

Model-based Robotic Dynamic Motion Control for the Robonaut 2 Humanoid Robot

Julia M. Badger¹, Aaron M. Hulse², Ross C. Taylor³, Andrew W. Curtis⁴, Dustin R. Gooding¹, and Allison Thackston⁴

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 soon be upgraded with two climbing appendages, or legs, as well as a new integrated model-based control system. This control system satisfies two important requirements; first, that the robot can allow humans to enter its workspace during operation and second, that the robot can move its large inertia with enough precision to attach to handrails and seat track while climbing around the ISS. This is achieved by a novel control architecture that features an embedded impedance control law on the motor drivers called Multi-Loop control which is tightly interfaced with a kinematic and dynamic coordinated control system nicknamed RoboDyn that resides on centralized processors. This paper presents the integrated control algorithm as well as several test results that illustrate R2's safety features and performance.

I. 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, shown in Figure 1, was launched in February 2011, checked out during the fall and winter of that year, and has been performing tasks on a specially designed task board 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 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 the plan is for R2 to be integrated with a mobility platform and a battery backpack. Both are currently under development and are planned to be launched in late 2013. Once R2 is integrated with its mobility package, it will be able to maneuver around the ISS. Its objective will be twofold. First, R2 will strive to contribute by completing maintenance and cleaning tasks inside the ISS. Some example tasks are inventory management, handrail cleaning, and data collection, such as air flow measurements. Second, as



Fig. 1. Robonaut 2

R2 gains operational experience inside the ISS, it will be essential in guiding mechanical and operational designs for an R2 unit to operate outside for extravehicular activities (EVA) in the future.

R2 in its mobile configuration has significant capability for moving from one location to another. Figure 2 shows a prototype version of the future on-orbit system in the NASA-Johnson Space Center's Active Response Gravity Offload System (ARGOS). Each appendage, or leg, features seven degrees of freedom as well as a multi-use gripping end effector. Each of the fourteen degrees of freedom are single-axis series elastic actuators. The overall length of the robotic legs was designed to be long enough to safely traverse the ISS nodes that connect lab modules together while still allowing R2 to maneuver within the size constraints of the lab. The end effectors have a sensor package that includes cameras, load cells, and position sensors. The end effector is capable of attaching to handrails and seat track inside the ISS, and the sensor package will be instrumental in automating safe grappling operations. The upper body currently on the ISS features several series elastic joints, including a waist joint and five joints per arm. R2's head has three degrees of freedom to supply good views around the robot's workspace. Each wrist has two degrees of freedom and the hands feature twelve degrees of freedom each. R2 has distributed joint-level processing on the motor drivers as well as centralized processing, called a brainstem, where coordinated control is computed.

This complicated system requires careful planning of its control system architecture. The controller must accomplish

¹NASA- Johnson Space Center, Houston, TX 77058, USA {julia.m.badger, dustin.r.gooding} at nasa.gov

²Jacobs Technology, Houston, TX 77058, USA aaron.m.hulse at nasa.gov

³Aerodyne Industries, Houston, TX 77058, USA ross.c.taylor at nasa.gov

⁴Oceanering Space Systems, Houston, TX 77058, USA {andrew.w.curtis, allison.thackston} at nasa.gov



Fig. 2. R2 testing mobility algorithms in ARGOS

many different types of objectives. First, the legs must be able to move the large inertia of the robot precisely so that the end effector can grasp the handrails in an efficient and safe manner. At the same time, the robot must be safe for astronauts to work around. So, for the mobility platform, any controller design must be able to support the ability to have low steady state error while maintaining low torque limits. The controller must be able to deal with astronauts moving the robot out of their way as it follows precise motion plans. Dynamically, the controller must accommodate high inertias on orbit but essentially no gravity. However, on Earth, it must be able to compensate for gravity in order to do testing and development. Finally, the controller must be able to plan for coordinated motion over the many degrees of freedom in the robot.

A. Contributions

The contributions of this work are two-fold. First, the Multi-Loop controller on the embedded processors is designed to achieve high performance impedance control at the joint level. It provides flexibility and performance while delivering important safety assurances. Its custom design complements the model-based dynamic motion controller on the brainstem. The contributions of this controller, nicknamed RoboDyn, include online trajectory generation and inverse kinematics to achieve precise position commands, as well as a model-based inertia and gravity compensation dynamics algorithm that is commanded as feed forward torque to the embedded controllers. Joint torque limits and controller dynamics are tightly coupled with the current robot trajectory and dynamics to increase the performance of the robotic control system.

