapple resident space objects; a tool to grapple Payload Orbital Deliver System (PODs) boxes (Figure 4(a)); a tool to grasp assemble-able components; a large two–jaw gripper tool for grasping large structure, which may also serve as a leave-behind anchor point for an external payload (Figure 4(b)); and a general purpose parallel–jaw gripper tool for preparing external electronics attachment sites, free stuck deployable structures on newly launched spacecraft, and other external robotics servicing tasks. Marmon ring and separation bolt hole tools were designed and demonstrated under the FREND program; the PODs capture tool is a modification of the spacecraft grapple tool built by MDA and demonstrated in–space operations as part of the DARPA Orbital Express program [14].

4 Major Software Components

The top–level software architecture for the Phoenix robotic system is summarized in Figure 5. Note that there are essentially three layers of software control for the robot arm: a mission sequencer, which is currently instantiated as a state machine with special–purpose software entities to detect state changes by analyzing various sensor streams; an inverse kinematics layer; and a force/position servo control layer.

4.1 Robotic Control Modes, Mission Sequencing and Fault Detection

The Phoenix program's primary aim is to demonstrate an orbital capability and not to advance the state of the art in robotics in an academic sense. As a result, we have chosen to operate the robotic system using the least amount of autonomy necessary to carry out any given task. The types of robotic control we currently envision are:

scripting the robotic arm moves through a pre–planned trajectory using only proprioceptive sensors.

partial autonomy the robotic arm carries out a task using environmental sensors such as end effector cameras or the force–torque sensor. Tasks are carried out in steps, with the human operator explicitly issuing authority to proceed after each step is completed.

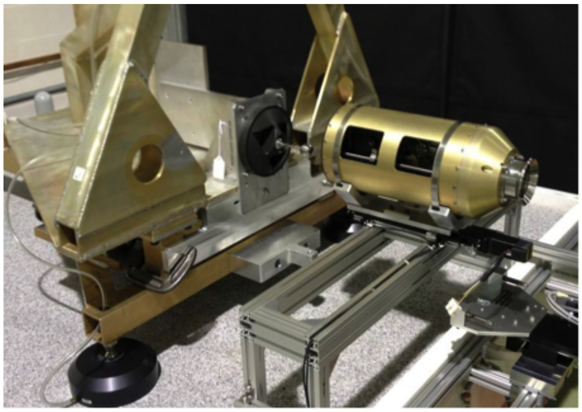
full autonomy the robotic arm carries out a task using environmental sensors, detects completion of steps, and issues itself authority to proceed if the last step was successful.

teleoperation motion of the robotic arm is commanded directly by a human operator using a hand controller, usually as commanded end effector motions. The robotic arm executes onboard servo control, and compliance control as needed, but all other control loops are closed via the human operators eyes and hands.

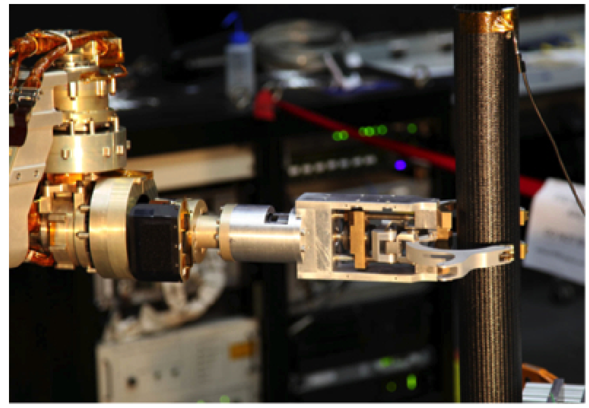
We anticipated that scripted operations would be used for large arm slews in order to prepare the system for other tasks. We anticipate that partial autonomy would be the most common command mode, and would be used for tool change operations, PODs unpacking tasks, and robotic assembly tasks. Teleoperation would only be used where the environment is not well characterized, such as during the emplacement of an external payload on an RSO. Teleoperation requires considerable attention from the operator, and it is difficult to fully validate teleoperation tasks prior to flight because of the variability inherent in human operators. Full autonomy would be used only in cases where the spacecraft does not have a reliable communications link to the ground – such as may be the case during the grapple of an RSO, for instance, due to the need of the spacecraft bus to match rotational rates with the RSO – or during operations where precise timing is critical, such as the post–servicing release of an RSO.

