Robot Tracking of Human Subjects in Field Environments

Jeffrey Graham, Titan, jgraham@titan.com
Kimberly Shillcutt, NASA/JSC, kimberly.j.shillcutt1@jsc.nasa.gov

Abstract

Future planetary exploration will involve both humans and robots. Understanding and improving their interaction is a main focus of research in the Intelligent Systems Branch at NASA's Johnson Space Center. By teaming intelligent robots with astronauts on surface extra-vehicular activities (EVAs), safety and productivity can be improved. The EVA Robotic Assistant (ERA) project was established to study the issues of human-robot teams, to develop a testbed robot to assist space-suited humans in exploration tasks, and to experimentally determine the effectiveness of an EVA assistant robot [2].

A companion paper [5] discusses the ERA project in general, its history starting with ASRO (Astronaut-Rover project), and the results of recent field tests in Arizona. This paper focuses on one aspect of the research, robot tracking, in greater detail: the software architecture and algorithms. The ERA robot is capable of moving towards and/or continuously following mobile or stationary targets or sequences of targets. The contributions made by this research include how the low-level pose data is assembled, normalized and communicated, how the tracking algorithm was generalized and implemented, and qualitative performance reports from recent field tests.

Introduction

Robots are necessary for planetary EVAs. Astronauts on EVA operations must overcome severe restrictions imposed by the spacesuit and work environment. A pressurized suit encumbers the astronaut's movements and significantly restricts range of motion. The gloves severely degrade manual dexterity and tactile sensing, while the helmet drastically limits field-of-view and audition. EVAs are normally planned for teams of two to provide redundancy, contingency, and improved situational awareness [21]. Robots can provide these and other capabilities. On a planetary surface where gravity is an additional burden, the robot's ability to act autonomously while carrying heavy equipment may extend an EVA to enable complex tasks. Finally and cumulatively, the ability to function in place of humans in risky/off-nominal/emergency situations makes robots indispensable for planetary EVAs.

With human and robot mobile agents operating in the same workspace, safety requires pose awareness. Safety cannot be guaranteed and risk may increase if the agents are unaware of each other's location and attitude. Agent pose data must be high quality (fresh, continuous, confident), and distributed in timely fashion. Safety demands that the sensors, architecture, and algorithms concerning the pose data be robust in the face of noise, delay or loss.

Tracking plays a key role in design reference missions for surface exploration. Tracking is a necessary core functionality for intelligent mobile robots that depends on high-quality pose information. The Mars Reference Mission [20] allows for some form of high-quality surface localization, probably similar to GPS, possibly using a mini-constellation of satellites, ground-based pseudo-lites, or beacon-based navigation. ERA's tracking architecture makes use of pose data from a variety of sensors including GPS. Likely scenarios for human-robot teams on a planetary surface require robot tracking: navigating great distances, interacting and cooperating with other agents performing mapping, construction, maintenance and science tasks.

Tracking Architecture

ERA provides the required tracking functionality. ERA research has resulted in the development of an architecture and generic tracking algorithm for guiding an actuator (mobile robot base, arm, camera head) relative to a moving or stationary target (human, robot, waypoint) or sequence of targets, while maintaining a variety of pose-specific constraints (separation distance, relative pose, etc.). The focus of this paper concerns moving an entire robot using its base drive actuators (wheels) to track a human as he moves about.

Several pieces of the architecture enable tracking. The ERA architecture is distributed across multiple CPUs, processes, and threads. There are approximately thirty servers running on 5 CPUs on the ERA, each providing a single, highly cohesive, loosely coupled piece of the architecture. The servers communicate exclusively through various CORBA structures and methods defined in their interface description language.

For tracking, there are several key servers. Each raw pose sensor (Laser, differential GPS, inertial measurement unit with compass, stereo vision), located on the robot, spacesuit, or elsewhere externally, is driven by a dedicated server. These servers are responsible for enforcing a pose update policy that filters/massages the raw data, identifies significant changes in pose, and pushes those changes to a centralized clearing house called the pose server (PS). See Figure 1.