B. Related Work

There are many related control system designs that inspired the current work. Impedance control for robotics was

outlined by Hogan [1] and has been used by many robots to affect the coordinated control of manipulators [2], especially on robots that work around humans [3], [4]. Robonaut 2's current on-orbit control system is an impedance control law with the position loop closed on the brainstem [5].

Featherstone's Recursive Newton Euler algorithm is used as part of the dynamics formulation [6]. A coordinated control algorithm called Whole Body Control that is also based on Featherstone's work allows torque control of higher-level skills and tasks [7]. Walking control algorithms such as [8], [9] also use model-based torque control for fast and coordinated motions that combine tasks of balance and gait control in an intelligent way.

C. Outline

The overall control architecture is described in Section II. The kinematic and dynamic components of the brainstem controller, RoboDyn, are discussed in Section III. Experimental results of climbing and other tasks are presented in Section IV. Finally, future work in extending this controller to the R2 upper body on orbit is discussed in Section V.

II. CONTROL ARCHITECTURE DESIGN

The control system for the integrated R2 humanoid robot is unique for several reasons, one of which is the overall architecture of the system. The architecture consists of model-based, joint-level (embedded) impedance control components designed to accurately track command inputs from the centralized coordinated dynamic trajectory controller, RoboDyn. Both levels of controllers are designed to reject disturbances up to an allowable threshold, then safely and gracefully adjust to accommodate them. Figure 3 shows the overall software architecture that will be presented here. This figure also shows the Supervisory Controller layer, which consists of the higher level controllers that interface with the RoboDyn control system to accomplish tasks in a somewhat intelligent manner. These supervisory controllers and user commands are arbitrated with a Mode Arbiter. The description of these components is outside the scope of this paper.

A. Multi-Loop Embedded Impedance Control

The Multi-Loop embedded controller forms the foundation upon which the overall control system is built by ensuring that each joint tracks its commanded trajectory while conforming to desired dynamic performance characteristics and safety thresholds. It achieves this by employing four consecutive high rate (5kHz) control loops, each consisting of sensor-fed, model-based feed forward control terms and traditional PID controllers to achieve tight tracking performance. The feed forward terms have been designed to largely compensate for the non-linear physical characteristics of the R2 joints, thus minimizing and linearizing the contributions needed from the PID control loops.

The lowest level loop is a current loop designed to track the desired torque-generating motor current of the R2 joint brushless DC motors. The feedback for this loop is

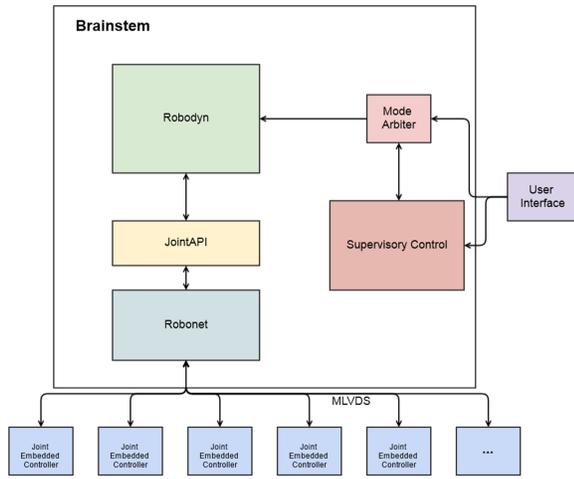


Fig. 3. Overall control architecture for Robonaut 2

measured by sensing the current on two of the three motor bridge circuits. Feed-forward terms compensate for the motor and bridge electrodynamics. The current loop can also be used without direct current sensing while maintaining good current tracking performance. The current loop serves two purposes. First, it linearizes the electrodynamics of the motor, making it a pure torque source, and second, it provides current limiting to protect the motor and motor controller.

A motor velocity loop wraps the current loop to track the desired motor velocity. The velocity loop includes a feed-forward model that compensates for Coulomb and viscous friction and feedback is provided by a motor-mounted incremental encoder. The velocity loop has three important functions. First, it linearizes friction and the input inertias to the gearbox. Second, it provides motor speed limitations. Finally, it allows a “virtual hardstop” capability to avoid hitting joint limits.

