

Advancing Safe Human-Robot Interactions with Robonaut 2

Julia Badger*, Aaron Hulse**, Allison Thackston***

*Software, Robotics, and Simulation Division, NASA-JSC, USA
e-mail: julia.m.badger@nasa.gov

**Rethink Motion, USA
e-mail: aaron.hulse@rethinkmotion.com

***Oceaneering Space Systems, USA
e-mail: allison.thackston@nasa.gov

Abstract

Robonaut 2 (R2), an upper-body dexterous humanoid robot, has been undergoing experimental trials on board the International Space Station (ISS) for more than a year. R2 will be integrated with a mobility platform in 2014 and will be able to maneuver around the ISS. Allowing R2 to go mobile inside the ISS requires new developments on several fronts, necessitating mechanical, electrical, software, and safety system upgrades. The ISS has stringent safety requirements for systems that work with and around the human crew members, and a redundant safety monitoring system is required to satisfy these constraints. This system, for the most part, responds to a requirement that the robot can only impart a limited force on any structure, equipment, or crew members in the ISS. R2 satisfies this requirement using several different force monitoring systems. These and the other safety monitors and controllers will be discussed here.

1 Introduction

Robonaut 2 (R2), an upper-body dexterous humanoid robot, has been undergoing experimental trials on board the International Space Station (ISS) for more than a year. R2 was launched in February 2011 and has been performing tasks designed to show its capabilities using human interfaces since early 2012. R2 has executed its tasks successfully, and has even shown its potential for meaningful contributions to the ISS community by demonstrating handrail cleaning, inventory, and data collection tasks that are currently chores completed by the human crew members.

Thus far, R2 has been restricted to working from a stanchion on orbit, but it will soon be integrated with a mobility platform. Once R2 is integrated with its mobility package, it will be able to maneuver around the ISS. Its objectives are twofold. First, R2 will strive to contribute by completing maintenance and cleaning tasks inside the ISS. Some example tasks are inventory manage-

ment, handrail cleaning, vacuuming filters, and data collection, such as air flow measurements. Second, the operational experience R2 gains inside the ISS will be essential in guiding mechanical and operational designs for a Robonaut unit to operate outside for extravehicular activities (EVA) in the future.

Allowing R2 to go mobile inside the ISS requires new developments on several fronts. The robot has to be upgraded with a set of legs, shown in Figure 1, as well as wireless communication capabilities and a battery to supply enough power for the larger machine. The computing capacity must be increased to deal with coordinating the control of more joints, integrating more sensors, and accommodating new functionalities, such as path planning, visual servoing, and localization. A novel control architecture that integrates high performance impedance controllers at the joint-level motor controllers with a model-based dynamic coordinated control system at the central processor was developed to satisfy performance and safety requirements for the mobile robot [1].

Additionally, a novel safety monitoring system had to be developed for the mobile robotic platform. The International Space Station has stringent safety requirements for systems that work with and around the human crew members, and a redundant safety monitoring system is required to satisfy these constraints. This system, for the most part, responds to a requirement that the robot can only impart a limited force on any structure, equipment, or crew members in the ISS. There are several ways that R2 could impart a potentially hazardous force. First, the robot could move too quickly and impact something. Second, the robot could be moving slowly, but with high torque or force, before impact. And finally, the grasping end effectors could impart a high pinching force.

Another safety mechanism of Robonaut 2 avoids its inadvertent release from constraints around the ISS. There are several controls in place while commanding the grippers to ensure that there is always one verified anchor point to structure at all times. Also, due to the redundancy requirements based on the ISS risk assessment, system



Figure 1. Robonaut 2 with Mobility Upgrade

health monitoring was implemented to stop the robot's operation if any essential component fails.

1.1 Related Work

The robotics industry is starting to move to human-safe practices similar to what R2 provides. Baxter, the Rethink Robotics humanoid, leverages series elastic actuators for static force limiting and velocity limiting for dynamic force limiting to remain human safe [2]. Baxter does not have precision coordinated control which reduces its operational capabilities to very specific sets of pick and place tasks. Universal Robots UR5 and UR10 have force limiting capabilities that rely on motor current for detecting over force conditions [3]. This allows the UR5 and UR10 to have sub-millimeter accuracy at the cost of force control/monitoring dynamic range. The KUKA Lightweight Robot uses torque sensors embedded at the joint level to prevent over force situations similar to the R2 safety strategy [4, 5]. KUKA is using their Lightweight Robot in the service and medical robotics fields due to the high level of safety inherent to a fully torque controllable robot, and it also features heavily in research involving human-safe control architectures [6].