Laser Target Server

It is a very good sensor. Tracking people based on laser data is becoming ever more popular because of its accuracy and fast data processing, especially compared to vision [7][1][12]. The ERA has a Sick LMS-200 laser on-board, mounted about 1 meter above ground and pointing straight out. It provides either 100 or 180° scan of measurements at angular resolution from 0.25 to 1.0 degrees.

Filtering is critical to performance. A scan line from the laser requires quite a bit of filtering to robustly identify the desired target, perhaps a human standing somewhere forward of the robot. Adjacent points are grouped into segments, the

segments outside a predefined 2D box are removed, wrong-sized segments are removed (a human can be expected to have a certain width), and the surviving segments are sorted by their distance. The nearest target is selected as the prime target. Similar filtering has been done by others [13].

This is not Star Trek. Filtering is used to minimize the effect of "teleporting" targets-- apparent targets that change location suddenly without continuity. This occurs when a new object enters the laser's field of view nearer than the old target, or when an old object leaves the field of view or becomes occluded. Related work addresses this difficulty [4]. For this filter, the prime target's position is again compared with the last known position. If the new position differs significantly, the new position is not reported to the PS, and the history is not updated until a target (hopefully the original target) reappears close to its previous position.

Seeing is believing. A big disadvantage of the laser is its requirement that the target is within range and line of sight. On hilly terrain, a target can seem to disappear as it scales a steep slope far ahead of the robot. The laser lens is sensitive to dust and specular reflections caused by precipitation. A final disadvantage is the difficulty in target discrimination.

Light and fast. The laser is much less dependent on lighting conditions than stereo vision with conventional cameras, and requires much less processing. The laser does not require calibration, and its measurements are extremely precise. Laser tracking is a relative position tracker, and does not suffer the need for an accurate heading value for the robot as GPS does.

GPS Server
Five meter accuracy is not good enough. Tracking using GPS introduces a unique set of advantages and disadvantages. Research in this area has not received much attention even though GPS is used frequently in localization and map registration and in commercial systems [14]. The ERA robot and spacesuit backpack are each equipped with differential GPS, and a fixed base-station tripod wirelessly transmits correction data to them that improves position accuracy to 2cm. This level of accuracy is necessary since the robot and astronaut are typically within 2m of each other.

Which way am I facing? A robot must accurately know it's heading to track with GPS data; heading error from GPS is unreliable over short distances. Without accurate robot heading, the tracking server (TS) will not be able to determine the proper direction for the robot to turn. Two methods for obtaining a heading were used in this research: a Crossbow heading and attitude sensor, and an algorithm which computes heading based on past GPS positions. Difficulties with each of these methods will be discussed in the field test section.

GPS is superior in many ways. Laser and Vision target tracking have the disadvantage of require line-of-sight With GPS, the target object can be occluded or out of sight at great distances from the tracking sensor. Target acquisition and discrimination issues are non-existent, since each target has its unique GPS unit to identify them. Unlike stereo vision, there are no calibration requirements, lighting dependencies, nor lenses to protect from dust or sunlight. Filtering GPS data reduces to simply determining when a significant change has occurred in target position or a timer has expired before reporting updated pose to the pose server.

Nothing is free. The downside of GPS is that each target and the base station must carry and power its own GPS receiver, correction radio, and antennas. Finally, differential GPS is expensive.

Stereo Vision Server
It's been done before. Computer vision has been the primary means for identifying and tracking humans in much prior and current research [9,10,8,3,6]. The ERA project itself has used stereo vision quite successfully in past field tests to identify and localize a human standing somewhere in front of the robot and to guide the robot to follow along. The ERA stereo vision tracking algorithm uses texture to identify regions kinematically approximating the human form, and commands a pan-tilt head to keep the subject within its field of view. These field tests were conducted without the benefit of the new modular tracking architecture.

Not this time. Just recently has the stereo vision server been updated to send its target pose information to the pose server, so field test results using it are forthcoming. Past field tests using stereo vision tracking identified the need to correct its pose information relative to the robot's pose, and algorithms effecting this requirement have been implemented.

A picture says a lot. One benefit of stereo vision tracking is that it provides a full frame of raw data, unlike the laser's single scan line or single GPS point. The abundance of raw data can be used to distill additional target features, pose and target gestures. Thus, target discrimination can be better than with laser tracking. In fashion similar to laser and gps processing, aggressive filtering is done to prevent overloading the pose server with trivial updates. As with laser tracking, stereo vision tracking does not require additional hardware on the target as GPS does.