The third loop is a torque loop that tracks the desired joint torque and has a feed forward term for acceleration compensation. Joint torque feedback is calculated by measuring the deflection of the series elastic joint spring. The outermost loop is an impedance loop for tracking desired joint position subject to desired stiffness and damping constraints. This loop carefully integrates and limits the desired torque contributions generated by the position error, joint velocity, feed forward dynamics, and gravity compensation terms to ensure smooth torque trajectory commands are sent to the torque loop.

All loops except for the impedance loop are configured upon start up with pre-tuned gains, limits and parameters. The impedance loop receives real-time updates to its parameters from Robodyn, including desired stiffness and damping terms calculated from the joint’s effective inertia and user-specified natural frequency and damping ratio. These configuration-dependent gains give consistent robot performance throughout its workspace. By compensating for the effects of system dynamics in the joint level embedded controller, the RoboDyn controller can run at a slower rate

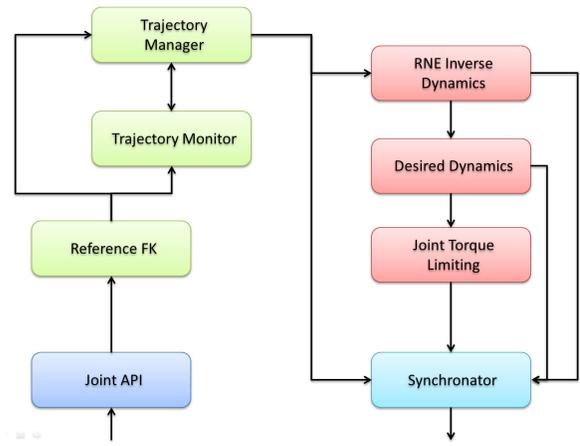


Fig. 4. Robodyn: brainstem control architecture for Robonaut 2

of 50 Hz with no noticeable degradation of performance.

B. Model-based Robotic Dynamic Motion Control

The Robodyn controller handles the coordinated control calculations necessary for smooth, integrated Cartesian control of R2. It sends synchronous commands to the joint embedded Multi-Loop controllers such that, if the embedded controllers achieve those commands, then the desired Cartesian trajectory is accomplished. A block diagram of the controller is shown in Figure 4. RoboDyn can be broken up into kinematic and dynamic components. These components are tightly coupled via inputs and outputs as well as in execution order for smooth coordinated motion.

The kinematic part takes joint and Cartesian commands from the user or from a supervisory control component and creates joint trajectories to achieve the commands. It generates position, velocity, and acceleration references per joint. These velocity and acceleration references are fed to the dynamic part, which creates feed-forward torques to compensate for inertia and gravity. The dynamic part also outputs a stiffness and damping for each joint based on each joint’s calculated effective inertia and desired natural frequency and damping ratio. Finally, it outputs joint torque limits.

C. Extension to Non-Series Elastic Joints

While the overall impedance controller works well on joints that can be torque controlled, it must be adapted for use on non-series elastic joints, including Robonaut’s neck, wrists, fingers, and grippers. Each of these joint’s motor controllers features an embedded control system that is similar to the Multi-Loop controller, except that the torque loop is removed and the outer loop is a pure position loop that generates the desired velocity command. This is similar to many commercial off-the-shelf motor controllers. In some cases, such as the wrists and fingers, the “embedded” controller actually lives on the brainstem due to the lack of programmable motor drivers for these joints. This loop runs at about 500 Hz to ensure smooth control of the hands.

The non-series elastic joints are generally treated the same way by the kinematic RoboDyn components as the series elastic joints are. This is not true for the dynamic components, which essentially ignore the existence of these joints, except for their mass properties. Consequently, these joints use static gains.

III. ROBODYN COMPONENT DESCRIPTIONS

A. Kinematic Components

The kinematic components are responsible for taking joint and Cartesian commands from the user or supervisory controllers and creating joint “breadcrumbs” or waypoints that are sent down to the Multi-Loop embedded impedance controller. The Trajectory Monitor serves as the gateway to this set of components. It accepts joint and Cartesian reference commands and tracks the progress of these requests. The Monitor passes on these pose and joint references to the Trajectory Manager, where the trajectory is achieved. The overall functionality of the kinematic components is online as opposed to pre-planned in order to accommodate both command interruptions as well as to monitor task achievement and issue corrective actions as necessary.