The academic community has devoted considerable effort into developing advanced planning and scheduling techniques, and these techniques have been demonstrated in a variety of guises on spacecraft mission such as Deep Space 1 [21], Earth Observing 1 [6], and the Mars Exploration Rovers [12]. Ordinarily, planners perform a tree search through a variety of potential state–action pairs in order to find an acceptable sequence of actions that will lead from the initial state to some desired final state. For Phoenix, however, we have chosen to use a much simpler state machine–based sequencer. This requires that the sequences of states and the action sequences needed to transition between them be designed by hand, but it is also straightforward to validate and to modify if needed; furthermore, it is computationally efficient. We utilize special–purpose "helper" modules to recognize when a state transition has occurred; for instance, two common states encountered when doing a tool change operation correspond to "aligning with the tool" and "aligned". We use a hand–coded software module to recognize when the end effector is aligned sufficiently with the tool to allow the tool changer to seat properly. During partially autonomous operations, both onboard recognition of a state transition and an explicit state transition command from the human operator are required for the state machine to transition. During fully autonomous operations, only onboard recognition of a state transition is required.

For fault detection we have added a number of states corresponding to a handful of common faults. A more commonly implemented spacecraft fault detection system would check various telemetry signals and notify the ground operations center in case of a less serious fault such as a temperature limit being exceeded; a more serious fault would require the system to automatically place the spacecraft into a safe state. Safe states normally turn off or take off–line noncritical systems, and reliably keep the spacecraft's solar panels towards the sun and the communications system in contact with the ground. The fault is then isolated and corrected via human intervention. On the FREND system, however, there is not one safe state than the spacecraft could always reach; instead, the correct response to a fault depends critically on the nature of the fault and the current state of the system. For instance, if the arm experienced a significant following error prior to the grapple of an RSO, the arm could immediately be turned off to prevent an inadvertent collision between the



(a)



(b)

Figure 4. : (a) POD capture tool being tested in an air–bearing facility, and (b) Universal Gripper Tool (UGT). The UGT is shown grappling a vertical spacecraft boom.

arm and the spacecraft deck. But if a following error is experienced after the RSO is grappled but before the relative rates have been removed, simply turning off the arm runs a considerable risk of damaging the arm or the end effector due to the large transient forces this would incur; thus, it may be safer to switch to a simpler control mode for the arm but allow it to continue operation until is can safely be stopped. Thus, for Phoenix we must carefully consider the proper course of action when a fault is encountered, and build these actions into the mission sequencer. To date, we have implemented fault detection and response logic for faults including an inability to find the feature of interest during visual servoing; arm following errors; and consistent, unexpectedly large sensed forces or torques. The philosophy for the planner and fault detection system and a more detailed description of it as it currently exists may be found in [11].

4.2 Inverse Kinematics and Trajectory Planning

The inverse kinematics layer is a resolved motion rate controller [20], which requires desired Cartesian end effector velocities as input, and calculates the joint trajectories necessary to execute them:

$$\dot{\theta} = \mathbf{J}^{+}(\theta)\left(\dot{\mathbf{x}}_d + \lambda\left(\mathbf{x}_d - \mathbf{x}\right)\right). \tag{1}$$

$\mathbf{J}(\cdot)$ is known as the Jacobian and is the partial derivative of the end effector velocity vector with respect to the vector of joint velocities:

$$\mathbf{J}(\theta) = \frac{\partial \dot{\mathbf{x}}_{ee}}{\partial \dot{\theta}} \tag{2}$$

$\mathbf{J}^{+}(\cdot)$ constitutes a pseudoinverse operator. Note that, here, $\mathbf{x}$ is a generalized pose vector, which ordinarily consists of three translational and three or more rotational elements. Addition and subtraction work in the usual way for the translational components, but orientation cannot be represented as a vector; therefore, addition and subtraction are assumed to refer to the appropriate mathematical operator for combining orientations. In the case of a quaternion representation [5], for instance, the terms are combined using quaternion multiplication:

$$\mathbf{q}_3 = \mathbf{q}_1 \cdot \mathbf{q_2} = \left(s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1\mathbf{v}_2 + s_2\mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2\right) \tag{3}$$

where the quaternion is representated by a scalar and a three element vector component, $\mathbf{q} = [s, \mathbf{v}]$, and $\times$ represents the cross product operator [10]. In the calculation given above, the difference between the desired and reported quaternion would be computed as the multiplication of the desired quaternion $\mathbf{q}_d$ with the conjugate of the reported quaternion $\mathbf{q}$, where the conjugate of $\mathbf{q}$ is $[s, -\mathbf{v}]$, i.e.

$$\tilde{\mathbf{q}} = \mathbf{q}_d \cdot -\mathbf{q} \tag{4}$$

We then use the vector component of $\tilde{\mathbf{q}}$ as a three–element set representing the orientation error in the RMRC control law given in equation 1.