3D vision is not robust. Stereo vision requires very precise calibration, and errors are unforgiving. Processing time and memory use can be extensive, and changes in field lighting (clouds, glare) may invalidate vision data or make it unusable. The target can be occluded, and background clutter can reduce preciseness. Distance data from Stereo Vision is not as precise as from laser or GPS, and target discrimination is imperfect.

Pose Server
The pose server (PS) is keeper of the data. The PS accepts pose data and normalizes it into a standard world coordinate system such as Universal Transverse Mercator projection (for compatibility with the GPS hardware). It maintains a brief history list of the data, organized and sorted by sensor type, confidence and time stamp. When a new pose is submitted,

the history list is improved and an automatic recalculation is triggered to update two standard products served out by the PS: a single "best" pose for a target, and a "fused" best pose for a target (for now these are equivalent, though a fusing algorithm is planned for the next baseline). During recalculation, all pose data is aged in consideration of it's original quality, confidence, and useful life. To help manage the data volume, the size of the history list can be limited.

The PS makes pose data available to clients. It offers to minimize the messaging overhead to the TS concerning the two required poses (one for the actuator base and one for the target) via several "fat" methods that return various forms of a relative delta pose (what is needed by the TS) given the names of the target and tracking objects. The pose server effectively compartmentalizes the burden and complexity of managing the volume of pose data, allowing the tracking server to concentrate on its sole purpose. For efficient communications, the PS and TS are run on the same CPU.
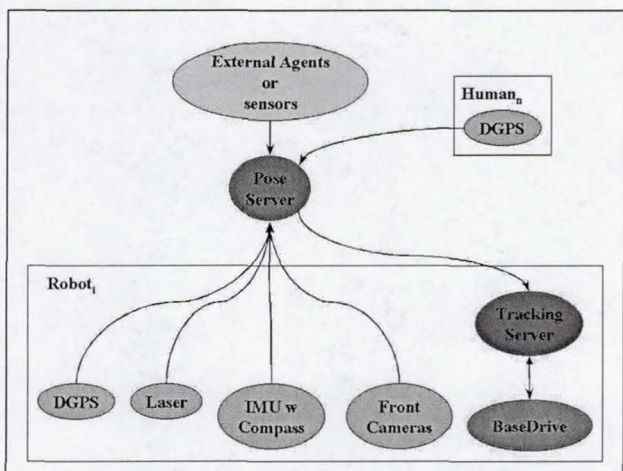

Figure 1 Servers involved in tracking

Tracking Server

The tracking server makes it happen. When a tracking request is made, a "tracking package" containing data about the target is passed to the TS. The package includes the target's name (perhaps the name of an astronaut, robot, GPS waypoint, or some otherwise labeled location such as "habitat" or "home"), the particular sensor to be used for the tracking (can be "best" or "any"), and a confidence threshold below which tracking will pause.

Many more parameters affect tracking. The client can also specify a separation distance to maintain, a dead band, a time limit to reach the desired separation, an amount of time to hold once separation is achieved (useful for tracking a sequence of waypoints), a flag indicating if continuous tracking is requested (useful for tracking moving targets), and finally a flag indicating if temporary target loss is acceptable (useful if a specific tracking sensor might experience transient losses). The Tracking server accepts a sequence of one or more tracking packages, allowing a coordinated traverse between targets.

The Tracking Server is nothing more than a thread manager. When a request is accepted by the TS, a new processing thread is spawned to manage it and its tracking package. Each thread runs a copy of the generic tracking algorithm (GTA). When processing completes, times out, or otherwise aborts, the actuator is safed and the thread is reclaimed. The tracking server itself is nothing more than a manager of threads running the GTA.

Generic Tracking Algorithm

Tracking is homing, for now. The GTA currently only supports homing mode, i.e., the actuator continuously recalculates a straight-line path to the target and dynamically adjusts the actuator appropriately. In many cases this mode is preferred and optimal. A new parameter to support target path following in the GTA is planned for next baseline; this mode might be preferable for rough terrain where the target is choosing a safe path (as an astronaut might on a surface EVA).