The Trajectory Manager includes trajectory generation as well as inverse kinematics. Joint commands are achieved using a trapezoidal velocity trajectory generator that allows nonzero initial and final conditions (position, velocity, and acceleration) of the joint. This ensures smooth joint commands even when interrupting movements in progress. The trajectory generator uses maximum velocity and acceleration as well as a desired duration which is achieved unless the constraints require a longer duration. Joint trajectories are synchronized so that all joints complete their trajectories at the same time. This is accomplished by calculating the shortest possible duration for each joint and then recalculating the trajectories so that they all match the duration of the longest running joint trajectory. In addition, a list of joint commands, or breadcrumbs, can be directly fed into the Trajectory Manager, bypassing the trajectory generation step.

Cartesian commands use the same trapezoidal velocity trajectory generator as joint commands. Orientations are converted into quaternion space and then seven trapezoidal velocity trajectories are generated. The only difference is that when each point is produced, the quaternion must be normalized. The trajectory generator can handle nonzero initial and final conditions for Cartesian trajectories, as well, but it is important to only allow this when the node is currently under Cartesian control. Transitioning from joint to Cartesian requires a smooth stop of motion before beginning the Cartesian trajectory because a smooth transition of the node in Cartesian space does not guarantee a smooth transition at the joint level. A multi-node Jacobian pseudo-inverse iterative inverse kinematics calculation converts from task space to actuator space while allowing the constraint of multiple nodes simultaneously. It also supports ignoring axes by removing rows from the Jacobian.

For both joint and Cartesian trajectories, the parameters of the trapezoidal velocity trajectory are calculated when a new trajectory is received and then each breadcrumb is generated online from these parameters. Current commands are used for initial conditions rather than actual conditions to guarantee smooth commanding. This motivates the Trajectory Monitor which makes sure the actuals and commands do not deviate too significantly.

The primary role of the Trajectory Monitor is to monitor joint commands and actual measurements and to trigger a replan when they deviate too much. The position error of each joint is multiplied by the position gain and a scaling factor and is compared to the joint torque limit to determine excessive deviation. When the scaled deviation is exceeded, it can be concluded that the trajectory is not being followed correctly, possibly due to insufficient torque limits or an obstacle. The current response is to gracefully stop trying to achieve the trajectory and hold the current position. The joint that is replanned gets a rapid trajectory from its current command to its actual position providing a smooth reduction in force on the obstacle. Since Robonaut works with humans and they are often the obstacle, this is desirable to prevent a jarring drop in force. Joints that are part of the same trajectory are also stopped with a smooth transition to zero velocity. It is left up to a higher level supervisor or human operator to decide how to continue.

In addition, the Trajectory Monitor keeps track of overall progress. Trajectories may have multiple steps and it is important to know where the plan was interrupted so that more complex moves need not be restarted. The Monitor keeps a record of joint and Cartesian goals and documents their completion over time.

The integrated design of the kinematics components allows for many interesting capabilities, including allowing the designer to effectively plug in different trajectory generators and replan responses. Because Robonaut 2 works closely with humans in a sensitive space environment, the ability to dictate settings such as maximum velocity and acceleration in the Trajectory Manager helps ensure that the robot will maintain safe speeds and forces.

B. Dynamics Components

The intent of the dynamic components is to compensate for the inertia of the system both in microgravity and while testing the robotic system on Earth. In the Inverse Dynamics component, the recursive Newton-Euler algorithm (RNEA) and the composite-rigid-body algorithm (CRBA), both presented in Featherstone [6] are combined into two functions. One function handles the kinematics pass (finding the velocity and acceleration of each joint) of the algorithm, while the other does the dynamic work (composing force/torque wrenches and calculating effective inertia per joint). This algorithm calculates the compensation needed due to the acceleration of the effective inertia and the Coriolis effect based on the desired velocity and acceleration at each point of the commanded trajectory. This algorithm also calculates gravity compensation by simply including

the vector opposing gravitational acceleration in the tree’s base frame; this is then carried through to each subsequent joint as a torque needed to counteract that acceleration. The inertial and gravity compensation torques can be calculated separately by running the algorithm twice, first by including only the desired acceleration and velocity terms and the second by only specifying the gravitational acceleration vector.

This algorithm is highly effective and similar formulations have been used on many manipulators. The interesting part of the algorithm is the production of the effective inertia for the Desired Dynamics component. The intent of this component is to be able to specify the second order linear dynamics of each joint. It is possible to specify a natural frequency and damping ratio for each of the joints using the following equations.

$$k = m\omega_n^2 \quad (1)$$

$$d = 2\zeta\sqrt{km} \quad (2)$$

This gives a stiffness and damping that is configuration-dependent while ensuring the overall dynamics of the system remain consistent.