In general, for a six–degree–of–freedom arm in a nonsingular pose, the pseudoinverse is exactly the normal inverse, and any six–element set of end effector velocities (the usual three translational velocities augmented with three rotational velocities) will result in a unique joint velocity vector. However, an arm with more than six degrees of freedom has a nonsquare Jacobian, and as a result there is an infinite set of pseudoinverses and a range of joint velocities that result in the same end effector velocities. The nullspace of such a Jacobian results in "self–motion", motion of the joints of the arm which do not result in motion of the end effector. Depending on which pseudoinverse is chosen, the resulting arm motion may be used advantageously to minimize the total L_2 norm of the joint ve-

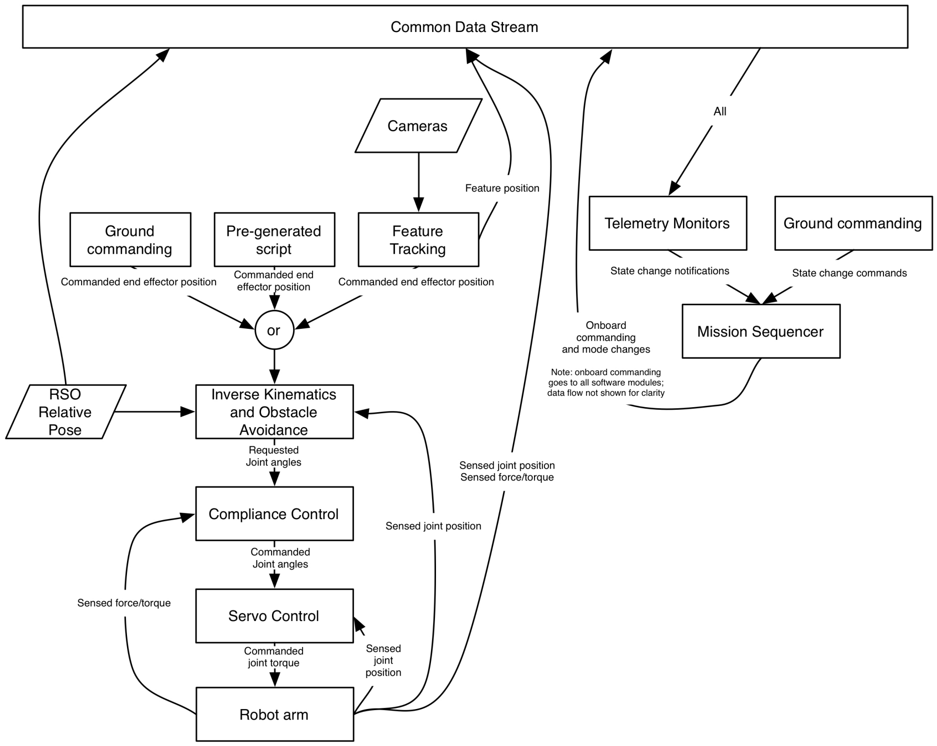


Figure 5. : Phoenix software major components and connectivity

locities – the effect of the standard Moore–Penrose pseudoinverse – or to perform more complex optimizations such as maximizing distance from obstacles in the arm's workspace or distance from singular arm poses. Utilizing the pseudoinverse in this way usually requires performing an explicit optimization within the Jacobian nullspace. Our instantiation of the resolved motion rate controller is based on Energid Corp.'s Actin toolbox. Actin implements a highly optimized Jacobian pseudoinverse calculation, and performs an optimization within the nullspace of the Jacobian in order use redundant degrees of freedom to optimize a metric specified by the user [3]. Under ordinary usage, the Phoenix software system primarily optimizes distance from workspace obstacles, and secondarily distance from singular arm poses. The servo control layer consists of seven independent PID servo loops running in dedicated FPGA hardware.

Phoenix also has visual servoing and compliance control modules. Visual servoing is used primarily for aligning the end effector to a hardpoint when the position of the hardpoint is not well known *a priori*, such as during an autonomous grapple of an RSO, when the relative pose of the RSO with respect to the robot's end effector is primarily determined using the RPO sensor suite. We expect this error to be on the order of 2 centimeters, which is not sufficiently accurate to guarantee proper seating of the end effector to the grapple point on the RSO; machine vision is used to reduce this error to an acceptable level. Machine vision is not expected to be routinely used to perform mating operations when the position of the mating feature is well known, such as during tool change operations. Special purpose feature detectors are being written to perform various visual servoing tasks. Feature detectors for Marman rings and separation bolt holes were developed under the FREND program. A feature detector for an ARToolkit–style fiducial is currently being developed for use for PODs grapples. Other special purpose feature detectors may be written as new visual servoing tasks are identified. Prior work done for visual servoing of the FREND robotic arm is described in more detail in [13].