It's called generic because, regardless of the actuator being using or the target being tracked, it is the same algorithm. Adding a new actuator for use by the GTA requires only that (1) the actuator maintain its pose information in the PS, (2) the actuator interface has methods to start, stop, pause, and move relative, and (3) the actuator registers with the tracking server by adding an enum. The algorithm itself is unchanged; any specialized actuator needs are handled in the above methods. It is generic because it is modular.
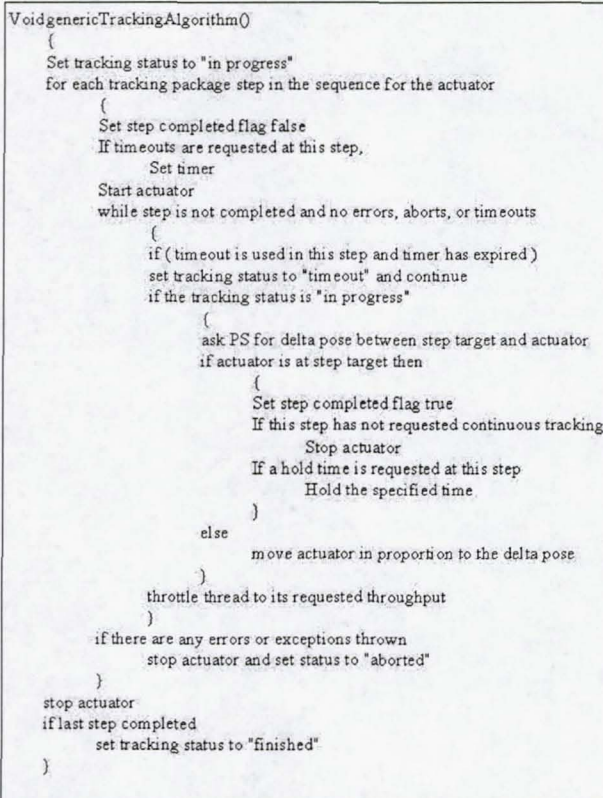
```
Void genericTrackingAlgorithm()
    {
    Set tracking status to "in progress"
    for each tracking package step in the sequence for the actuator
        {
        Set step completed flag false
        If timeouts are requested at this step,
            Set timer
        Start actuator
        while step is not completed and no errors, aborts, or timeouts
            {
            if ( timeout is used in this step and timer has expired )
            set tracking status to "timeout" and continue
            if the tracking status is "in progress"
                {
                ask PS for delta pose between step target and actuator
                if actuator is at step target then
                    {
                    Set step completed flag true
                    If this step has not requested continuous tracking
                        Stop actuator
                    If a hold time is requested at this step
                        Hold the specified time
                    }
                else
                    move actuator in proportion to the delta pose
                }
            throttle thread to its requested throughput
            }
        if there are any errors or exceptions thrown
            stop actuator and set status to "aborted"
        }
    stop actuator
    if last step completed
        set tracking status to "finished"
    }
```

Figure 2 Generic Tracking Algorithm

Thanks to the modular architecture, the GTA is short and sweet. With actuator details abstracted away and the PS assuring high-quality data, there is little actual code to the GTA, so its pseudocode can be presented in its entirety. See Figure 2.

It's been tested in the field. The ERA robot with its described compliment of pose instruments, architecture, and generic tracking algorithm has been tested in the field with promising results.

Field Tests
Déjà vu. Multiple field tests have been performed in various locations over the past four years. Additional information about these tests can be found elsewhere [5], but this section will concentrate on tracking tests performed in September of 2002 near Meteor Crater in Arizona and tests performed since then in the rock yard at Johnson Space Center in Houston, Texas.

Follow me. The basic tracking scenario concentrated on following a human in a spacesuit with a mobile robot, using position and pose data from the variety of sensors mentioned earlier, passed through to the pose server and used by the tracking server in its generic tracking algorithm.