Practically, the minimum damping must be limited per joint. Because the robotic legs naturally start in a singular straight out position, the roll joints have very little effective inertia, which makes the damping in those joints very small. Small discrepancies in the joint’s construction and sensor calibration causes the joint to swing when the damping is very low, so a minimum damping value is enforced. The Desired Dynamics component requires that the desired damping ratio is respected in lieu of desired natural frequency when limits are encountered, so stiffness is recalculated in the following way.

$$k = (d/(2\zeta))^2/m, \zeta \neq 0 \quad (3)$$

Maximum stiffness is also limited for a stability safety check, though this limit can be set very high. A more effective way to command the joints is to choose the appropriate desired natural frequency value for the given joint such that stability is ensured for all effective inertias. If a maximum stiffness limit is exceeded, the damping value is recalculated based on the limited stiffness and then subsequently checked against the damping limits. If necessary, the stiffness value will be calculated once again using the minimum or maximum damping limit. One can prove that this amended stiffness will be below the maximum stiffness limit when reducing the damping to its maximum, but must be checked once again if the damping was raised to its minimum value.

The final dynamic component sets joint torque limits for the impedance feedback loop. This is an important function because it allows these joint torque limits to be set by the user for operational concerns as well as systematically set by supervisory components for more autonomous behaviors. These limits affect only the feedback torque on each joint, which allows the full effort of the gravity and inertial compensation to be applied. This results in better performance as well as increased safety due to the low joint torque limits that

can be used to affect forces on the environment. An example of why this is useful will be discussed in Section IV.

The dynamics components feature a blending functionality to avoid large jumps in feedforward or feedback torque commands on the joint level. Changes in desired natural frequency and/or damping ratio are blended over time. The user can specify how much stiffness or damping can change per time step to ensure a smooth blend between soft and stiff configurations (these will be described in the following section). Joint torque limits are blended in a similar way between weak and strong configurations. Because it is important to be able to turn gravity compensation on and off for this system due to its testing and operational environments, a gravity vector scale factor exists and is blended when gravity compensation is turned on or off or when the base frame changes in the testing configuration. The user can specify how much this unit scale factor changes per time step.

IV. EXPERIMENTAL RESULTS

A. Simulated Microgravity Climbing

The R2 mobility system has been successfully tested in NASA-Johnson Space Center’s Active Response Gravity Offload System (ARGOS), as shown in Figure 2. R2 takes steps on a simulated ISS handrail test mock-up to aid in control system and algorithm development for eventual climbing on the International Space Station. The algorithm that is used to successfully grasp handrails and take multiple steps leverages many capabilities of the overall control system. The climbing algorithm currently relies on operators specifying an approximate location of the handrails with respect to one another; future iterations of this algorithm will take advantage of the robot’s sensor packages to use visual references to aid in handrail grasping.

In the simulated microgravity climbing test, R2 starts with one end effector grasped to a handrail. In ARGOS, gravity is offloaded from the robot’s center of mass, so the reaching leg still feels the effect of gravity while the base leg is effectively offloaded. So, gravity compensation is used on the reaching leg. Both legs are in a stiff configuration ($\omega_n = 20, \zeta = 1.2$) with the base leg strong (joint torque limits, $\tau_{lim} = 100$) and the reaching leg weak ($\tau_{lim} = 20$). The integrator on the embedded impedance loop is turned on and both legs are stiff in order to allow the leg to have very low steady state error. From testing, the legs are able to move the nearly 300 kg robot and reach over 1 m with less than 1 cm error.

Once the reaching leg is in a rough position over the handrail, its end effector sensor package determines the location and orientation of the handrail. A low rate visual servoing supervisory control component sends Cartesian commands to put the gripper in position over the handrail. It is then lowered and the gripper is commanded to close. The reaching leg is softened ($\omega_n = 0.5, \zeta = 1.5$) to allow the gripper to close completely. Before the next step begins, the base frame of the kinematic chain switches to the newly attached leg as the joint torque limits and desired dynamics smoothly increase while the releasing leg’s joint torque limits decrease. This ensures that the kinematic loop acts more like

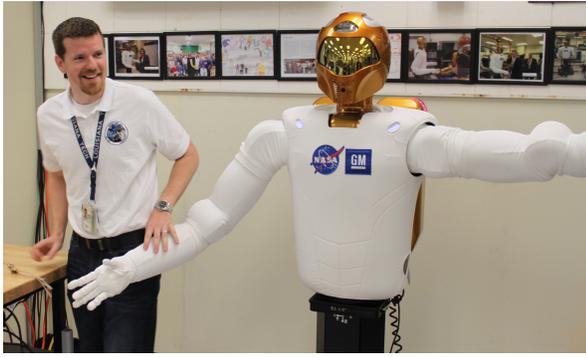


Fig. 5. Human interaction with Robonaut 2

a chain by allowing the non-base leg to always yield to the base leg.

