A Web-Based System for Home Monitoring of Patients With Parkinson's Disease Using Wearable Sensors

Bor-Rong Chen, Shyamal Patel, Thomas Buckley, Ramona Rednic, Douglas J. McClure, Ludy Shih, Daniel Tarsy, Matt Welsh, and Paolo Bonato*

***Abstract*—This letter introduces MercuryLive, a platform to enable home monitoring of patients with Parkinson's disease (PD) using wearable sensors. MercuryLive contains three tiers: a resource-aware data collection engine that relies upon wearable sensors, web services for live streaming and storage of sensor data, and a web-based graphical user interface client with video conferencing capability. Besides, the platform has the capability of analyzing sensor (i.e., accelerometer) data to reliably estimate clinical scores capturing the severity of tremor, bradykinesia, and dyskinesia. Testing results showed an average data latency of less than 400 ms and video latency of about 200 ms with video frame rate of about 13 frames/s when 800 kb/s of bandwidth were available and we used a 40% video compression, and data feature upload requiring 1 min of extra time following a 10 min interactive session. These results indicate that the proposed platform is suitable to monitor patients with PD to facilitate the titration of medications in the late stages of the disease.**

***Index Terms*—Body sensor network (BSN), home monitoring, Parkinson's disease (PD), telemedicine, wearable sensors.**

I. Introduction

PARKINSON'S disease (PD) affects more than 500 000 U.S. residents. The main motor features of PD are tremor, bradykinesia, rigidity, and impairment of postural balance [1]. Current therapies are effective in managing Parkinsonian symptoms in the early stages of the disease. However, in late-stage PD, patients develop motor complications, namely the abrupt loss of the efficacy of medications at the end of a dosing interval and involuntary hyperkinetic movements referred to as dyskinesia [2]. Monitoring changes in the severity of symptoms and motor complications (referred to as motor fluctuations) could facilitate the titration of medications, an aspect of the clinical management of patients with PD that becomes challenging as the disease progresses.

In clinical practice, information about the severity of motor fluctuations is obtained via self-reports and diaries. These methods are subject to perceptual and recall bias. Direct observation by a PD specialist is impractical because motor fluctuations cover the time span of several hours between medication dosages. Wearable systems have the potential to provide a tool to address the shortcomings of existing approaches. Wearable systems have been used to gather data from a variety of different patient populations [3]. Our team and others [4]–[7] have shown that wearable units equipped with accelerometers can be used to track the severity of symptoms and motor complications in patients with PD.

The translation of these experimental results into clinical practice requires: 1) the development of a system with wearable sensors that allow one to carefully manage resources such as battery life and processing power to achieve monitoring over several days and 2) the implementation of a web application to provide remote access to data collected using the wearable units. Recent research has shown that, by carefully managing system resources dynamically, wearable wireless units can collect data for several days without recharging their batteries [8], [9]. However, wearable sensors with such capability have never been combined before with software to remotely control the collection of data.

In this letter, we present a system called MercuryLive that provides an integrated platform to enable access to data gathered using wearable sensors via a web application. The system provides clinicians with a means to interact remotely with patients in the home setting, to configure the sensor nodes for the application at hand, and to record annotated data. We are currently deploying the system. In this paper, we present preliminary data suggesting that MercuryLive is suitable for use in clinical practice. Collecting data in the home setting using MercuryLive has the potential to allow clinicians to improve quality of care in patients with PD while reducing its costs.

Manuscript received August 21, 2010; accepted September 24, 2010. Date of publication October 28, 2010; date of current version February 18, 2011. This work was supported by the Michael J. Fox Foundation for Parkinson's research under the project entitled "Wearable Sensors and a Web-Based Application to Monitor Patients with Parkinson's Disease in the Home Environment." *Asterisk indicates the corresponding author.*

B.-R. Chen, T. Buckley, and M. Welsh are with the Harvard School of Engineering and Applied Sciences, Cambridge, MA 02138 USA (e-mail: brchen@eecs.harvard.edu; tbuckley@fas.harvard.edu; mdw@eecs.harvard.edu).

S. Patel is with the Department of Physical Medicine and Rehabilitation, Harvard Medical School, Spaulding Rehabilitation Hospital, Boston, MA 02114 USA (e-mail: spatel19@partners.org).

R. Rednic is with Cogent Computing Applied Research Centre Coventry University, Coventry, CV1 5FB, U.K. (e-mail: rednicr@coventry.ac.uk).

D. J. McClure is with Healthreageous, Boston, MA 0214 USA (e-mail: doug.mcclure@healthrageous.com).

L. Shih and D. Tarsy are with the Department of Neurology, Beth Israel Deaconess Medical Center, Boston, MA 02215 USA (e-mail: lshih@caregroup.harvard.edu; dtarsy@bidmc.harvard.edu).