The tracking scenario was conducted over varying terrain, flat versus sloped, sandy versus rocky, daytime versus night time. The variations exposed the strengths and weaknesses of the pose sensors.
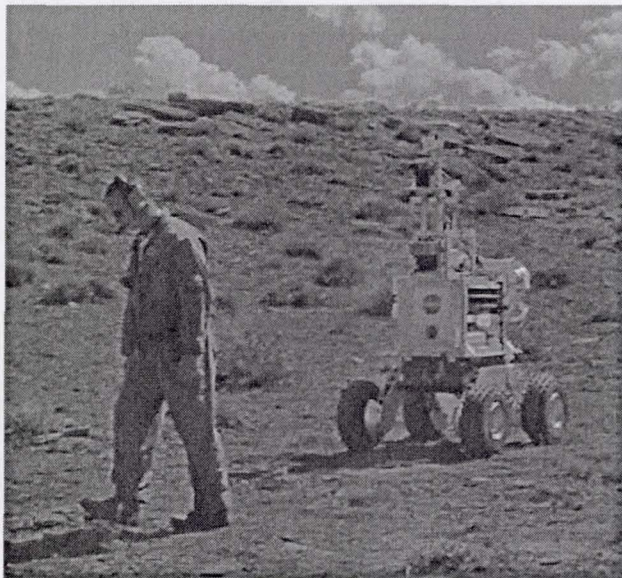

Figure 3 Laser Tracking a Human

Laser Tracking field results
Just one problem. In Arizona, laser tracking worked very well with one exception—the hill descent problem. When the robot was atop a ridge line and the astronaut began his descent of the hill, the combination of separation distance, mounting angle of the laser, and grade of slope resulted in the robot loosing sight of the astronaut, with his head going below the laser's scan line of sight, before he got far enough away for the separation threshold to dictate forward motion. Manual target reacquisition part way down the slope was required before autonomous tracking could resume. Reducing the separation threshold when there is an expectation or active sensing of hilly terrain mitigates this problem.

Figure 3 shows a human test subject being tracked using laser data. The desired separation distance was normally set at 2.0m, with a dead band of 0.5m. This kept the robot close enough to the human test subject for him to know the robot was following, but far enough away to keep from crowding the human. As an interesting side note, to confirm the robot was indeed tracking, the human test subject would typically walk towards the robot to make sure it would back up, and then continue forward on the path.
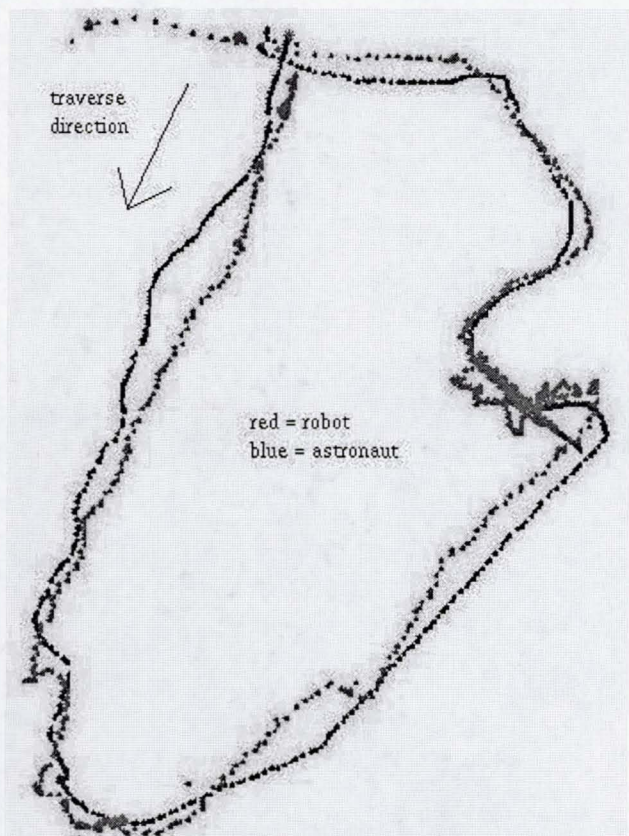

Figure 4 Ground track of robot using laser to follow human

Figure 4 shows the ground tracks of both the robot and human as he walked a pre-defined traverse and the robot tracked him using laser data. Both the human and the robot started in the upper center of the image, and proceeded along the left-hand side of the image. The bottom portion of the image is where the human started walking up a hill onto the ridge line about 5 meters higher than the plain. Then the section on the right side, slightly above the middle of the image, is where the human and robot descended from the ridge line, with both the human and the robot going back and

forth several times before the robot made it down the hill. In most cases throughout the traverse, the robot followed the human quite well: not following the exact same path ("homing"), but nearly always keeping the human within range of the laser.

