Fetch & Freight: Standard Platforms for Service Robot Applications

Melonee Wise, Michael Ferguson, Derek King, Eric Diehr and David Dymesich
Fetch Robotics Inc.
San Jose CA, USA

Abstract

Since the introduction and widespread adoption of the Robot Operating System (ROS), the mobile manipulation and mobile service robot communities have seen great advances in robot capabilities. However, the lack of affordable and commercially available fully integrated standard platforms remains a major barrier to further and faster advances. The Fetch mobile manipulator is designed to be the affordable standard platform for the next generation of mobile manipulator applications. Fetch's little brother, Freight, is designed to be the affordable standard platform for mobile service robot applications. This paper highlights the design decisions and trade-offs made in achieving the low cost of the platforms while continuing to provide the required capabilities for such applications.

1 INTRODUCTION

The Fetch mobile manipulator is designed to be robust, high performance, and low-cost. Through careful design decisions, leverage of commodity components, and building on the lessons learned by our team while working with world class robots, such as the Willow Garage PR2, we have developed a mobile manipulator that is both ready for commercial applications and available for research and development. Freight is the lower half a Fetch robot, utilizing the same drive and computation components. Freight also features a set of extensible mount and power points that allows the robot to be configured for a wide variety of tasks.

Our team has had significant experience in developing mobile manipulators. Members of our team were involved with the design of the Willow Garage PR2, Willow Garage PlatformBot, Unbounded Robotics UBR-1, and other designs.

When setting out to create Fetch and Freight, the team developed a series of extensive design requirements that helped to guide important decisions. At a high level, these included:

Figure 1: Fetch and Freight.

- Mobility to traverse ADA-compliant buildings. Specific attention was paid to the door threshold, elevator gap, and ramp requirements.
- Manipulation space suitable for normal human work environments. While Fetch Robotics is primarily interested in shelf picking, the robot workspace was also designed to work well in homes and labs. The arm was designed specifically to be able to reach items on the ground, enabling recovery if an item was dropped during manipulation.
- A sensor suite suitable for the perception of objects, navigation, and manipulation in dynamic environments.
- Sufficient battery power to work an 8-hour day. This requirement affects both the battery selection as well as other power-related trade-offs such as choice of computer processor or communication buses.

As highlighted in figure 3, the end products are a mobile manipulator consisting of a differential drive mobile base, an arm with 7 degrees of freedom and 6kg payload, a pan and tilt head, a torso lift actuator, and a standalone mobile robot platform. The mobile base includes a SICK laser scanner with a 220 degree field of view and 25 meter range. Freight also includes a base-mounted 3D camera. Fetch includes a head-mounted Primesense

Carmine 1.09 depth camera. The gripper is a modularity point, allowing custom grippers to be swapped in, but supplied with a default parallel-jaw gripper capable of grasping a wide range of objects. Intel-based computers provide processing power for navigation, manipulation and perception activities, while extensive battery capacity gives each robot an 8- to 10-hour runtime.

2 RELATED WORK

In recent years, there has been significant interest in mobile manipulators. Likely the best known mobile manipulator is the PR2 robot from Willow Garage [1]. The PR2, built originally in 2010, has two arms, an omni-directional mobile base, and a multitude of sensors. While a formidable platform, the PR2 was heavy (about 450 lbs), slow (0.6m/s), and had a short runtime (only two hours). The PR2 was based on the earlier PR1 robot [2] which, like our robot, had a differential drive base. Significant research has been done on PR2 robots, and there are numerous published results [3], [4], [5], many of which can be reproduced in labs around the world.

A number of research groups have developed notable custom mobile manipulators for their research. The Stanford AI Lab built the STAIR bot, which coupled an off-the-shelf Katana 5 degree-of-freedom arm with a Segway RMP [6]. Jain et Al [7] have created El-E, a mobile manipulator designed for assistive tasks and built from the same arm atop of a different off-the-shelf differential drive base. More recently, the Herb 2.0 platform has been created with a pair of Barrett WAM arms and a mobile base [8]. While these robots generally serve the needs of their labs, they lead to an inability to collaborate with other labs directly on identical platforms. Further, each lab has spent considerable time solving integration problems, and cost savings from volume production are not possible.

3 DESIGN

The following sections describe the design of the robot, including discussion of the choices made with certain design trade-offs.