4.3 Compliance Control

Compliance control is used whenever contact between the end effector and any other object is expected, including tool change operations, assembly operations, and grapple operations. The design of the Phoenix compliance control is detailed in [18]. Compliance control is particularly vital for grappling RSOs, since we wish to avoid imparting tumble rates as a result of the contact between the robot arm and the RSO. The Phoenix compliance control algorithm follows the form of a standard admittance controller [7], [19], but is formulated to operate in joint space:

$$\mathbf{C_m}\ddot{\tilde{\theta}} = -\mathbf{C_p}\tilde{\theta} - \mathbf{C_d}\dot{\tilde{\theta}} + \tau \tag{5}$$

where $\tilde{\theta}$ is interpreted as a modification of the nominal reference trajectory $\theta_{\mathbf{r}}(t)$, $\mathbf{C_p}$ is the desired stiffness matrix, $\mathbf{C_d}$ is the desired damping matrix, $\mathbf{C_m}$ is a the virtual inertia matrix, and τ is the vector of joint torques as calculated by the well–known transformation

$$\tau = \mathbf{J}^T(\theta)\,\mathbf{f}$$

where $\mathbf{f}$ is the vector of forces and torques exerted on the end effector as expressed in the end effector frame. Ordinarily, a compliance control law would be expressed directly in the end effector frame for the sound reason that in a joint space formulation as expressed above, the effective compliance of the system varies with the arm pose, whereas with a cartesian formulation the compliance is constant throughout the arm's workspace. However, this feature comes with a price: with a cartesian formulation, the size of the joint variations $\tilde{\theta}$ are not constant throughout the workspace for a given end effector force $\mathbf{f}$. The size of the joint deviation becomes larger as the arm nears a singularity. Furthermore, the compliance controller does not take singularities or potential collisions with the environment into account. Therefore, a compliance controller can actually force the arm towards a singular position or into a collision with the environment. For Phoenix, a joint space formulation was considered to be safer because the joint deviations can be bounded as a function of the joint torques, and therefore if the nominal trajectory stays sufficiently far from singularities and potential collisions, it can be guaranteed that neither will occur.

4.4 Ground Control: the Integrated Robotics Workstation

NRL is concurrently developing the Concepts of Operation for a variety of candidate missions and the robotic operator interface that would be needed to carry out these operations. The Integrated Robotics Workstation or IRW would serve as the primary human operator control for the robotic components of a robotic servicing spacecraft. As such, it would provide data displays, camera views, and other environmental awareness information to the operator, track the current position of the various tools and assembly modules, allow the operator to command the robot by calling up precomputed trajectories, directly command the arm via hand controllers, and possibly to design new scripted trajectories to carry out unforeseen tasks. It would also allow the operator to review sensor and data streams and to issue authority to proceed to the onboard planner state machine, or to flag failure modes and responses. The primary interface for relaying information to the operator is anticipated to be an interactive graphical display of the robotic system over which image streams from the cameras can be laid. Information from the joint angle sensors would be displayed by reflecting the motion the sensors detect onto a graphical model of the arm. The display would be rotatable and zoomable, and would allow the operator to specify vantage points where no physical cameras are located, thus allowing the operator to augment the camera views with virtual cameras. Obviously, the design of such an interface is a nontrivial task, and NRL is currently in the study and specification phase of this design.

5 Conclusion

The current program of laboratory technology development and demonstrations plans to demonstrate the necessary technologies, techniques, and concepts of operation to carry out a wide range of useful robotic servicing tasks, including orbital tug service, rapid resupply to GEO, orbital assembly, and limited robotic repairs in the laboratory at TRL 7 by the end of 2015. This development is the necessary precursor for a variety of useful satellite servicing missions. DARPA and NRL are currently developing concepts for such missions, which could be ready to launch later this decade.

6 Acknowledgements

The author would like to thank the members of the NRL Phoenix team, including Scott Chappie, Bernie Kelm, Tim Kennedy, Kiera Vincent, Jamie Lennon, Michael Mook, Fred Tasker, Sam Butcher, Paul Oppenheimer, Ken Weldy, Glenn Creamer, Jamison Hope, and Steve Koss. The author also thanks the DARPA Program Manager, David Barnhart, for his unflagging support.