First test ever of nighttime EVA traverse. Laser tracking was tested during a nighttime traverse of the same path described above. This traverse was primarily designed to test a lighting system on the spacesuit, but it was an excellent test of the GTA and greatly increased confidence in the system: during the daytime traverses, someone was always monitoring robot vital statistics and behavior on a laptop display while sitting within line of sight of the robot, prepared to halt the robot if any problems developed. During the nighttime traverse, however, monitoring was of marginal use since the robot was not visible. The traverse proceeded successfully and without event.

Science package deploy. Finally, laser tracking was used during a geophone deployment run. This scenario required the robot to travel a straight course, stopping every N meters to deploy one of the science instruments. If GPS had been available, a sequence of waypoints could have been feed to the tracking server to accomplish the task (more) autonomously. Instead, a person walked in front of the robot, stopping every N meters. The robot tracked the human using laser data, and thus stopped when the human stopped and moved when the human moved. As long as the human followed the straight line, the robot did also.

GPS Tracking field results

It's always something. Software and hardware problems delayed use of GPS tracking until the final days of the September field tests. Radio frequency leakage from the PC104 stack interfered with the GPS and correction radios collocated in the backpack. Shielding solved this problem at the cost of many field hours.

As mentioned, GPS tracking will not work if the robot's heading data is poor or non-existent. The digital inertial measurement unit (DMU) attitude and heading sensor on board the robot did not always function properly, and gave erroneous heading values. Insufficient shielding was again the root cause, with calibration issues an aggravating contributor. The proper corrections have been made *since* returning from Arizona. The other source of heading information was computed within an averaging algorithm in the GPS server itself. Without a 2nd GPS onboard, heading is derived by sampling multiple readings from the one unit. The position of the robot over time can be used to calculate the current heading of the robot in certain cases. However, when the robot is backing up, standing still, or turning sharply, this simple algorithm does not produce a valid heading. Thus, in tracking runs which required the robot to back up or pause, the temporary lack of heading information caused unpredictable behavior.

Getting better all the time. These problems have now been corrected, and further tracking tests have been performed in the Mars Analog yard at Johnson Space Center. These tests demonstrate the utility and effectiveness of homing mode tracking and GPS. With GPS tracking, the human does not need to remain directly in front of the robot to maintain a tracking lock, but can walk ahead to a new site and still have the robot catch up as it can. Of course, the robot will need to have obstacle avoidance, and perhaps mapping and path planning abilities of its own in order to safely navigate. These capabilities are being actively developed for field tests in the desert in March 2003.
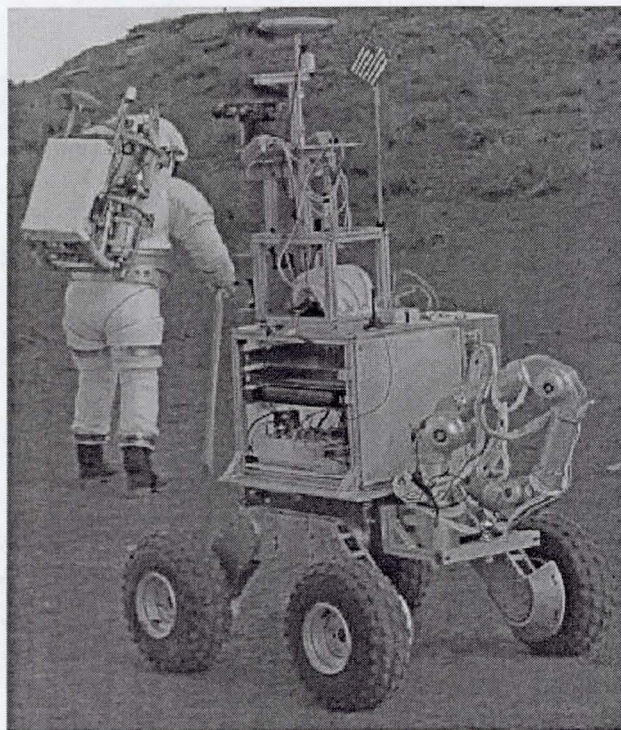