3.1 Mechanical Design

The mobile base is driven by two brushless hub motors in a differential drive configuration. Each hub motor is held in a drop suspension configuration, allowing the robot to keep traction when crossing obstacles, without allowing the robot to sway side-to-side when manipulating on flat ground. Four casters provide stability during movement. The base is designed to work in environments that are compliant with the Americans with Disabilities Act (ADA). This primarily entails being able to climb an incline of no more than 1:12, drive over cracks of up to 1 1/4", and navigate door thresholds up to 3/4" tall. Additionally, the base fits within a circle of 22" (0.5588m) diameter. This allows the robot to turn in place within the smallest of typical door sizes (26") even with localization errors.

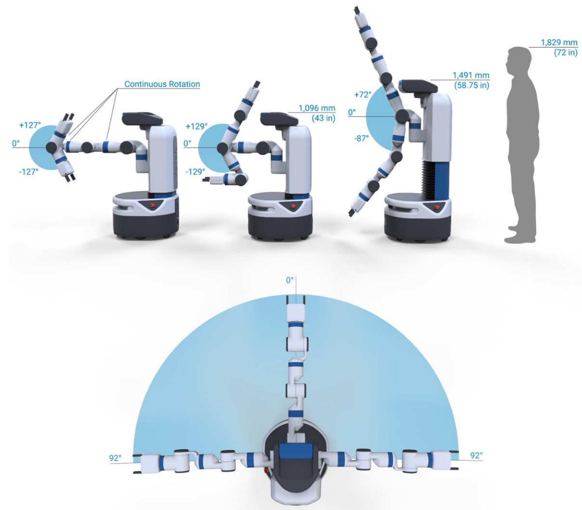

Figure 2: Kinematic configuration and reachability of the Fetch arm. The upper arm roll, forearm roll, and wrist roll joints have continuous rotation. The shoulder pan, elbow flex and wrist flex joints have symmetric limits.

Fetch is equipped with a single 7 degree-of-freedom arm which supports up to a 6kg payload, including the gripper. Kinematic optimization was important for maximizing the workspace of the arm. As can be seen in figure 2 the wrist flex, elbow flex and shoulder pan joints have symmetric workspaces. Each of the upper arm roll, forearm roll, and wrist roll joints are capable of continuous rotation, allowing complex maneuvers with the arm. The shoulder lift joint is capable of significant upward motion, giving the robot a higher reach than the taller PR2 robot. The arm is designed to be light weight with a 12.63kg swinging mass. In addition, the top speed of the end effector is limited to 1.0m/s. While a second arm might be useful for a limited number of applications, it would also have added significant weight and size to the robot, likely increasing the base dimensions. For the few tasks requiring bimanual manipulation being performed by mobile manipulators today, two Fetch mobile manipulators could work together.

Each of the Fetch arm joints are built from a harmonic drive coupled to a brushless frameless motor, mounted inside a custom cast aluminum housing. Each joint has two 14-bit absolute magnetic encoders, one coupled to the joint output shaft, and one coupled to the motor backshaft. Empirical testing has shown that the encoders have roughly 1.5 bits of noise when the arm is holding position, providing an absolute accuracy of about 0.001 radians. The use of harmonic drives with joint side encoders on Fetch is a large improvement over the belt driven mechanisms with motor side encoders found in the PR2 whose accuracy changes with belt

Figure 3: Fetch Hardware Feature Overview

stretch over the life of the PR2.

The absolute magnetic encoder directly coupled to the output shaft allows the robot to know exactly where the robot joints are when the robot is powered on, avoiding the need for a potential dangerous power-on movement to pass through an optical flag. During assembly, the zero position of the joint is recorded. Through the use of a highly robust calibration system, we were able to avoid the use of calibration pins or fixtures for the zeroing, as technicians are able to simply zero the joint by eye and let the system calibration software calculate the precision adjustment.

Gravity compensation is done in software, unlike the mechanical gravity counterbalance of the PR1 or PR2 robot. While the counterbalance reduces the static power draw of the arm holding position, it adds significant weight to the upper half of robot, which translates into even more ballast being required in the robot base. Static power draw to hold the arm out with a full 6kg payload is less than 35 Watts, and under 20 Watts when holding full payload in a tucked configuration.

The arm kinematics are designed such that the arm can be tucked fully within the robot base. When combined with a circular footprint, this allows the robot to turn in-place. Planning for a circular footprint robot also offers significant runtime gains over more complex footprints.

All of these components are then put into a beautifully designed robot package. The application of great industrial design is an essential part of robot design. A well designed robot hides away potentially unsafe mechanisms and is more acceptable for HRI studies. Fetch has already won a Spark design award for great design.