1.2 Contribution

Robonaut 2's safety system has a novel method of monitoring the system force that could be imparted in several ways (static, dynamic, and crushing). This, partnered

with the controls in place that help the system avoid triggering the safety monitors, make R2 redundantly human safe while still allowing it to have good performance and capabilities. R2's operating concept is also different than many human-safe robots in that it is expected to operate and impart real loads while around people, instead of reducing its capabilities when humans enter its workspace.

This paper is organized as follows. Section 2 discusses the architecture of the R2 mobility system. Section 3 describes each of the safety monitors that the robot uses to satisfy the ISS safety requirements. Section 4 discusses the safety controls implemented on R2 and Section 5 concludes the paper.

2 System Architecture

The Robonaut 2 upper body has two seven degree of freedom arms with complex, tendon driven hands that feature another twelve degrees of freedom each. R2's neck has three joints for expanded viewing capabilities, particularly for teleoperators, and the upper body has a joint at its waist. R2 features series elastic actuators that are torque-commanded in the waist and the five major joints of the arms. Force and torque can also be measured in the six-axis load cells in each shoulder and forearm. The head features several vision sensors, including analog "teleoperation" cameras, machine vision cameras, and a time of flight sensor. The fingers feature several strain gauges to simulate a sense of touch.

The mobility upgrades to R2 bring many changes to the original system. R2 will be outfitted with two seven degree of freedom legs that are designed to be long enough to climb between nodes inside the ISS. Each leg has a gripping end effector capable of grasping handrails and seat track internal to the ISS. Each of the seven leg joints are series elastic actuators that are capable of being torque-controlled. The end effector is an over-center locking mechanism with a manual release in case of emergency. The end effector has a sensor package that includes a machine vision GigE camera, a 3D time-of-flight sensor, and a six-axis load cell. In both the upper body and the legs, the joints all communicate with the central processors (or "brainstem") via a communication protocol loosely modeled after the MLVDS standard.

R2 will be getting three Core i7 processors running Ubuntu as part of the mobility upgrades. The processors each have different functions in the control and safety monitoring of the robot. There is one processor whose primary job will be control of the robot; a second processor will listen to the communication with the joints and run the same calculations as the primary control processor as a check. The third processor will be connected to the other two via ethernet and will be comparing these calculations as another monitor. The third processor will also

be used for vision processing and other high-level supervisory control functions. All robot code will be running in the ROS framework [7].

R2 will be outfitted for a battery backpack to allow for wireless climbing through the ISS. R2 will be connected to the ISS wireless network via an embedded processor called the Cortex, which will be speaking ROS in a Digi Linux operating system. This Cortex will guard the ISS network from the traffic on the robot's internal network, limiting what data can be passed in or out to the ground control station. The Cortex also adds some safety monitoring capabilities to R2.

3 System Safety Monitoring

The safety monitoring system involves nearly all of the available torque and position sensing on R2, as well as all of the processors. For the ISS safety requirements involving excessive loads, there are three different types of force monitoring that are present in this system. Each of those (static, dynamic, and crushing) will be detailed in this section. Two other types of safety monitoring are also described. System health monitoring is required by the ISS safety community to ensure that the robot has redundancy equivalent to the criticality of the perceived risk of R2's excessive force capabilities. Trajectory monitoring is not a required part of the safety monitoring system, but is used to reduce the occurrences of potential robot safing actions due to other force monitors reaching a limit.

3.1 Static Force

The ISS safety community has imposed a requirement that R2 may not impose a static load of more than 45.45 kg (100 lbf) on any structure, equipment, or crew on board the ISS. R2 monitors static force by calculating the force that each limb is exerting at any time, as well as calculating the system or body force. This force is calculated using two different sets of sensors. First, the joint sensor package in the series elastic actuators are used to calculate the torque on each joint and a forward dynamics calculation finds the effective force and moment at each end effector. These forces are translated and combined in the body frame to ensure that the body is not imposing a force greater than the limit in the case that more than one end effector is contacting structure. Because of the many sensors that can be used to determine the torque and position in each joint, it is possible to consider this calculation redundant with health monitoring, which will be described later.