*P. Bonato is with the Department of Physical Medicine and Rehabilitation, Harvard Medical School, Spaulding Rehabilitation Hospital, Boston, MA 02114 USA, and also with The Harvard-MIT Division of Health Sciences and Technology, Cambridge, MA 02139 USA (e-mail: pbonato@partners.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBME.2010.2090044

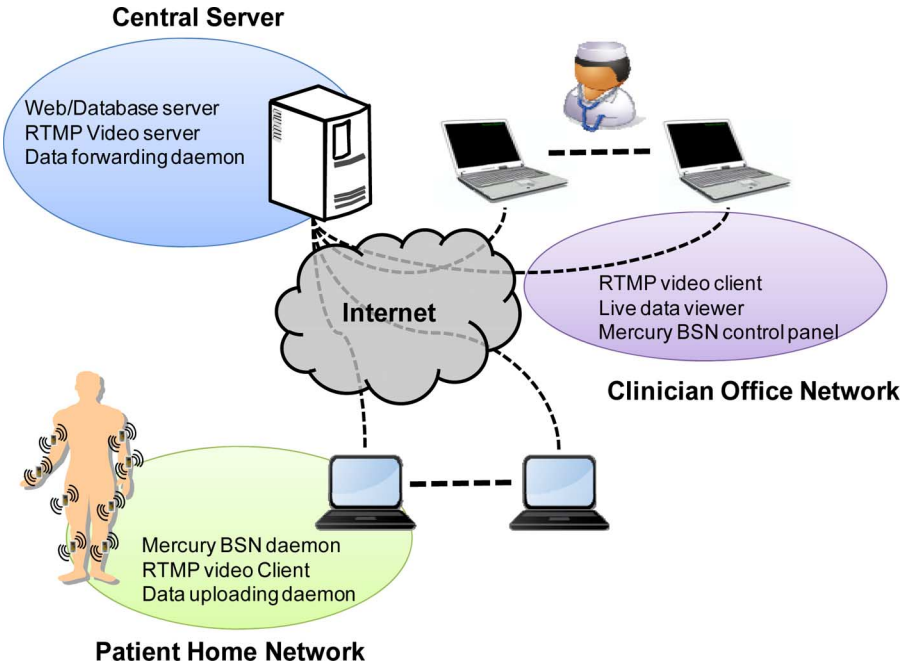

Fig. 1. Schematic representation of the MercuryLive system architecture.

II. System Description

Fig. 1 shows the general architecture of MercuryLive. The platform includes software services running at three tiers: 1) central server; 2) patient's hosts; and 3) clinician's hosts. In the following, we describe the services implemented at each tier.

A. Central Server

A central portal server provides a secure central location for coordinating data collection and video conferencing services. The portal server resides in a secure healthcare provider data center and allows access to resources only over securely encrypted services, including Secure Sockets Layer (SSL), Secure Shell (SSH), and virtual private network (VPN). Using secure channels, both patients' and clinicians' software clients can access the database, web server, video conferencing service, and a live data forwarding service to perform background data logging and live interactive sessions.

Three network services run on the central server. The web/database server provides data storage and access control to the patient data. A live video streaming service, provided by open source Red5 server, runs on the central server. A third daemon that runs on the central server is a live data forwarding daemon. This daemon forwards a decimated version of a patient's live data stream to the clinician's web graphical user interface (GUI).

B. Patient's Host

The patient's host runs Mercury, a body sensor network (BSN) platform developed by our team [8] that is based on the SHIMMER sensor [10]. A Mercury network consists of multiple sensor nodes worn by a patient and a base station, which is typically a laptop with an 802.15.4 transceiver. Mercury provides clinicians with the ability to adjust remotely different parameters of the sensor nodes (e.g., number of recorded sensor channels, sampling rate, and data features estimated on the node). Clinicians can set a desired battery life and Mercury adjusts how system resources are used to achieve the target battery life. This requires tuning sensor operations based on power consumption. As sensor data are being collected, a data uploading daemon runs in the background to connect to the central server and upload sensor data opportunistically. A web-based flash video conferencing client running on the patient's host provides live video interaction features. During an active video session, the Mercury sensor network streams decimated raw signals through the live data forwarding services at the central server.

We implemented several techniques to improve the robustness of Mercury and prevent data loss. First, the firmware running on the SHIMMER sensors periodically stores checkpoint states to a specific sector of the flash storage. When the sensors encounter software errors that crash the firmware, a watchdog timer will reboot the node. Once the node boots up, the firmware loads the previously stored checkpoint information from the flash and resumes data collection. Data during the rebooting process are lost. When such failure happens, a SHIMMER sensor can recover within 10 s. The data uploading daemon may encounter loss of Internet connection to the central server. When that happens, the data uploading daemon will attempt to reconnect to the server periodically. Such a connection problem will delay the data uploading to the database but will not cause data loss.

