# FRASIER: Fostering Resilient Aging with Self-Efficacy and Independence Enabling Robot
## Team Northeastern's Approach for RoboCup@Home

Tarik Kelestemur, Anas Abou Allaban, and Taskin Padir

College of Engineering, Northeastern University
360 Huntington Ave., Boston, 02115 MA, USA
http://robot.neu.edu/frasier/

**Abstract.** This report describes Team Northeastern's progress to meet the requirements of the 2017 RoboCup@Home Domestic Standard Platform League (DSPL). We three novel methods on human supervised autonomy, risk-aware compositional autonomy and single source any angle path planning to be implemented on Toyota HSR. We also present our results on autonomous navigation guided by natural language commands, manipulation of unknown objects, and object detection. We demonstrate our results using a FetchResearch platform as we await the arrival of our Toyota HSR to participate in the 2017 RoboCup@Home DSPL in Nagoya, Japan.

## 1 Introduction

Our overarching goal in this research and development effort is to advance the capabilities of Toyota Human Support Robot's (HSR) for a successful team performance and technology demonstration at the 2017 RoboCup@Home DSPL. We will achieve this goal by (1) leveraging our team's robotics competition experience from the DARPA Robotics Challenge (DRC), NASA Sample Return Robot (SRR) Centennial Challenge, NASA Exploration Robo-Ops Challenge, and Intelligent Ground Vehicle Competition (IGVC), (2) developing a systematic model-based task validation methodology, (3) implementing novel perception based navigation, manipulation and human-robot interaction techniques, (4) developing novel autonomy techniques for mobile robot manipulation, (5) making the developed software available to the robotics community post competition to broaden the impact of our participation. Successful completion of this project will not only progress the technological readiness of autonomous personal service robots for practical applications but also contribute new knowledge and methods to the RoboCup@Home DSPL community.

Our research at the Robotics and Intelligent Vehicles Research Laboratory (RIVeR Lab) at Northeastern University made research contributions in experimental robotics for disaster response, service and space exploration; human-in-the-loop robot control; and whole-body motion planning and control for humanoid robots. RIVeR Lab led the Team WPI-CMU in the DARPA Robotics


**Fig. 1.** Team Northeastern's robot inventory relevant to RoboCup@Home: (a) NASA'a full-size humanoid robot Valkyrie (b) Mobile manipulation platforms AERO (NASA Sample Return Robot Challenge participant in 2013 and 2014) and Oryx 2.0 (1st Place in the NASA Robo-Ops Challenge 2012), (c) Our personal service robot prototype developed under an award from the National Science Foundation.

Challenge (DRC). Our team has more than 10 years of experience in design, control and validation of reliable robot hardware including mobile manipulation systems and humanoids. Figure 1 depicts a sample of our current robot platforms enabling our research in autonomous navigation, manipulation, human-robot interaction, and perception. Our team has a unique opportunity as we have been selected by NASA to receive a Valkyrie humanoid robot [1]. Our goal is to advance the capabilities of Valkyrie (6' tall, 275 lbs, shown in Figure 1) [1] to perform maintenance and construction tasks in pre-deployment missions to get ready for the manned missions to Mars in 2030s. This platform enables our team to rapidly validate locomotion, manipulation, perception and human-robot interaction algorithms on a complex robot.

## 2 Research Plan for RoboCup@Home DSPL

We claim novelty in three areas towards realizing robot capabilities for Toyota HSR in order to validate daily tasks in support of elderly and individuals with disabilities: (1) We will extend and adapt our human-supervised autonomy framework developed for the DARPA Robotics Challenge to achieve task-level autonomy [2, **?**]. (2) We propose to develop a new method called *compositional autonomy* informed by a risk-aware decision-making mechanism [3]. (3) We will implement a new single-source any-point path planning algorithm called *C-Wave* that results in significant computational improvements in 2D autonomous navigation as the algorithm relies on integer arithmetic only [4].

### 2.1 Supervised Autonomy Framework

Realization of reliable and sufficiently agile autonomous behaviors with Toyota HSR will require the design of a holistic control architecture which is modular and reconfigurable. Based on our prior work [2, **?**], we propose to design and implement a library of capabilities for HSR.

