A Cyber Physical System Testbed for Assistive Robotics Technologies in the Home

Velin Dimitrov, Vinayak Jagtap, Mitchell Wills, Jeanine Skorinko, Taşkın Padır
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts 01609
{vdimitrov, vvjagtap, mwills, skorinko, tpadir}@wpi.edu

***Abstract*—We present a cyber-physical system (CPS) testbed to enable the rapid development, testing, and deployment of assistive robotics technologies in the home of elderly individuals. We built a CPS testbed in a lab environment with initial capabilities allowing for the testing of both individual systems and collections of systems. The CPS testbed has communication, computation, sensing, and control resources available that can be leveraged by individual subsystems within the CPS. We present projects built by different design teams to be integrated in the CPS environment to help the elderly live independent lives and age in place. Finally, we describe a case study for the use of a mobile robot within the CPS to detect and respond in case an elderly person falls at home.**

I. Introduction

Today, there are 7 individuals in the United States for each person over the age of 65. Moreover, 23% of the younger adult population in the U.S. declare themselves as informal caregivers for an individual [1]. It is projected that in 2030, there will be 4 people for each person over the age of 65. Among these four people, one will be a child, one will be sick and one will be at a distant geographical location [2]. This implies that the ratio of younger adults as caregivers to older adults as individuals in need of care will be 1-to-1 in 2030. According to National Institute of Aging [3], it takes approximately 6.5 hours per day to care for a frail older adult and this is not sustainable for a caregiver while maintaining full-time employment. Furthermore, 75% of the seniors prefer to age-in-place [4].

In terms of psychological consequences, as individuals' age, they experience dramatic changes in their cognitive functioningtypically the speed at which cognitive functions can be performed decreases [5]. Thus, as individuals age, their cognitive abilities slow down and it takes them longer to react in different situations, remember names of others, and solve cognitive puzzles [6]. Although cognitive reactions slow down, the expertise an individual gained over time takes over to guides them, so older adults can still competitively play chess [7], and type as quickly and accurately as young adults [8].

Life satisfaction and happiness are also important considerations in understanding the aging process. Happiness doesn't differ for those over 65 [9]. In fact, older adults may be happier than younger adults. One explanation of these findings is that older adults tend to focus more on positive events. Research has shown that older adults experience and convey more positive emotions [10], and focus less on negative information [11]. This is not to say that older adults experience more intense positive feelings; rather they typically experience less intense joy, but a greater sense of contentment [12].

978-1-4673-7509-2/15/$31.00 2015 IEEE

Robotics, the integration of sensing, computation and actuation in the physical world, within the framework of a cyber-physical system (CPS) can potentially transform the capabilities of an individual with a disability in performing the activities of daily living [13]. In this paper, we describe our approach to build a CPS testbed and present one case study of a fall-detection robot that is part of the CPS. Technology will never replace a human caregiver. However they can provide an extended independent living for older adults, and hence, improve the quality of life for humans.

II. Testbed Description

Our overarching goal is to improve the quality of life for the elderly by developing an open-source, open-hardware CPS testbed for verification and validation of control algorithms. The research focuses on the design and validation of shared control algorithms for the human-in-the-loop CPS, the development of models for the proper autonomy levels in a human-robot team and the implementation of adaptive and context-aware control algorithms for reliable semiautonomous behavior in dynamic environments. We have an initial smart home environment deployed in which wheelchairs (and other smart robots) can operate safely to demonstrate that we can improve the quality of living for individuals with disabilities and older adults and hence, we can enhance the efficiency of the healthcare system by effectively interconnecting and operating CPS. We envision that smart robots (wheelchairs, assistive mobile robots, and even prosthetic devices) in the testbed will represent a CPS personalized in the sense that the control interface and autonomy level depend on the person's abilities to enable the human to independently navigate in the home.