3.2 Electrical Design

Each motor has a dedicated motor controller board based on an STM32. This microcontroller runs a 17kHz effort controller on the brushless motor as well as real-time 1kHz PIV loops for velocity or position control of the actuator.

Whereas a number of robots use expensive internal buses like EtherCat, Fetch and Freight use primarily RS-485. While RS-485 does not intrinsically provide the strict real-time characteristics of EtherCat, it is both less expensive to implement as well as significantly lower power than EtherCat. Within a robot like Fetch, adding EtherCat could consume 10s of Watts of additional power just for the Ethernet interfaces. Real-time performance of the RS-485 then comes down to proper implementation of communications libraries in the microcontrollers.

There are two RS-485 buses, one for the arm and the other for the base, torso and head. Each RS-485 bus is designed to handle timing requirements for either 500Hz control of position and velocity and effort, or up to 1kHz control of effort alone. Each bus is routed back to a central "mainboard" which then connects the RS-485 buses to an Ethernet connection which connects to the robot computer. Charging control is implemented on the mainboard. The connectivity of the various data buses is shown in figure 4.

If price is the biggest drawback to robots such as the PR2, power would be a close second. Many robots list a 2-hour runtime, not enough for a full day of work. Fetch and Freight use two Sealed Lead Acid (SLA) batteries which provide 8-10 hours of continuous use. The batteries are in series, which provides 24V, a good trade-off

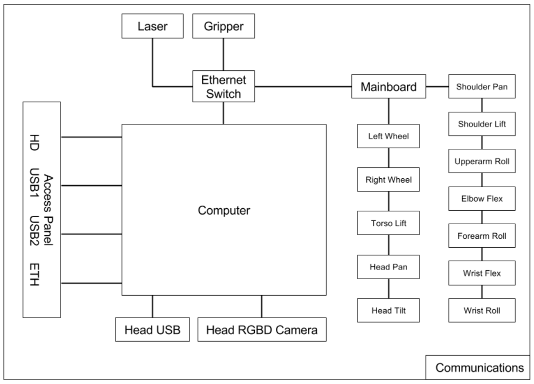

Figure 4: Routing diagram of communications buses within the Fetch mobile manipulator. Shown are the RS-485, Ethernet, and USB connections.

between a higher voltage which is difficult to work with and a lower voltage which requires larger cabling.

Fetch and Freight have a number of electronic breakers, designed to turn off power to various subcomponents. Each RS-485 bus has an associated breaker. An additional breaker is configured on the battery power cable and another on the charger supply inlet. Each breaker provides current measurements which are available through ROS diagnostics.

Fetch and Freight are configured with a single Intel-based computer. On Fetch this consists of an Intel i5 processor, 16 gigabytes of RAM, wireless card and a solid state drive. Freight uses the same motherboard configuration but with an Intel i3 and 8GB of RAM since the robot does not have the overhead of a high-resolution RGBD camera or arm motion planning software. The computer runs Ubuntu Linux, providing a familiar environment for users of ROS. By moving the real-time control loops to the motor controller board microcontrollers, the computer can run a non-real-time kernel, greatly reducing the issues users encounter when attaching new hardware. A secondary improvement is that the lack of a real time kernel allows use of SpeedStep technology which reduces the power consumption of the processors when not under full load. The computer uses a standard Mini-ITX motherboard, ensuring that the computer can be repaired or upgraded, and cutting costs over a custom form factor. The motherboard and other components are installed in a custom sheet metal enclosure which is installed in the back of the mobile base section of the robot.

Modularity is an important consideration in the design of any robot, especially considering how fast sensors and other technology develops. An interface panel on the side of the robot exposes dual USB ports, an Ethernet jack and HD video interface, allowing for easy connectivity of external peripherals to the robot. Additional USB ports are located on the head of the Fetch robot. Both robots feature mount patterns located on the base of the robot, and Fetch has additional mount patterns on

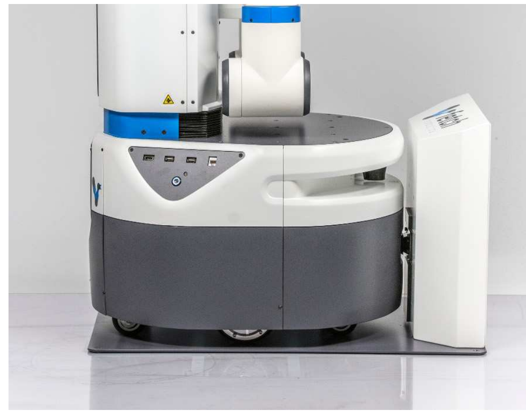

Figure 5: Fetch on charging dock.

the top of the head pan and head tilt stages and the back of the gripper. In addition, the gripper itself is modular, the mechanical mount having been based on an ISO standard and the electrical interface being a simple 24V power and Ethernet communications.