A similar calculation is performed using the six-axis load cells in each appendage. These direct force and moment measurements are transposed to the end effector frame for both arms and both legs. Also, the measured forces are transposed and summed in a body frame to get

the system force. As these measurements are not affected by singularities in the kinematic chain as the previous calculations are, this calculation is the sole source of force data on an appendage if that appendage becomes singular.

If any of the calculated force or moment values exceed the required limit, a safing event occurs. Once a safing event has been triggered, the motor power is removed from the system in several redundant ways.

3.2 Dynamic Force

Instances of dynamic force must also be accounted for and controlled in a safe manner. To limit the amount of dynamic force that can be applied by R2, the amount of momentum that is generated is monitored. An accurate mass model is used to calculate the inertia of each link, which is then multiplied by the velocity of each link, calculated from the forward kinematics of the robot, to produce momentum. To create a meaningful momentum, a base reference frame for the velocity calculation must be chosen. For R2 there are two scenarios that would produce high momentum, the end effectors separately and the body as a whole. Therefore the momentum is calculated on each end effector chain, as well as from a base world frame. For the end effector chains, the base is taken at the center body frame. For the body momentum, the base must be set at a frame connected to world. For R2, this frame coincides with the gripper that is attached to station, and therefore changes as R2 traverses throughout ISS. In order to determine an appropriate limit for momentum, the robot was commanded to impact a force plate at various speeds. The impulse produced from the impact was then analyzed. The requirements for station limit R2's peak load to 1000 lbf for short durations. This translates into approximately 34 N-s for metal on metal contact. By controlling the amount of momentum an individual end effector as well as the whole body can produce, the maximum dynamic force that can be produced by the system is limited to within safe parameters. If any calculated momentum value exceeds the required limit, a safing event occurs.

3.3 Crushing Force

The end effectors on the legs must clamp tightly around handrails or mated to seat track in order to support moving the robot's large inertia. The hands are not used for support of R2 and so they are not a part of this discussion. Because the clamping mechanism requires some amount of motor torque to drive it over center, caution must be taken with ensuring that the end effectors are only using the minimal force required during closing and clamping in order to not crush pipes, conduits, and wires that may be mistakenly grabbed while reaching for a handrail. There are two major ways that crushing force is limited. The force at the end effector is limited to a con-

stant value while the jaws are not closed. Once the jaws reach a position that has been defined as closed, this force limit is increased to allow the motors to drive the mechanism over center. In order to achieve this constant force limit, current to the motors are limited based on jaw position. The second protection is a pulse width modulation (PWM) duty cycle limit that is constant throughout all jaw positions.

3.4 Health Monitoring

In order to ensure that the sensors required for all the essential force, moment, position, and momentum calculations are working correctly, system health monitoring has been implemented as part of the safety monitoring system. The health monitoring is very complex, with many comparisons between sensors happening on various processors and on various calculated values from the measurements. The essential theme of the system is, however, that each sensor is compared against others to ensure that values are consistent. If any comparisons fail, a safing event occurs, stopping the robot so that more investigation can occur. The position sensors (two absolute position sensors, an encoder, and a halls effect sensor per series elastic joint) are compared in several places and as calculated position and velocity. The load cells are compared against the calculated force and moment at each end effector while the corresponding limb is not singular. As control of the limb is difficult near singularities, most operation occurs in regions where the force comparison is valid.

3.5 Trajectory Monitoring

Because the monitors discussed thus far are required by the ISS safety community to satisfy the risk assessment for R2 and the force requirement on board the ISS, when one of those monitors are triggered, the robot must safe itself by removing motor power. Once a safing event occurs, there is a process that must be followed in order to clear the safing event. This process often involves contacting several people in the operational chain, including the flight director. In order to reduce the number of safing events, a soft stop has been implemented at warn limits that are slightly inside the limits that trigger the safing events for the static force, dynamic force, and health monitors. This soft stop does not remove motor power, but instead only smoothly interrupts and arrests any commanded trajectories. This is very useful to guard against commanding the robot into a safing event while not affecting the safety monitors that will still guard against any failure situation that may trigger run-away conditions.