Within this CPS testbed, we have enabled design teams to rapidly prototype assistive robotics technologies by following a model-based design methodology to design, model, simulate, and implement a unique smart sensor, actuator or controller node to create a smart living environment. The integration of a variety of embedded systems allows students with different backgrounds to complete level-appropriate and focused course projects. The smart environment is composed of proximity sensors, pressure sensors, motion detection sensors for activity monitoring, actuator modules for interacting with doors, windows, lights, and electronic devices, human-interfaces as personal assistants, and cameras for safety (fall-detection). Thus, design teams can define and validate subsystem requirements within this heterogeneous human-in-the-loop cyber-physical system.

The CPS testbed consists of a room with approximately 400 sq. ft. of space. Inside, a variety of commercially available products form the core of the CPS. Communications capabilities are provided by the combination of Ubiquiti Networks products: a UniFi AP-AC, Edge Router, and ToughSwitch. The 3x3 MIMO access point accommodates 450 Mbps at 2.4 Ghz and 1300 Mbps at 5.8 Ghz. The access point, router, and switch provide flexibility to implement VLANs, QoS, multiple SSIDs, custom routing and triggering providing scalability as more devices enter the CPS.

In terms of embedded devices, the CPS includes a Ubiquiti mPort to which a mFi-DS door sensor, mFi-MSC motion sensor, and mFi-THS temperature are connected. This provides the system with information about when and how long doors have been open, if there is any motion in the room, and what the ambient temperature is. In addition, three streaming IP cameras are placed throughout the room to accommodate any computer vision approaches to detecting, tracking, or assisting people in the CPS. In addition, a series of six Bluetooth low energy Estimote beacons are dispersed throughout the environment. These beacons are completely self-contained without any user serviceable components. By comparing the received signal strength (RSSI) from each beacon, a device that has Bluetooth 4.0 capabilities can localize itself to about 1-2ft accuracy in the room. Finally, a series of Ubiquiti mPower power strips are in the CPS, allowing remote control of electrical outlets and data collection on electricity usage on a per port basis.

In order to aggregate the data, a server running Ubuntu is also installed in the CPS. We treat this server essentially as a local "cloud". In a real deployment to a home, the devices would be connected to a commercially available cloud platform, but to help with development and debugging, we connect the devices locally in the testbed. The server runs the Ubiquiti mFi Controller and UniFi WiFi Device Management which interfaces and controls all the Ubiquiti devices. Finally, we deployed a local copy of a Phant service to allow embedded and networked devices to quickly and efficiently push/pull data through HTTP requests.

III. Elements of the CPS

As part of the Fall 2014 graduate-level course on model-based design, teams of students were tasked to create individual subsystems that would form elements of the CPS. During the 12 week project phases, the teams conducted brainstorming, design, implementation, and testing to develop devices and software elements that would be beneficial to an elderly population, enabling them to more safely live and age-in-place. Ultimately, the success of the projects ranged from robust implementations that could be deployed in a matter of months to proof of concept systems that would need more development, but nonetheless the projects demonstrated that the CPS testbed can be used to test and develop assistive robotic technologies in a quick and resourceful way. Next, we describe a sampling of the projects and how they fit within the CPS testbed.

A. Adjustable Height Cane

The team designed and developed a cane with an integrated actuator that allows it to adjust its height dynamically as the user moves around their environment. The purpose of the cane is to facilitate and ease the movement of the user, especially on stairs, landings, and when bending over to pick up objects. A 1/3" per revolution and 30:1 gearmotor are retrofitted in a milled slot on a regular metal can.

The actuation can be either commanded by the user, or commanded autonomously at certain areas like the top and bottom of stairs. When the user enters the bottom of the staircase area, the can retracts 7" to help the user up the stairs. When the user enters the top of the staircase, the cane extends 7" to help the user down the stairs. The retract and extend behavior of the cane is controlled by a finite state machine (FSM) which provides a desired length to a PID controller. The PID controller then spins the motor the correct amount to achieve the length.