Finally, as continuous operations are important, the robot was designed to interface with a charge dock. The charge dock uses a floating blind mate connector so that Fetch and Freight can autonomously dock with the charging dock. The frontal shape of the dock was specifically designed to be highly visible and unique in the laser range finder data so that the robot can locate the dock autonomously using only the range finder.

3.3 Sensor Selection

The mobile base contains a 2D scanning laser range finder. This is a 25 meter range version from the SICK TIM family. With the wide cutouts in the base skins, the sensor is able to see a full 220 degrees field of view. The long range and wide field of view make this an ideal sensor for localization in nearly any environment using the amcl ROS package [9].

We originally had planned to only deploy Freight robots in conjunction with Fetch robots, however, we quickly found numerous applications that required only Freight robots. As such, we found there were numerous environments in which a 2D laser scanner alone was insufficient. Freight robots now include a 3D camera mounted in the base of the robot which is used for added obstacle avoidance capabilities.

The head of the robot contains a Primesense Carmine 1.09 short range depth camera. This sensor offers VGA resolution depth and color images at up to 30fps. The short range version of the sensor is factory calibrated for the 0.35 meter to 1.4 meter range, however it returns data out to 4 or 5 meters depending on the finish of the surface being measured. Our own full-system calibration improves the calibration in this 0.35 meter to 1.4 meter range. This calibrated range is ideal for manipulation, and sufficient for navigation. In practice, we use data up to 2 meters away for navigation.

In addition to ranging sensors, the base of Fetch contains an IMU consisting of a 3-axis accelerometer and 3-axis gyro. We use the gyro in combination with an Unscented Kalman Filter (UKF) to improve the base odometry. A second IMU is located in the gripper.

Finally, each joint uses the current measurements of the brushless motor controller to estimate the effort being applied by the arm.

4 APPLICATIONS

Fetch is built to work with the Robot Operating System (ROS) [10]. By using a standard software platform, a large number of developers are already familiar with the tool set and can quickly start working with the robot. In the following sections we describe some of the applications deployed on Fetch and Freight.

All of the applications shipped on the Fetch Research Edition, with the exception of firmware and some low-level drivers, are open sourced under permissive licenses. This allows developers to continue improving the autonomous capabilities of the robot, and sharing their contributions with the developing community.

4.1 Navigation

The research editions of Fetch and Freight use the ROS Navigation stack. We use the standard planners and costmaps, having contributed several patches and improvements back to the mainline code base. Both the base laser and the 3D cameras are used as obstacle sensors for updating the costmaps. In addition to the standard components, we have added several new components.

First, a new recovery behavior has been added that controls the robot head. This module uses the head pan and tilt stage to "look around" and try to clear obstacles when the robot is unable to find a plan to the goal. In addition, when the robot is navigating, this module tilts the head up and down to get a taller field of view for the head camera, and points the head towards where the robot will be in several seconds of travel. This last addition makes a marked improvement in navigation, primarily through social engineering, as humans move out of the path of the robot since they know where it will be going.

Another improvement is the costmap updater within the fetch_depth_layer ROS package. This module is designed to handle the unreliable timing of the head camera. Instead of applying a fixed transform to the data, the module detects a ground plane in the depth image, using an algorithm similar to [11]. Once the ground plane has been found, we can adjust the timestamp on the depth image based on the more reliable tilt motor timestamps.

4.2 Manipulation

Fetch Research Edition uses the MoveIt! manipulation package [12]. MoveIt! provides planners, collision checkers, trajectory smoothers, and more in a modular package. For our commercial development efforts, we have largely replaced the standard planners and collision checkers with faster or more robust components, however, they still make a great baseline for the research and development community.

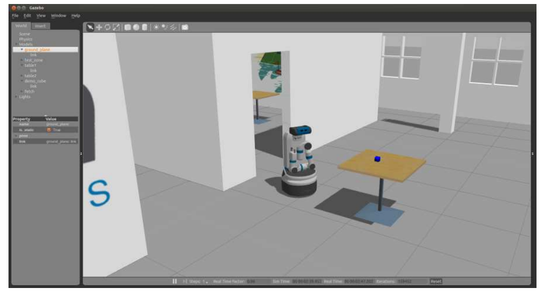

Figure 6: Fetch in simulated environment.