[1] https://www.nasa.gov/press-release/nasa-awards-two-robots-to-university-groups-for-rd-upgrades

Knowledge Base– stores global strategies and approaches for the system, providing options to the action engine. We will explore minimum-parameter and context-agnostic knowledge representations for each use case. The representations will be stored in a reconfigurable database to allow for inferences to be quickly generated from queries. At a minimum, the knowledge base will be aware of the history of the action engine, and provide the high-level goal and the control modality most appropriate given both preprogrammed and learned approaches. We will adopt openEASE [5] and NELL [6]. We will generate a semantic representation of the map stored in the knowledge base so the operator can direct HSR in the home by specifying rooms, for example.

Action Engine– takes the global strategy and plan given by the knowledge base and generates a set of potential actions for the robot to complete the tasks. The action engine has access to the information from the perception engine, which provides filtered state information from the robot. The action engine will be implemented as a state machine. For example, to navigate in an unknown, dynamic environment, a series of dynamically-seeded spirals can be executed to generate the initial map. Moreover, a hierarchical navigation architecture for the HSR will be implemented. Our motion template based manipulation framework [7] will be implemented as part of the action engine.

User Interface– serves as the human interaction interface which can be adapted to various different scenarios and users. Model-based awareness algorithms can selectively change and adjust the feedback to the human. The user will only transmit high-level control commands such as points of interest in the home (kitchen, bathroom, etc.). We will integrate technologies such as 3D mice, game controllers, touch/gesture-enabled screens, and face and voice recognition to provide multi-modal interfaces.

Perception Engine– aggregates all the sensor and state information. By selectively and dynamically filtering the incoming data, the perception engine generates relevant environment models. We will rely on autonomous grasping techniques such as grasp pose detection based-on machine learning to model the local object surface geometries given only point cloud data as input. Perception engine will be unified with the action engine for fast SLAM implementation for navigation in dynamic environments.

Achievable Action Gate– takes the desired actions from the action engine and checks their feasibility. In addition, the achievable action gate fuses human input with the robot input. For example, the action gate can limit the human input that will result in collisions with the environment, both static (e.g. furniture) or dynamic (e.g. people/pets) obstacles, in a navigation task. We plan to integrate these human inputs to the system as constraints on the system model. In order to find a solution that satisfies the task within the constrained model, we need to map the human-inputs to constraints on the planner.

Robot Control– accepts the actions from the action engine and translates them to low-level motion controllers. The robot control implements the algorithms and methods that control the robot to be able to take the set of actions from the action gate and implement the motion planning, localization, trajectory following, and manipulation required for the task. Each control method will adhere to an application specific interface so that the addition of new systems are easily accommodated by the achievable action gate and perception engine.

Cloud Engine– receives limited state information about the robot system through the human robot interface and passes the information to the cloud, harnessing the combination of significant human and computational resources, to provide simulations and model-based awareness and control improvements. For example, a complete model of the robot system as well as human input models can be run in the cloud engine. This is a highly desirable capability and we propose to implement foundational blocks for HSR.

Context Engine– receives information from the human-robot interface to develop context information based on environment models and task descriptions. Context information can be deduced from location, identities of nearby people and objects, time, etc [8]. We will generate the context information using Bayesian inference system for complex interpretation and decision-making by evaluating the performance on a task by task basis. The context engine's results are passed to the knowledge base so they can be recalled to compare previous contexts with the current state of the system as time evolves. The knowledge base, context engine, and cloud engine all operate in close coordination to provide robustness by allowing the system to adjust to changing conditions.

### 2.2 Risk-Aware Compositional Autonomy

We assume that a task can be completed by *composing* a sequence of behaviors selected from a set of feasible actions generated by the motion planned. These actions make up the task-level robot behaviors and the *compositional robot autonomy* can be achieved by stitching these tasks in some order by the high-level mission planner. We posit that it is possible to develop a theory of robot decision-making under uncertainty by introducing measures of risk to robot behaviors with the goal of enabling risk-averse or risk-taking robot autonomy. We will introduce our methodology to evaluate the risk associated with an action composition for completing a given task using a two-link arm shown in Fig. 2 and by taking into account **the collision risk probability** $P_{collision}(\mathcal{A}_i)$ where $\mathcal{A}_i$ are feasible actions for the robot to complete.