The cane is useful not only in a stair climbing scenario, but would also be beneficial to assist the user while rising, either from a chair, getting out of bed, or bending over to pick up an object. The ultimate goal of the cane would be improve ease of use, so an untrained elderly user can understand how the cane operates in less than 30 seconds. The cane also provides a nice platform to integrate additional sensing such as heart monitors, fall detection, floor anomaly detection, and obstacle mapping within the home.

B. Pressure Sensitive Rug

The team integrated an array of force sensors on the bottom of a small area rug that would be common to a home environment. As an elderly person's senses and cognitive abilities degrade, one problem that occurs is they have a harder time recognizing if anyone is with them in a given room. Vision-based implementations to help track people throughout their homes have been developed before (in addition, we present one such project for tracking an assistive robot or wheelchair), but privacy concerns limit their applicability in real-world environments. Instead, a pressure sensitive rug can easily distinguish and track people throughout rooms of a home without the privacy concerns that cameras introduce.

To execute the pressure sensitive rug, an array of force sensors was attached to the bottom of an existing floor rug. A simple mass-spring-damper model was employed to model the forces of the users foot on top of the rug. In order to eliminate false positive detection from dropped objects, pets, etc, the action of a user stepping on and off the rug, from heel strike to toe off, was modeled in logic and detections that did not fit the forces associated with a step are rejected. This type of filtering proved very good at rejecting spurious and accidental detection.

Given the simplicity of the system, several assumptions were made including one person stepping on the rug at a time, the array of sensors is across the direction of travel (the array is parallel to a door opening for example), the person does not U-turn on the rug, and the person does not jump but walk onto the rug. Testing showed the system did not miss a single detection when a user walked onto the rug from within $\pm 45\deg$ of the centerline of the rug. In addition, various objects were dropped onto the rug to test the false positive rejection, and not a single objects registered as a footstep.

C. Biometric Sleep Detection

One of the most significant concerns about elderly aging-in-place and living independently is forgotten appliances that pose a fire hazard. One of the uses for the wearable health monitoring shirt from above is to monitor sleep. If a person's sleep can be reliably detected, an appliances connected to the WiFi controlled power strips can be automatically checked and turned off if they pose a fire risk.

By monitoring the heart rate, respiratory rate, and acceleration, we can accurately determine if the user is sleeping. To eliminate false negatives (which carry a heavy cost in this scenario), only three of the conditions, heart rate, respiration rate, respiration rate, or accelerometer data need to be below their thresholds to indicate the user has fallen asleep. A simple FSM defines these thresholds and controls the power to the appliances.

The system was tested with the shirt on a user who was in the process of falling asleep. The sensor data was then monitored by the FSM and within a prescribed time period detected the sleeping state and shut off the connected devices. Because of the relatively low thresholds for the sleep state, no false positives were detected during the user's daily routine. Future versions should have adaptive thresholds that eliminate the need to tailor them manually to each individual using the system.

D. Intelligent Stool

Short stools are a popular furniture item in many households. Elderly individuals like to use them because they are helpful for placing objects on top, seating, or using them as a footrest. The stool serves as an extension that helps the elderly by providing flexibility, comfort, and support as they need it. A serious issue though, stools present a tripping hazard when they are not in use. They are short, and may be more difficult to recognize than larger furniture such as tables making them a potential hazard.

We implemented an intelligent stool that can be summoned or dismissed based on voice commands. When the stool is needed, it approaches the user and provides the necessary assistance. When the user is done using the stool, they can command it to move away. The stool is implemented using the commercially available TurtleBot 2 platform, which is a small differentially driven robot approximately the size of a stool. The robot has a preloaded map of the environment and uses a 3D Adaptive Monte Carlo Localization (AMCL) algorithm to localize within the map. Using the Bluetooth localization beacons to locate the user, the robot can navigate to the user and assist them.