B. Line Drawing

The Multi-Loop controller allows very tight control of the compliant series elastic joint such that even though the robot is capable of working around humans, it still has precision that is comparable to stiffer industrial manipulators. This capability is very important for Robonaut's application of climbing through the ISS using handrails and seat track interfaces while also allowing humans to enter its workspace and interact with the robot. This ability is shown by a simple drawing example, where the robot leg draws a rectangle on a white board repeatedly. While the leg is very stiff, the joint torque limits can be held low enough that human interaction is possible during this task. This simulates the conditions that are present during the climbing task described in the previous section. Another application of this is welding, which is an extra-vehicular task that has been proposed for Robonaut 2.

C. Human Interaction

Robonaut was designed to work safely alongside humans, and this control architecture is also based around that requirement. One of the essential properties of the system for this requirement is the torque limited embedded impedance controller. Because the gravity and inertia compensation provided by the RNEA provides a sufficient feed-forward model of the motion, the joint torque needed for the feedback loop can be very limited. In addition to the relatively weak overall forces generated by the system, the trajectory monitoring that triggers the automatic replan ability can be used to detect when the robot hits an obstacle along its path and smoothly arrest its trajectory. This is demonstrated in Figure 5; a human enters the robot's workspace, stopping the trajectory of the arm. Robonaut has been programmed at the application layer to continue upon sensing a small impulse in its six-axis load cells. At this point, it continues along its trajectory until it completes successfully or encounters another obstacle.

V. CONCLUSIONS

This paper introduced a novel control architecture for the Robonaut 2 humanoid that currently is working on board the International Space Station. The new addition of legs to

the current upper body will require that the system maintains its safety around humans while also operating with precision similar to stiff industrial manipulators. The integrated control architecture presented here allows both requirements to be met; this is important for the space application as well as for the possibility of applying this technology to other precision robots that must have humans in their workspaces. The algorithm presented features an embedded impedance controller on the motor driver level that is tightly interfaced to the brainstem-level control system.

This architecture is extensible in many ways. First, a task reconstruction method of inverse kinematics would allow more control over the vast nullspace that Robonaut 2 enjoys [10]. Because many safety monitors must be in place to satisfy the International Space Station requirements, this method would allow more control over the values that are monitored, namely momentum and force. Hybrid force/position control is also possible using this architecture [11], and would add many application capabilities to the system. Future plans include adding more sophisticated path planning and supervisory control algorithms on top of this control system architecture.

ACKNOWLEDGMENT

The authors thank the Robonaut 2 team, particularly Ron Diftler, Harrison Pham, and Mark Pape for their contributions to this work.

REFERENCES

- [1] N. Hogan, "Impedance control: An approach to manipulation: Part ii-implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, 1985.
- [2] R. Colbaugh, H. Seraji, and K. Glass, "Direct adaptive impedance control of robot manipulators," *Journal of Robotic Systems*, vol. 10, no. 2, pp. 217–248, 1993.
- [3] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *Proc. 1995 IEEE International Conference on Robotics and Automation*, vol. 3, 1995, pp. 3097–3102 vol.3.
- [4] A. Albu-Schaffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbock, 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] M. Diftler, J. Mehling, M. Abdallah, N. Radford, L. Bridgwater, A. M. Sanders, R. S. Askew, D. Linn, J. Yamokoski, F. Permenter, B. Hargrave, R. Piatt, R. Savely, and R. Ambrose, "Robonaut 2 - the first humanoid robot in space," in *Proc. 2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2178–2183.
- [6] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2008.
- [7] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *Proc. 2006 IEEE International Conference on Robotics and Automation*, 2006, pp. 2641–2648.
- [8] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.
- [9] J. H. Park, "Impedance control for biped robot locomotion," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 870–882, 2001.
- [10] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison Wesley, 1991.
- [11] T. Yoshikawa, T. Sugie, and M. Tanaka, "Dynamic hybrid position/force control of robot manipulators-controller design and experiment," *IEEE Journal of Robotics and Automation*, vol. 4, no. 6, 1988.