One aspect of manipulation not provided by MoveIt! is perception and grasp planning. For a great out-of-the-box experience, we developed a ROS package called simple_grasping, which includes two components: a basic tabletop perception system capable of segmenting the world into graspable objects and support surfaces to place them on. Our approach to grasp evaluation is similar to [13], however, we avoid all attempts at modeling objects so that the code is more easily deployed in numerous labs.

4.3 Calibration

With any mobile manipulation platform calibration of sensors and end effectors is essential to proper operation. Fetch uses a calibration system that comprises the use of an LED pattern on the gripper. The robot can then re-project the samples taken by the head camera to the estimated location through the kinematic chain of the arm, similar to the approach used in [14]. With a given set of samples a nonlinear least squares minimization is computed by ceres-solver [15]. This approach has a number of improvements over the calibration procedure previously used on the PR2, most notably, because the calibration target is always on the robot, there is no need to give the robot a checkerboard calibration target. This means that the robot could recalibrate itself at any time in the field.

4.4 Simulation

In addition to the physical robot, we have developed a simulated version of both Fetch and Freight using the Gazebo simulator [16]. The simulator allows off-robot testing, as well as allowing additional users to leverage a single robot in a lab. We have developed a standard test environment in Gazebo for the robot, as seen in figure 6. This environment includes a mobile manipulation demo in which the robot navigates to a table, picks an object, and then navigates to a different room to place the item.

5 Commercial Applications

A number of interesting applications are possible when pairing Fetch and Freight. In particular, we are interested in teaming the robots within the warehousing and logistics space. Here, Fetch can pick items from shelves, while Freight robots carry the picked items and are able to quickly shuttle the items around a warehouse.

Having started with a ROS-based system, which is already permissively licensed, we have been able to incrementally replace components of the system one at a time with more robust solutions. Complete coverage of our commercial applications are beyond the scope of this paper, but more details can be found on our website at http://fetchrobotics.com.

6 TESTING

A critical part of the success of any complex robotic platform is system reliability. In addition to the standard burn-in testing performed on each robot, we have created a logging system which collects data from robots during normal operation as well as during special tests. During normal operation, the drivers collect usage information such as joint travel, temperatures, currents and voltages and relay this information periodically to the logging server.

We operate a test warehouse which runs robots autonomously in a 24/7 environment through a variety of tests. In particular, our fleet of Freight robots each travel an average of 30Km per day in the test warehouse, and have collectively traveled thousands of kilometers and autonomously docked and recharged thousands of times. This type of testing is one of the aspects that sets Fetch and Freight apart from many of the other platforms available to the robotics research community.

References

[1] http://www.willowgarage.com/pages/pr2/overview.

[2] K. Wyrobek, E. Berger, H. V. der Loos, and K. Salisbury, "Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008.

[3] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. Rusu, B. Marthi, G. Bradski, K. Konolige, B. Gerkey, and E. Berger, "Autonomous door opening and plugging in with a personal robot," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 729–736.

[4] J. Bohren, R. Rusu, E. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, W. Meeussen, and S. Holzer, "Towards autonomous robotic butlers: Lessons learned with the pr2," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011.

[5] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *International Conference on Robotics and Automation*, 2010.

[6] M. Quigley, E. Berger, A. Y. Ng *et al.*, "Stair: Hardware and software architecture," 2007.

[7] A. Jain and C. C. Kemp, "El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots*, vol. 28, no. 1, pp. 45–64, 2010.

[8] S. S. Srinivasa, D. Berenson, M. Cakmak, A. Collet, M. R. Dogar, A. D. Dragan, R. Knepper, T. Niemueller, K. Strabala, M. Vande Weghe *et al.*, "Herb 2.0: Lessons learned from developing a mobile manipulator for the home," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2410–2428, 2012.

[9] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial intelligence*, vol. 128, no. 1, pp. 99–141, 2001.

[10] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source robot operating system," in *Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA)*, 2009.

[11] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3d range images," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008.

[12] I. A. Sucan and S. Chitta, "Moveit!" [Online] Available: http://moveit.ros.org.

[13] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones, "Contact-reactive grasping of objects with partial shape information," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.

[14] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *International Symposium on Experimental Robotics (ISER)*, 2010.

[15] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[16] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004.