E. Assistive Walker

Walkers help the elderly by providing support and a platform to rest on when they have to stand for extended periods of time. We implemented a system that assist the elderly to navigate and move around the environment when using their walkers. Most commercially available walkers are simple aluminum frames with wheels and skids. Some have brakes included as well, but none have any sort of built in intelligence. We developed an assistive walker that has object avoidance using passive control through user cues. An ring of LED lights can help the user navigate around obstacles or through doorways without getting their walker stuck.

The system consisted of a regular aluminum walker, retrofitted with wheel encoders to provide odometry information, ultrasonic sensors and LIDAR to detect obstacles in front of the walker, and a ring of LED on the front bar to guide the user to a given direction. In testing, the walker was able to correctly guide a user through the center of a doorway, preventing them from bumping the sides. While the passive system would be useful to many individuals, and system that also activates the brakes may be of more use to the elderly with more severely degraded physical capabilities.

IV. Case Study: Fall Detection within the CPS

While the projects presented above provided a good breadth of different applications, we implemented a more in-depth case study of an important topic to aging-in-place: detecting falls. Accidental falls are a major cause of both fatal and non-fatal injuries in the elderly population with one in three experiencing a fall each year [14]. While in many cases people are transferred to a nursing home, this is not always an improvement as about 20% of falls for people 65 and over occur in nursing homes and only 5% of this population lives in a nursing home [15]. We demonstrate that a fall-detection robot as part of a HiLCPS within the home environment can allow the elderly to age-in-place.

Our approach, described here, uses a RGB-D camera to detect the presence of a person and an algorithm to detect when the detected person falls. This solution was implemented with the intention of being used on mobile robots; this would allow for it's use on a companion robot that would stay with the person.

A. Sensing

This approach is implemented using a Xbox Kinect RGB-D camera. While originally developed for the Xbox 360 gaming console, this sensor is cost-effective and produces a 640x480 color image and a 320x240 depth image at 30 frames per second making it extremely useful for various robotics applications. The combination of the camera and depth images can then be used to detect a fall. The high framerate is particularly important because a fall is a very dynamic event so having a number of frames of information is important for detecting it. The sensor also has existing drivers for retrieving images from the device and performing person detection as discussed in the next section.

B. Person Detection

The OpenNI NITE [16] person skeleton tracking framework is used in order to detect people in the sensors view. The implementation proved to be very robust to different orientations and positions of people. Other detectors were

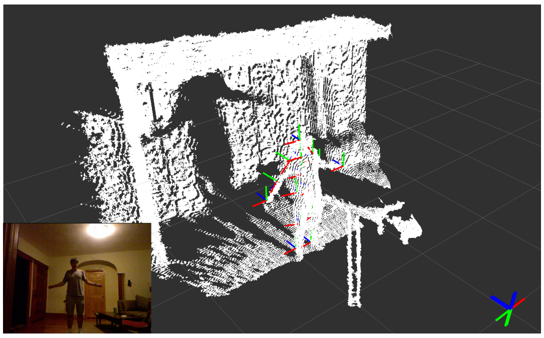

Fig. 1. OpenNI Person Detection

investigated, including the PCL ground plane people detector [17], but proved to perform significantly worse when the person was partially obscured or in a non-standing position. The PCL detector also requires extraction of the ground plane, which adds additional complexities and needs validated for dynamic environments if mounted on a mobile robot. OpenNI was also able to directly communicate with the Kinect eliminating the overhead of marshaling the data into a different format. The use of OpenNI also allows the algorithm presented to run with a number of different sensors other than the Kinect including the Primesense Sensor. Figure 1 shows the point cloud generated by the Kinect and places axes at the points identified on the body by the OpenNI skeleton tracking.

Other approaches, such as [18], involve attaching a sensor to the person being monitored. While this removes the need to distinguish a person from their surroundings, it reduces reliability since they have to remember to attach the sensor or always keep the device with them.

C. Detecting a fall

Previous work has taken similar approaches to detecting falls using RGB-D cameras [19] [20]. The approach described here expands on these approaches by adding additional processing for removal of false negatives and handling of partially obscured people.