Another “unofficial” protection, similar to the warn limits, is trajectory monitoring. When this is active, each joint’s motion is monitored to ensure that the joint is following its trajectory within user-defined bounds. If the

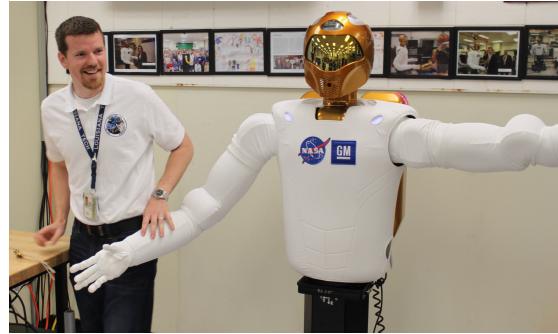


Figure 2. Trajectory monitoring allows R2’s arm to be stopped with little effort.

joint position lags by more than the pre-determined limit, a soft stop is called, cancelling each joint’s trajectory. Trajectory monitoring works when commands are given in joint or task space, as all user commands become joint commands before being sent to the individual joint controllers to be achieved. Because the joint-level controllers are embedded impedance controllers, where position references are achieved using a torque control loop, the distance measure that the trajectory monitor uses is actually a limit on the ratio of commanded torque for a joint to that joint’s torque limit. More discussion on the joint and coordinated controllers for R2 can be found in a previous work [1]. Because the joint torque limits can be set very low due to the accuracy of R2’s mass model, this protection enables R2 to be stopped using very little effort. Figure 2 shows trajectory monitoring in action.

4 Safety Controls

4.1 Inadvertent Release

The other major control requirement imposed by the ISS safety community restricts R2 to always have at least one attachment to structure (handrail or seat track). In order to avoid inadvertent release, a sophisticated control system is required to avoid commanding an end effector open if there is no other verified attachment point (the manual release mechanism will still function, however, in case of emergencies). The controller essentially filters all commands to enable or release a gripper through several conditions; if all the conditions are not met, the gripper remains parked and locked.

Figure 3 shows the overall control architecture. There is an arbitration component between the user and the supervisory level inputs and the coordinated control system, Robodyn. Since enabling the gripper is considered to be unsafe when it is the sole attachment point to structure, the commands to enable or park the gripper motors are handled only through the gripper supervisor in the coordi-

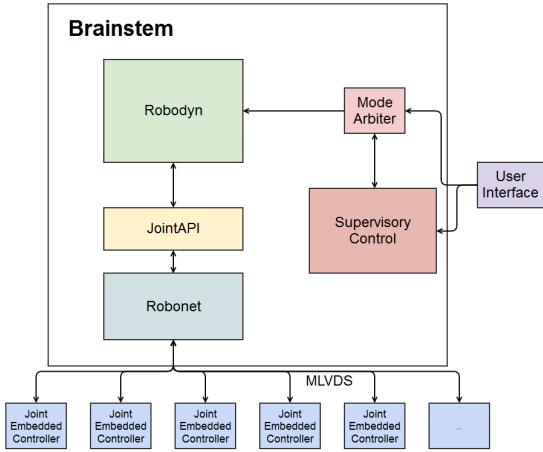


Figure 3. Overall control architecture for Robonaut 2

nated controller. Any command to change the state of the gripper motors are rejected by the Mode Arbiter.

The user or supervisory controllers can command the grippers to open, close, and lock, however. The command to lock is always allowed, as long as the gripper is not currently in the locked state. Commands to open or close the grippers, however, are only allowed when certain conditions have been met. The non-commanded gripper must be locked, there must be no faults in its joint-level controller, the motor state must be parked, and the gripper must have been verified as a base frame. Commands to grippers that are sent together in one package are handled separately and sequentially, so that commands to open two grippers that both satisfy the requirements will result in one gripper opening and the command to open the other gripper being rejected.

Grippers can be verified as a base frame through a special maneuver designed to determine that the gripper is indeed locked to structure. This command, called a “push-pull” command, is actually a supervisory controller that runs through a state machine to verify the attachment points of the robot. The push-pull command is only valid when both grippers are locked, healthy, and parked. The supervisor, once enabled, locks out and changes settings such as the desired dynamics and joint torque limits of the leg joints, and then sends up to two trajectories to the legs. The first trajectory is a roll maneuver about the gripper coordinate frame. If a gripper is not truly over-center and locked, this would cause the jaws to open. If the trajectory succeeds or if the jaws open, the base frame verification fails and the push-pull supervisor releases control. If the trajectory fails and the grippers remain locked, a second trajectory is sent. This trajectory includes both a roll in the opposite direction about the gripper coordinate frame as well as a short “pull” away from the attachment point. If this trajectory succeeds or if the gripper jaws

open, the base frame verification fails, and the push-pull supervisor releases control. If the trajectory fails and the jaws remain locked, the verification succeeds. Both grippers are marked as verified base frames and the push-pull supervisor releases control. Trajectory success and failure are judged using the trajectory monitoring described in the last section.