References

[1] N Glenn Creamer, Ralph L Hartley, Jerome L Obermark, C Glen Henshaw, Stephen N Roderick, and Jamison R Hope. Autonomous release of a snagged solar array: Technologies and laboratory demonstration. *Journal of Spacecraft and Rockets*, pages 1–10, 2013.

[2] Thomas J Debus and Sean P Dougherty. Overview and performance of the Front-end Robotics Enabling Near-term Demonstration (FREND) Robotic Arm. In *AIAA Infotech@ Aerospace Conference, Seattle, Washington, AIAA*, volume 1870, page 2009, 2009.

[3] James D English and Anthony A Maciejewski. On the implementation of velocity control for kinematically redundant manipulators. *Systems, Man and*

Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 30(3):233–237, 2000.

[4] Michael L Gernhardt, Alistair P Johnston, and William D Wightman. Microconical interface fitting and interface grasping tool, June 14 1994. US Patent 5,320,395.

[5] Sir William Rowan Hamilton. Inscribed on the stones of the Brougham Bridge, Dublin, Ireland, 1843.

[6] Sandra Hayden, Adam Sweet, and Scott Christa. Livingstone model-based diagnosis of Earth Observing One. In *Proceedings of AIAA 1st Intelligent Systems Conference*, 2004.

[7] Neville Hogan. Impedance control: An approach to manipulation: Part II — Implementation. *Proceedings of the ASME*, 107(8), March 1985.

[8] Stephen Hughes, Joseph Manojlovich, Michael Lewis, and Jeffrey Gennari. Camera control and decoupled motion for teleoperation. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 2, pages 1339–1344. IEEE, 2003.

[9] BE Kelm, JA Angielski, ST Butcher, NG Creamer, KA Harris, CG Henshaw, JA Lennon, WE Purdy, FA Tasker, WS Vincent, et al. FREND: Pushing the Envelope of Space Robotics. Technical report, DTIC Document, 2008.

[10] DI Kolve. Describing an attitude. *NASA STI/Recon Technical Report A*, 95:80409, 1993.

[11] Jamie A Lennon, C Glen Henshaw, and William Purdy. An architecture for autonomous control of a robotic satellite grappling mission. In *AIAA Guidance Navigation and Control Conference and Exhibit, Honolulu, HI*, 2008.

[12] Jeffrey S Norris, Mark W Powell, Marsette A Vona, Paul G Backes, and Justin V Wick. Mars Exploration Rover operations with the science activity planner. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 4618–4623. IEEE, 2005.

[13] Jerome Obermark, Glenn Creamer, Bernard E Kelm, William Wagner, and C Glen Henshaw. SUMO/FREND: vision system for autonomous satellite grapple. In *Defense and Security Symposium*. International Society for Optics and Photonics, 2007.

[14] Andrew Ogilvie, Justin Allport, Michael Hannah, and John Lymer. Autonomous satellite servicing using the Orbital Express demonstration manipulator system. In *Proc. of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'08)*, pages 25–29, 2008.

[15] Naoji Shiroma, Noritaka Sato, Yu-huan Chiu, and Fumitoshi Matsuno. Study on effective camera images for mobile robot teleoperation. In *Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, pages 107–112. IEEE, 2004.

[16] Brook Sullivan, David Barnhart, Lisa Hill, Paul Oppenheimer, Bryan L Benedict, Gerrit Van Ommering, Laurie Chappell, John Ratti, and Peter Will. DARPA Phoenix Payload Orbital Delivery System (PODs): "FedEx to GEO". 2013.

[17] Brook Sullivan, Roger Hunter, Janine Bruhn, Erin Fowler, Lucy Hoag, Scott Chappie, Glen Henshaw, Bernie Kelm, Tim Kennedy, Mike Mook, et al. Phoenix Project Status 2013.

[18] Frederick Tasker and G Henshaw. Managing contact dynamics for orbital robotic servicing missions. In *2008 AIAA Space Conference, San Diego*, volume 86, 2008.

[19] Richard Volpe. Techniques for collision prevention impact stability and force control by space manipulators. In *Teleoperation and Robotics in Space*, pages 175–208. American Institute of Aeronautics and Astronautics (AIAA), 1994.

[20] Daniel E Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, 1969.

[21] Brian C Williams and P Pandurang Nayak. A model-based approach to reactive self-configuring systems. In *Proceedings of the National Conference on Artificial Intelligence*, pages 971–978, 1996.