1) Skeleton Processing: In order to detect a fall the system looks at the motion of the upper body of the subject. The system initially processes the information from the Kinect into a skeleton using the OpenNI NITE framework. The skeleton that is generated includes 3D points representing the estimated position of the head, torso, neck, shoulders, elbows, hands, hips, knees, and feet. While the positions of these points tend to be accurate when the entire body is in view, if obscured, the positions of the legs and arms default to fully extended or match to whatever looks most like the obscured body part. Because it can not be easily determined whether the limbs are visible or not the information is ignored and instead this algorithm focuses on the core body. The full skeleton is reduced to a bounding box defined by it's width, depth, and height and the vertical position of the torso. These values are then used in future steps to detect a fall. To compute the bounding box the detected skeleton points are reduced to those of the head, torso, neck, shoulders, and hips and the minimum and maximum positions are computed in all three dimensions. The width, height and depth of the box are then computed as in Equation 1. A value representing the average vertical position of the body is then extracted from the torso measurement.

$$\begin{aligned} W &= |X_{max} - X_{min}| \\ D &= |Y_{max} - Y_{min}| \\ H &= |Z_{max} - Z_{min}| \end{aligned} \tag{1}$$

2) Data Filtering: The width and height of the bounding box are then combined into a single measurement that represents the radius of the box face parallel to the ground (WD) using Equation 2. This eliminates the effect of the rotation of the person relative to the camera about the axis perpendicular to the floor.

$$WD = \sqrt{W^2 + D^2} \tag{2}$$

The rate of change of the bounding box WD (vWD), height (vH), as well as the absolute vertical position (vZ) from the skeleton are then computed between the previous and current frame. If any of the box size rate of changes are greater than a threshold speed, Tv, then the current frame is ignored and the velocity will be computed between the next frame and previous frame in order to smooth any sudden jumps in estimated position. The computed velocities are then passed into a Savitzky Golay filter as implemented in [21]; the filter operates over the past nine computed velocities and further reduces the noise that exists in the skeleton point estimation.

3) Fall Detection State Machine: In order to detect a fall the system looks for the height of the bounding box to begin decreasing simultaneously as WD is increasing. This would indicate that the person is getting shorter and wider, which could mean they are falling. Additionally, the vertical velocity of the person is checked to confirm that the person is moving downward. This value can be used to eliminate false positives, such as sitting down, by tuning it to be greater than the expected maximum speed during normal motions, but less than them falling. If all three of these conditions are met ($vH < -TvH, vWD > TvWD$ and $vZ < -TvZ$) for a given period of length $time_{falling}$ then the person is considered to be potentially falling. The skeleton is then monitored for inactivity ($vH < TiH$) for a period of time greater than $time_{inactive}$ seconds indicating that the person is no-longer moving. The period of inactivity must occur within another time period ($time_{settling}$) in order to give the person time to finish falling, eliminating potential false negatives. Once all of these conditions are true a fall has been confirmed. This process is described in Figure 2.

D. Relevance to Robots and CPS

While the method described here was implemented with a static sensor it could easily be added to an assistive robot, such as PARbot [22]. By placing the sensor on a mobile platform that can follow the user around, the user can always be in view of the system without the need to place sensors in every room. The robot could also reposition as the subject moves about a room in order to ensure full coverage of a room without the need for multiple cameras. The fall detection algorithm is independent of motion in the plane of the floor so the