Finally, in order to have a redundant system throughout, one more protection on the gripper joint enable is needed. When a successful gripper command is issued, the gripper supervisor on two separate processors must identify the command as a valid one and both processors must send a gripper enable request to the third processor. That processor will issue a positive reply when two such requests happen within a short time frame, which then allows the gripper supervisor on the controlling processor to issue a gripper enable command to the joint-level controller. The third processor also monitors the state of the gripper motors and issues a safing event if it sees a gripper enabled without recently approving a request. This avoids the joint-level controllers from enabling the motors without a command.

4.2 Task Reconstruction

Not only is it important to monitor the forces created by R2, it is equally important to create a control system that will limit the chances of producing these forces. In order to limit dynamic force, the motion of links with high mass must also be limited. This is accomplished during Cartesian motion using a kinetic energy minimizing Jacobian. This is created by weighting the Jacobian of the system by the current inertia of each link.

$$\delta r_1 = W * J_1(q)\delta q \quad (1)$$

Then, the pseudo inverse of the weighted Jacobian is taken using the standard Moore-Penrose pseudo inverse.

$$\delta q = J_{1w}^+(q)\delta r_1 + [I_n - J_{1w}^+(q)J_{1w}(q)]y \quad (2)$$

Where $J_{1w}^+ \in \Re^{m_2 \times n}$ is the weighted pseudo-inverse. In this way, the least squares solution which minimizes kinetic energy is calculated. R2’s inverse kinematics implementation also uses a task based reconstruction method. This method utilizes the null space of a redundant system to solve for additional lower priority tasks in an iterative way. The weighted Jacobian is used at each iteration of these lower priority tasks. In this way, the solution created minimizes kinetic energy and therefore the momentum of the system.

5 Conclusions

The control and safety monitoring system developed for R2 is among the most advanced in the field of human

safe robotics. Because R2 works in very close proximity to humans in the exceptionally critical environment of the ISS, it is important to have a state of the art system that can be functional and safe. As Robonaut 2 continues to operate inside the ISS, continued developments to the control system to add functionality while decreasing the frequency of safing events will be pursued. The research conducted on the R2 robot on board the ISS will directly influence the control and safety system design of the extravehicular version of Robonaut. This Robonaut will need to go beyond the “fail safe” mentality of the intravehicular R2 and adopt a “must work” control philosophy. Though the extravehicular robot may operate primarily in the absence of EVA astronauts, both the potential of Robonaut being an astronaut assistant during an EVA and the need for protection of sensitive and critical equipment will necessitate an advanced safety system as well. The work presented here is a step along the path to developing the EVA Robonaut system.

6 Acknowledgments

The authors would like to thank Ross Taylor, Dustin Gooding, and Charles Claunch for their contributions to the Robonaut software. Immeasurable effort in organizing the overall safety data package for the Robonaut 2 mobility package, which inspired the work presented here, was completed by Alan Noblitt. Gratitude also goes to Myron Diftler, Charles Joyce, and the rest of the R2 team.

References

- [1] J. M. Badger, A. M. Hulse, R. C. Taylor, A. W. Curtis, D. R. Gooding, and A. Thackston, “Model-based robotic dynamic motion control for the Robonaut 2 humanoid robot,” in *Proc. of IEEE-RAS International Conference on Humanoid Robots*, 2013.
- [2] (2014) Rethink Robotics. [Online]. Available: <http://www.rethinkrobotics.com/>
- [3] (2014) Universal Robots. [Online]. Available: <http://www.universal-robots.com/>
- [4] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, “The DLR lightweight robot: design and control concepts for robots in human environments,” *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 376–385, 2007.
- [5] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald *et al.*, “The KUKA-DLR lightweight robot arm-a new reference platform for robotics research and manufacturing,” in *Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK)*. VDE, 2010, pp. 1–8.
- [6] F. Flacco and A. De Luca, “Safe physical human-robot collaboration,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2072–2072.
- [7] (2014) Robot Operating System. [Online]. Available: <http://www.ros.org>