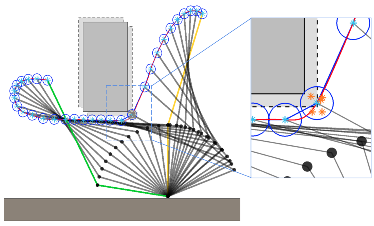

**Fig. 2.** Demonstration of collision risks introduced by sensing and control errors.

The arm needs to move from the right side of the obstacle to the left side. The shape and pose of the obstacle was detected by the robot vision system and provided to the motion planner. The gray rectangular object with solid outline shows the detected result. However, since there are uncertainties the actual object pose is different than the detected one. The light gray rectangular

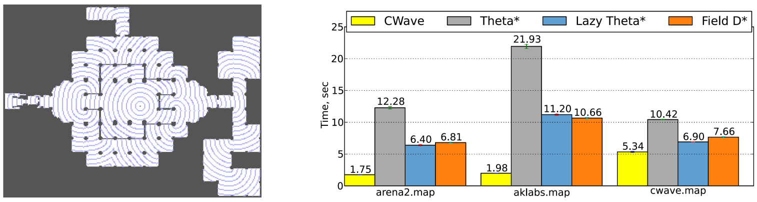


**Fig. 3.** (Left) The solution of single source any angle path planning problem using the CWave algorithm. (Right) Performance of CWave in comparison to existing path planning algorithms.

object with dashed outline depicts a possible actual pose. The solution generated by the motion planner is represented by a set of robot states, (Figure 2). A cubic spline interpolation method is implemented and the end-effector trajectory of the interpolated motion is marked in red. The blue circles represent the region of errors for the end-effector positions.

To model the collision risk, the continuous motion trajectory $Traj(\mathcal{M}_j(\mathcal{A}_i))$ is linearly sampled in joint space to a sequence of waypoints, $\mathbf{q}[n]$, where $\mathbf{q}[n]$ is a set of robot configurations and $n = 0, ..., N$. The result of the optimization-based motion planning algorithm can be directly used due to its property of subdivision. We define the collision risk probability $p_{collision}$ of a robot configuration $\mathbf{q}[n]$ on a given trajectory $Traj(\mathcal{M}_j(\mathcal{A}_i))$ by a parametrized piecewise polynomial function which depends on the shortest distance $d(\mathbf{q}[n])$ between the robot and the obstacles.

$$p_{collision}(d(\mathbf{q}[n])) = \begin{cases} 0 & \text{if } \; d(\mathbf{q}[n]) \geq d_{safety} \\ \left(1 - \frac{d(\mathbf{q}[n])}{d_{safety}}\right)^b & \text{if } \; d(\mathbf{q}[n]) < d_{safety} \end{cases} \tag{1}$$

where $d_{safety}$ is the distance between any point on the robot to the obstacle and $b$ is the degree of the polynomial function, which defines the steepness of the probability curve. We will implement variants of risk-aware compositional autonomy for achieving autonomous navigation and manipulation capabilities for HSR.

### 2.3 CWave Path Planning Algorithm

Conceptually CWave is a wave-propagation algorithm and, in this sense, can be considered a special case of the Fast Marching Method where the interface velocity is constant. It is also similar to a well-known Lee's wave algorithm that deals with octagonal or square waves propagating over an 8- or 4-connected graph, respectively. In case of CWave, however, the wave front has a circular shape to the extent permitted by the grid. The main idea of CWave is that it does not use a graph representation of the grid, and maintains the wave front as a set of discrete geometric primitives (discrete circular arcs and lines), rather than a set of individual points (vertices).


**Fig. 4.** (Left) Detection of candidate grasp poses for object pick-up. (Center) Results demonstrating object and person detection. (Right) FetchResearch platform used in qualification tasks.

In a nutshell, we want to calculate distances from a given point A to all other points in the bounded area. The gradual expansion of a circular wave from the start source point A allows to assign distances to all points which are directly visible from A. Then at every point where the wave meets an obstacle, a new source point can be placed. Simultaneous expansion of circular waves from new sources allows to further assign distances to points in the bounded area. At a certain moment, some of the waves may merge.