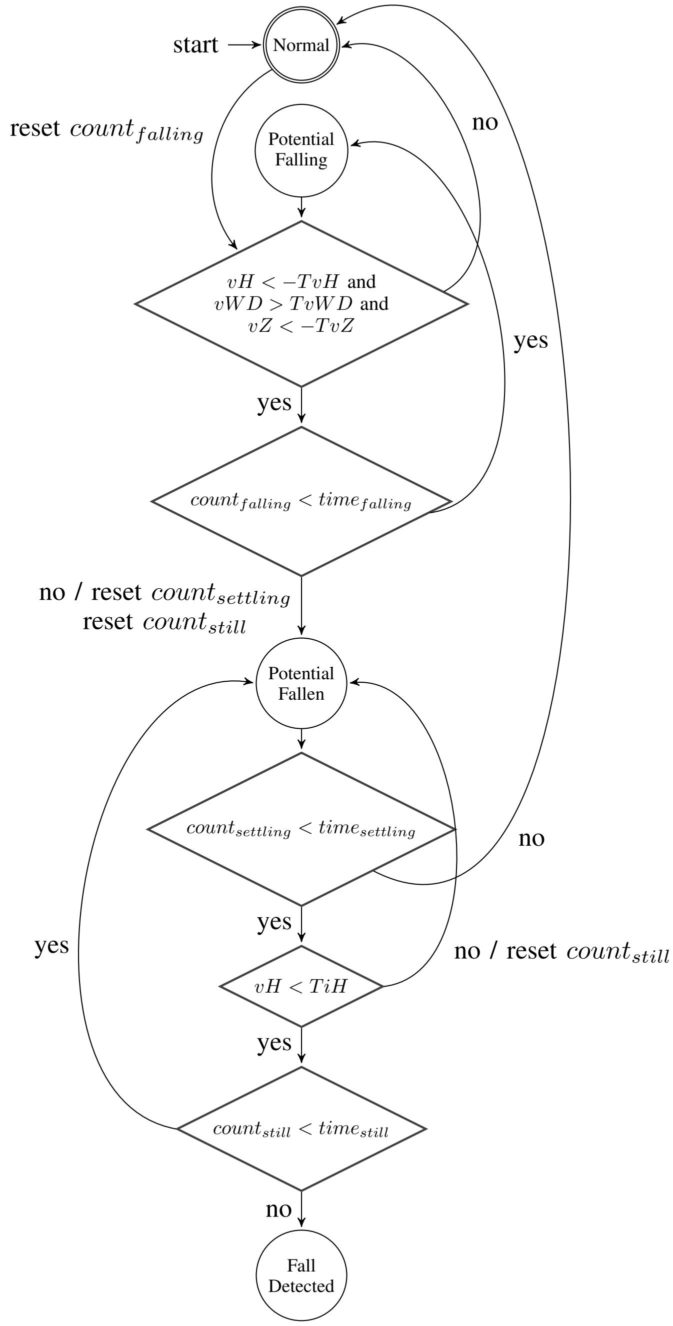

Fig. 2. State Machine of Algorithm

robot is free to move around a room, as long as it is able to keep the subject in view. Combined with the information from the ambient intelligence in a HiLCPS home environment, the robot can ensure the safety of the person in a wide variety of scenarios.

V. Experimental Results

A. Experimental Setup

The system was tested in a scenario placed in a room roughly 15' by 18' with various pieces of living room furniture (a couch, table and multiple chairs scattered around the room in different locations). The Kinect was placed in different locations around the room at a height of about 2.5'. This height was chosen because it represented a minimum height for a camera to observe a subject lying on the ground. While the camera could be placed higher in the room, such as ∼6.5' in [20], it would require a taller robot and placing it at a lower level tested the limits of what could be seen on the floor; placing the camera higher would have allowed better visibility of a person laying on the ground because the view of the person would be less angled. Placing the camera lower would have caused the view angle to decrease to the point where it would be impossible to detect a body on the ground; however, most of the fall testing was done onto a mattress that was a foot tall so the falling person was only being observed from a height 1.5' above the surface they fell on. Also, if the sensor is placed too low then it can be obscured by furniture in a living space so a height must be chosen based on potential visual obstacles.