Figure 5 Tracking an astronaut with GPS

Finally, some data plots. Several tests were performed with a shirt-sleeved person wearing the differential GPS-enabled backpack at the JSC Mars Analog yard. The system performed well in certain cases, but several issues were exposed. Figure 6 shows a suited astronaut tracked by the ERA robot. The figure shows the ground tracks of the robot and human during one tracking run. The human, not burdened by the spacesuit, is able to walk faster than the robot can close. The desired separation distance was 2.0m, though the robot originally started out farther away from the human. The separation distance between the robot and the human over time is shown in Figure 7. The red line at 2.0 meters indicates the requested separation. Several points are labeled to show the correspondence between the two images.

A tracking story. The human and robot both started out on the east, or right side of the image. As the human walks toward point Ah and pauses there, the robot eventually catches up and reaches point Ar, achieving a 2.0m separation. The human then walks directly past and behind the robot, causing the separation distance to momentarily become less than 2.0 m, and proceeds to the west, then north

to near point Br1, and then walks rapidly to point Bh1. The robot begins to follow, but slows down near point Br1, due to the terrain and a lack of traction. As seen in Figure 7, the separation distance grows larger at this time, reaching a peak just after point B1.
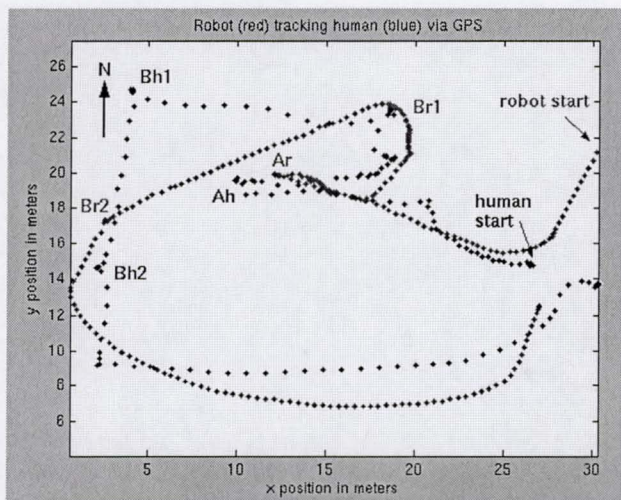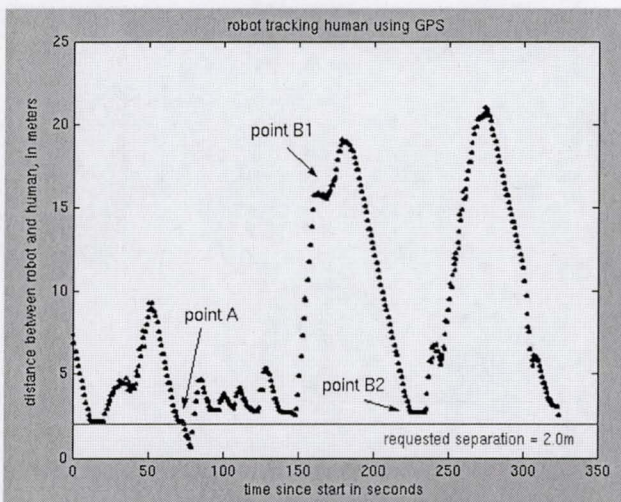

Figure 6 GPS ground track at JSC

Figure 7 Separation distances for GPS ground track

The human only briefly stayed at Bh1, and continued on to point Bh2, where he waited for the robot to catch up. The robot reaches point Br2, achieving a 2.0m separation. After a momentary pause, both human and robot proceeded to the end. The human walked rather quickly (about twice as fast as the robot), causing the robot to fall further behind and reach another peak in the separation distance plot. The separation distance closed when the human waited for the robot at the end of the traverse.