We developed the theory and practical considerations that fully support the C-Wave algorithm [4]. We adapted Theta*, Lazy Theta*, Field D* for single-source problems, and incorporated CWave algorithm for speed testing. The framework measures the time required by each algorithm to solve a set of path planning problems on a given map. The results (Fig. 3), demonstrate that on all maps, CWave performed faster than other algorithms. Given that CWave works using only integer arithmetic and bit shifting, it can be ported to low-cost embedded platforms that lack support for floating-point operations, for example, those used in swarm robotics.

## 3 Accomplishments To-Date Towards Qualification

In this section, we describe our methodology in completing the RobboCup@Home DSPL qualifications tasks. As we are still waiting for the arrival of our Toyota HSR platform, we demonstrate these tasks using a FetchResearch robot which has comparable capabilities to HSR.

**Speech Guided Autonomous Navigation.** Our speech-to-navigation module utilizes the CMU Sphinx[2][9] voice recognition library to interpret a user's command. The module is integrated into the ROS Navigation stack to provide autonomous driving to locations of interest. The Sphinx library uses a phonetic dictionary and language model that map words to phones and match recognized words to a known database respectively. We use Sphinx to extract the intention of a user and the desired location they would like the robot to travel to. For our

[2] `cmusphinx.sourceforge.com`

purposes, FetchResearch responds to navigation commands, such as GoTo, and extracts the location from the rest of the sentence. We then use the ROS MoveBase action type to send a navigation goal towards the desired location. In order to do so, we assume that the robot has knowledge of its environment and certain key locations, usually defined by the user. Our initial tests show that Sphinx with the navigation stack provide consistent results and work well with user-defined locations. Since FetchResearch does not have microphone, an external microphone connected to laptop is used. HSR provides a microphone array and will let us implement our speech module without external microphone.

**Manipulation of Unknown Objects.** The Storing Groceries task requires robust grasping of both known and unknown objects. Our grasping approach does not use any prior knowledge about object shape or texture and finds feasible grasps for every type of object. Our approach includes four steps: (1) Segmentation of table and clustering objects on the table to get individual point clouds. (2) Generation of grasp pose candidates for each object and selection of only vertical and horizontal poses. (3) Evaluating selected grasp poses for collision avoidance. (4) Placing objects on shelves.
**Pick and Place Pipeline.** (1) To find object point cloud clusters, firstly, our algorithm finds the table using RANSAC with plane model [10], then, we extract the the point cloud which are on the table with a threshold. Secondly, Euclidean Clustering is applied to extract point cloud to find individual clusters of objects. Finally, Statistical Outlier Removal filter is applied to remove noise. (2) The point cloud clusters of objects are fed to grasp pose detection algorithm. We have used *agile_grasp* ROS package [11] which returns a number of normal vectors of grasp pose candidates. In our trials, we saw that vertical and horizontal grasps are working better so that we only used these grasp poses. The biggest advantage of *agile_grasp* is that it doesn't require any prior knowledge about the objects. In Figure 4, the visualisation of robot, point cloud and normal vectors of grasp pose candidates can be seen. (3) The third step of grasping is evaluating the selected grasp poses with collision avoidance. We have used MoveIt! Motion Planning Frameworkfor calculating inverse kinematics and finding collisions. MoveIt! comes with different planners. After trying different planners and parameters we have seen that RRT-Connect planner [12] with planning time of 5 seconds and 100 attempts works fast and reliable. (4) To place objects on the shelf, a similar and simple approach is implemented. Once a successful grasping is achieved, robot turns it's head towards shelf to get a point cloud, then, planes are segmented to finds shelves. An empty location is then selected and send to robot's planner. If planner returns a feasible trajectory, robot executes the placing. If there is no feasible trajectory, robot then try other shelf. In future, our object detection will be integrated to place similar objects together on the shelf.