B. Testing

The algorithm was tested with three datasets, each containing a recording of actions from a different person. The datasets consisted of people performing a number of actions in the space described above; these included walking around, standing still, sitting down, standing up, and falling all in a number of different orientations. As shown in Table I once the system is tuned the algorithm is able to perform well with minimal false positives and only a few false negatives. The parameters used for this test are described in Table II. In order to prevent injury during testing all falls were done onto a twin mattress lying on the floor. This is a potential source of experimental bias introducing inconstancy between how a person actually falls and the recorded falls for the datasets. In some cases falling on the mattress would result in the subject bouncing upwards after initially landing; while this did not appear to affect most results, in a few cases it caused the subjects legs to bounce up in the air. This caused the person to have an apparent upwards velocity to the system. With these two results ignored the algorithm performs fairly well over all of the testing. Another potential inconsistency with use cases is that the speed of movement may have been greater than those who would be observed; however, this difference only makes it more difficult for the algorithm to distinguish between a fall and a person sitting down as the faster someone sits down the more it looks like a fall; this means that the algorithm should be more robust to false positives in the case of someone moving slower.

TABLE I. Detection accuracy results

Action	Detected Falls	Total	Accuracy
Falling	7	9	78%
Falling (no bounces)	7	7	100%
Sitting Down	0	4	100%

TABLE II. Algorithm Parameters

Name	Value
Tv	0.8 m/s
THv	0.03 m/s
$TvWD$	0.11 m/s
TvZ	0.65 m/s
TiH	0.45 m/s
$time_{falling}$	0.12 s
$time_{inactive}$	0.5 s
$time_{settling}$	2.0 s

VI. Future Work for Fall Detection

The system presented here has proved to be a robust way of detecting the fall of a person. By using the rate of change of a bounding box surrounding a subject the algorithm is able to detect the fall of a person that is independent of movement of the sensor along the ground. The use of a smaller bounding box containing just the upper body also allows for better handling of then the person is partially obscured. This makes is an optimal option for implementation on a mobile robot system. Two mobile robots that are currently being used to develop assistive robotic systems for older adults are PARbot and the commercially available Turtlebot 2. Both of these robots are wheel-based, allowing for smooth movement along a floor and already contain RGB-D cameras that are compatible with the OpenNI driver used in this implementation. Additionally, both robots are already using ROS, making it easier to integrate. The addition of a mobile robot also allows for interaction after a fall has been detected. The robot can reach out to the user and determine if they are responsive and determine their current state, such as whether they actually fell, if they are OK, or if they need help to get up.

Further investigation of the detection of the person could also be conducted. The system as implemented here only supports one user; however, it could easily be extended to support multiple users with the same algorithm as the OpenNI skeleton tracker supports up to 15 people. Additionally, the version of the OpenNI skeleton tracker used requires an initial calibration of the user when the program starts. Additional work could be put into making this a one time calibration that could be stored or removing the need for it entirely.

VII. Conclusion

We have presented a HiLCPS testbed to enable the quick development, testing, and deployment of assistive robotics technologies in the home of elderly individuals. We built testbed in a lab environment with initial capabilities allowing for the testing of both individual systems and a collection of systems built by different design teams. The CPS testbed consists of equipment enabling the individual systems to communicate and exchange information, a powerful server to emulate the computational power available in commercial cloud solutions, and a selection of sensors and controls over the ambient intelligence integrated within the CPS environment.

We presented eight different projects built by different design teams meant to integrate in the CPS environment to help the elderly live independent lives and age in place. Projects included a cane that can adjust height to help the user, pressure sensitive rug to detect people entering/exiting rooms, a wearable health monitor to detect emergency situations, a sleep detection algorithm to shut off unused appliance that are a fire risk, an intelligent stool to minimize the risk of tripping, an assistive walker to help the user navigate the environment, a vision-based localization device for generic mobile platforms, and finally a system to provide wardrobe suggestions based on the weather. Finally, we provided an in depth case study of the potential use of a mobile robot within the CPS to detect and respond in case an elderly person falls at home.

Acknowledgment

The authors would like to thank Intel Corporation for their support of the Fall 2014 Model Based Design curriculum and students of the class who implemented projects in the CPS.