Conclusions and Future Work

The generic tracking algorithm worked well in its critical scenario role and as part of the highly parallel ERA architecture. Tracking humans in realistic field environments is demanding work for mobile robotics and current technology: there is always the fine line to be walked

between necessity and sufficiency in this arena, and the ERA project is constantly challenged in the field in this regard.

Specific enhancements are planned. For the PS, implementation of kalman filtering to improve pose fusion and prediction is planned, along with improvements to the confidence decay algorithm. For the GTA, the addition of a path-following mode to compliment the homing mode is planned, along with a callback method for notifying tracking clients with status of their tracking requests (currently clients make request to the TS then must wait-and-poll for results). We plan also to instrument the tracking architecture to collect metrics for analysis and comparison.

Testing and more testing. We plan to test obstacle detection and avoidance within the GTA for the basedrive actuator (and the 7DoF arm later this year) when returning to the field in March at the Mars Desert Research Station in Utah. Updates on these field tests and our project can be found at vesuvius.jsc.nasa.gov/er_er/html/era.

References

[1] A. Bruce, I. Nourbakhsh, R. Simmons, "The Role of Expressiveness and Attention in Human-Robot Interaction," IEEE Intl Conf on Robotics and Automation, 2002.

[2] R. Burridge, J. Graham, "Providing robotic assistance during extra-vehicular activity", SPIE, 2001.

[3] L. Davis, V. Philomin, R. Duraiswami, "Tracking humans from a moving platform," 15th Intl Conf on Pattern Recognition, Vol. 4, pp. 171-78, 2000.

[4] Fod, A. Howard, M. Mataric, "A Laser-Based People Tracker," IEEE Intl Conf on Robotics and Automation, 2002.

[5] R. Burridge, J. Graham, K. Shillcutt, R. Hirsh, D. Kortenkamp, "Experiments with an EVA Assistant Robot," i-SAIRAS, May 2003.

[6] E. Huber, D. Kortenkamp, "Using stereo vision to pursue moving targets with a mobile robot," IEEE Intl Conf on Robotics and Automation, 1995.

[7] Jung, G. Sukhatme, "Tracking Multiple Moving Targets using a Camera and Laser Rangefinder," Tech Report IRIS-01-397, Univ of Southern California.

[8] S. LaValle et al, "Motion Strategies for Maintaining Visibility of a Moving Target," IEEE Intl Conf on Robotics and Automation, 1997.

[9] Poell, C. Wengert, "Human Oriented Tracking and Mobile Robot Gesture Driving," VRAI Group Technical Report, Swiss Federal Institute of Technology, Lausanne, Switzerland, Feb 2002.

[10] S. Shiry et al, "Localization and Approaching to the Human by Mobile Home Robot," 9th IEEE Intl Workshop on Robot and Human Interactive Communication, IEEE-ROMAN, Sep 2000.

[11] R. Simmons et al., "Experience with Rover Navigation for Lunar-Like Terrains," IEEE Conf on Intelligent Robots and Systems, 1995.

[12] Wang, C. Thorpe, "Simultaneous Localization and Mapping with Detection and Tracking of Moving

Objects," IEEE Intl Conf on Robotics and Automation, 2002.

[13] www.techfak.uni-bielefeld.de/ags/ai/projects/mobile

[14] http://207.158.226.154/home.htm

[15] S.A. Roth, P.H. Batavia, "Evaluating Path Tracker Performance for Outdoor Mobile Robots"

[16] J. Baltes, "A Benchmark Suite for Mobile Robots", IEEE Conference on Intelligent Robots and Systems, 2000.

[17] T. Chen, R. Luo, "Mobile Target Tracking Using Hierarchical Grey-Fuzzy Motion Decision-Making Method", IEEE International Conference on Robotics and Automation, April 2000.

[18] KLS Robotics, "The Value of Robot Performance Testing"

[19] ISO 9283:1998

[20] J. Charles, W. Knudson, R. Cunningham, W. Roy, T. Smith, J. Gruener, M. Duke, S. Hoffman, D. Hamilton, N. Carbol, "The Mars Surface Reference Mission: A Description of Human and Robotic Surface Activities", Stephen Hoffman, ed. October 1998.

[21] FY03 Task Proposals for the NASA Exploration Team Human/Robot Working Group. Draft 07/23/2002.