**Adaption to Toyota HSR.** Both HSR and Fetch has similar RGB-D sensors so that perception pipeline can be implemented to HSR easily. HSR has a Stereo sensor and we are planning to compare the results between Stereo and RGB-D sensor. We are also planning to use camera-in-hand to improve grasping accuracy. On grasp planning side, we want to try two different motion planning

| **Category** | **Software** |
|---|---|
| Operating System | Ubuntu 14.04 |
| Meta OS | ROS Indigo |
| Perception | Point Cloud Library, Octomap |
| Speech Recognition | CMU Sphinx |
| Face Recognition | OpenCV Face Recognizer |
| Object Detection | Darknet, YOLO |
| Motion Planning | MoveIt! |
| SLAM, Navigation | ROS Karto, AMCL |

**Table 1.** A summary of software and external computing resources.

framework, MoveIt! and TrajOpt. Since HSR is a ROS-enabled robot, implementing both motion planning frameworks is trivial.

**Object Detection and Classification.** In the recent years, deep-learning based object detection and face recognition algorithms showed great progress but they still lack implementation in practical robotics. Open-source and state-of-the-art algorithms will enable better human-robot interaction. As a preliminary work, we have implemented an object detection algorithm called Darknet/YOLO Object detection framework [13]. YOLO is capable of working in real-time and doesn't require high computational resources. Results from this object detection method using robot's RGB camera is depicted in Figure 4,

**Fetch Overview & Adaption to Toyota HSR.** We have developed our algorithms on FetchResearch platform. Fetch is an integrated mobile robot including a mobile base, a 7 DoF back-drivable arm and a head. It's mobile base has 2D laser scanner and IMU sensor for mapping and navigation. The arm has 6kg payload at full extension and 940.5mm length. The head has 2 DoF (pan-tilt) and a RGB-D sensor which provides VGA depth map and RGB image at 30 Hz. Fetch comes with a Intel i5 and 16GB computer. Given the specifications of Fetch, we noted that it has many similarities with HSR on both hardware and software which will make migration to HSR easy.

**Software & External Computing Overview.** The 3rd party software used in our work is listed in Table 1. For Pick and Place, Face Recognition and Object Detection, we have used a desktop computer with 5th Gen i7 Intel CPU, GTX 9080 Graphic Card and 32GB RAM. For navigation, we have used a laptop with 4th Gen i7 Intel CPU, Quadro K2000M 2GB and 8GB RAM. A 5G local network is used between computers and robot.

## 4 Conclusions and future work

In summary, our progress to date is well-aligned with the timeline set by our team at the beginning of the season. We are expecting to receive our HSR in April 2017. We are confident our team will have the time and resources to implement the methods described in the research plan to demonstrate unique robot capabilities at the 2017 RoboCup@Home DSPL in Nagoya, Japan.

## References


1. Nicolaus A. Radford, Philip Strawser, Kimberly Hambuchen, Joshua S. Mehling, William K. Verdeyen, and et. al. Valkyrie: Nasa's first bipedal humanoid robot. *Journal of Field Robotics*, 32(3):397–419, 2015.
2. V. Dimitrov and T. Padır. A shared control architecture for human-in-the-loop robotics applications. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 1089–1094, Aug 2014.
3. X. Long and T. Padir. Compositional autonomy for humanoid robots with risk-aware decision-making. In *(under review)*.
4. D. Sinyukov and T. Padir. Cwave: high-performance single-source any-angle path planning on a grid. In *Robotics and Automation, 2017. Proceedings. ICRA'17. IEEE International Conference on*. IEEE, 2017.
5. M. Beetz, D. Jain, L. Mosenlechner, M. Tenorth, L. Kunze, N. Blodow, and D. Pangercic. Cognition-enabled autonomous robot control for the realization of home chore task intelligence. *Proceedings of the IEEE*, 100(8):2454–2471, Aug 2012.
6. Ni Lao, Tom Mitchell, and William W. Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 529–539, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.
7. Xianchao Long, Murphy Wonsick, Velin Dimitrov, and Taşkın Padır. Task-oriented planning algorithm for humanoid robots based on a foot repositionable inverse kinematics engine. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, under review.
8. B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. *Netwrk. Mag. of Global Internetwkg.*, 8(5):22–32, September 1994.
9. Paul Lamere, Philip Kwok, Evandro Gouvea, Bhiksha Raj, Rita Singh, William Walker, Manfred Warmuth, and Peter Wolf. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5, 2003.
10. Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.
11. Andreas ten Pas and Robert Platt. Using geometry to detect grasp poses in 3d point clouds. In *Int'l Symp. on Robotics Research*, 2015.
12. James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 995–1001. IEEE, 2000.
13. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.