References

[1] "U.S. Census Bureau," 2011, http://www.census.gov.

[2] C. Angle, "iRobot entering healthcare robotics business," 2009, TEDMED Conference.

[3] "National Institute on Aging," 2011, http://www.nia.nih.gov.

[4] "California HealthCare Foundation," 2011, http://www.chcf.org/.

[5] T. A. Salthouse, "The processing-speed theory of adult age differences in cognition," *Psychological Review*, vol. 103, pp. 403–428, 1996.

[6] T. R. Bashore, K. R. Ridderinkhof, and M. W. van der Molen, "The decline of cognitive processing speed in old age," *Current Directions in Psychological Science*, pp. 163–169, 1997.

[7] N. Charness, "Aging and skilled problem solving." *Journal of Experimental Psychology: General*, vol. 110, pp. 21–38, 1981.

[8] T. A. Salthouse, "Effects of age and skill in typing." *Journal of Experimental Psychology: General*, vol. 113, pp. 345–371, 1984.

[9] D. G. Blanchflower and A. J. Oswald, "Hypertension and happiness across nations," *Journal of Health Economics*, vol. 27, pp. 218–233, 2008.

[10] D. K. Mroczek, "Age and emotion in adulthood." *Current Directions in Psychological Science*, vol. 10, pp. 87–90, 2001.

[11] L. I. Carstensen and J. A. Mikels, "At the intersection of emotional and cognition: Aging and the positivity effect," *Current Directions in Psychological Science*, vol. 14, pp. 117–121, 2005.

[12] R. E. Harlow and N. Cantor, "Still participating after all these years: A study of life task participation in later life." *Journal of Personality and Social Psychology*, vol. 71, pp. 1235–1249, 1996.

[13] C.-A. Smarr, C. B. Fausset, and W. A. Rogers, "Understanding the potential for robot assistance for older adults in the home environment," 2011, http://smartech.gatech.edu/handle/1853/39670?show=full.

[14] "Falls among older adults: An overview," Centers for Disease Control and Prevention, Tech. Rep., 2014, http://www.cdc.gov/homeandrecreationalsafety/falls/ adultfalls.html.

[15] "Falls in nursing homes," Centers for Disease Control and Prevention, Tech. Rep., 2014, http://www.cdc.gov/homeandrecreationalsafety/falls/nursing.html.

[16] "Primesense releases open source drivers, middleware that work with kinect," 2010. [Online]. Available: http://www.joystiq.com/2010/12/10/primesense-releases-open-source-drivers-middleware-for-kinect/

[17] "Detecting people on a ground plane with rgb-d data," 2013. [Online]. Available: http://pointclouds.org/documentation/tutorials/ground_based_rgbd_people_detection.php

[18] Y. Lina Tong, Quanjun Song and Ming Liu, "Hmm-based human fall detection and prediction method using tri-axial accelerometer," vol. 13, pp. 1849–1856, 2013.

[19] G. Mastorakis and D. Makris, "Fall detection system using kinects infrared sensor," *Journal of Real-Time Image Processing*, pp. 1–12, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11554-012-0246-9

[20] V. Bevilacqua, N. Nuzzolese, D. Barone, M. Pantaleo, M. Suma, D. D'Ambruoso, A. Volpe, C. Loconsole, and F. Stroppa, "Fall detection in indoor environment with kinect sensor," in *Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014 IEEE International Symposium on*, June 2014, pp. 319–324.

[21] "Cookbook / savitzkygolay," 2012. [Online]. Available: http://wiki.scipy.org/Cookbook/SavitzkyGolay

[22] K. Burns, N. Godani, O. Hugal, J. Orszulak, and J. V. Wambeke-Long, "Parbot: A personal assistive robot," Worcester Polytechnic Institute, Tech. Rep., 2014, http://www.wpi.edu/Pubs/E-project/Available/E-project-043014-173250/unrestricted/PARbot_MQP_Final_Report.pdf.