C. Clinician's Host

Besides collecting, storing, and securely providing patient data, our system also supports live video communication capability between clinicians and patients. Using the video

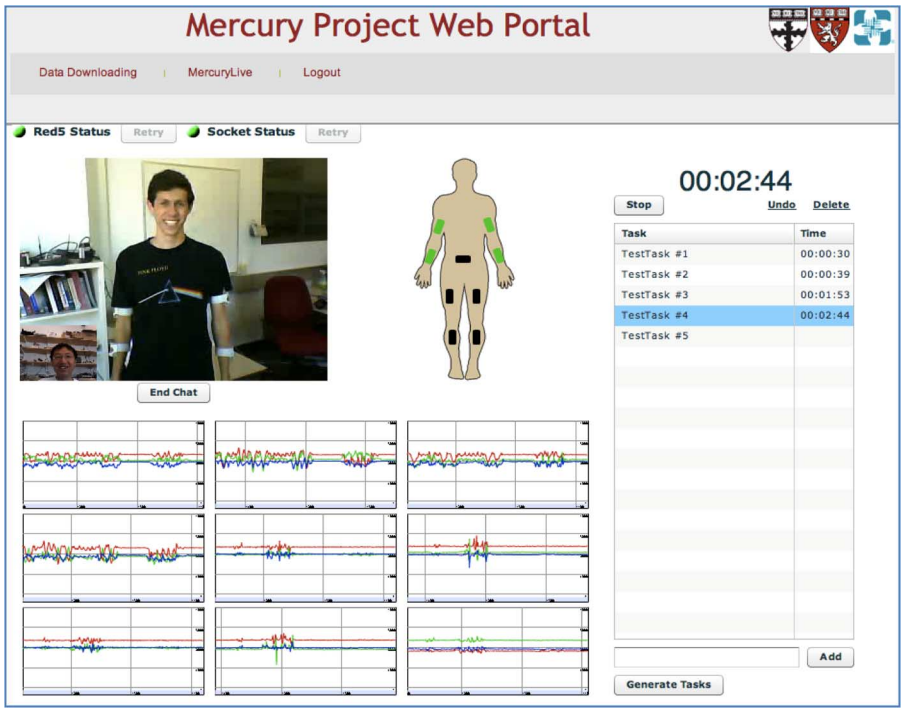

Fig. 2. Screenshot of the MercuryLive GUI as seen by the clinician.

interaction feature, clinicians can remotely conduct supervised data collection sessions. Clinicians can access the patient data and choose to speak to patients using the video conferencing service if needed. Fig. 2 shows the web-based GUI for clinicians to conduct supervised data collection sessions and view data in real time. To provide a user-friendly interface, we implemented a cross platform web application that runs as a flash plug-in. This application contains a GUI to display live decimated accelerometer data alongside the video session to allow clinicians to perform spot checks. Clinicians can also annotate events using integrated annotation tools. Finally, using MercuryLive, clinicians can access and download data.

III. System Characterization

To characterize MercuryLive, we assessed latencies and bandwidth requirements at different tiers of the system. Latency calculations were based on timestamps collected at each host as relevant control or data packets reached each tier of MercuryLive. To ensure measurement accuracy, the system clocks of all hosts were synchronized to the same Network Time Protocol (NTP) server.

A. Command Latency

Command latency was defined as the time between when a command is issued on the clinician's host and when it is received and acknowledged at the patient's host. Commands are issued by the clinician's host to reconfigure the BSN (e.g., change in data sampling rate). Three services contribute to the total command latency value: 1) the clinician's GUI; 2) the data forwarding daemon; and 3) the Mercury BSN daemon. Tests to estimate the command latencies introduced by each service were performed with the patient's host located in an urban home with cable Internet service that provided bandwidth of 12 Mb/s for downloading and 2 Mb/s for uploading. The central server was located in a research laboratory while the clinician's host was connected to the central server through a 100 Mb/s LAN. These conditions were chosen to simulate a clinical monitoring situation in which clinical personnel access patient's data from a clinical location. Measurements were repeated 100 times and average and standard deviation of the estimated latency values were derived. A latency of approximately 620 ± 240 ms was found to be introduced by the Mercury BSN daemon, whereas latencies lower than 10 ± 5 ms were introduced by the clinician's GUI and the data forwarding daemon. It is worth emphasizing that the Mercury BSN daemon needs to communicate with each sensor node, execute the command, and send back an acknowledgment. Also, it is worth noticing that the Mercury BSN daemon latency was measured using a system with nine sensor nodes. In a clinical scenario, we anticipate that fewer sensor nodes would be needed and thus the latency would be lower.

B. Data and Video Latency

Data latency was defined as the delay in the live streaming of the decimated version (from 100 to 10 Hz) of the sensor data. The latency time was defined as the time between when a sensor sends a packet and the time when the packet is received by the clinician's host for display. Tests were performed using the same configuration of computers used to estimate command latencies (i.e., the patient's host was located in an urban home, while the clinician's host was connected to the central server—located in a research laboratory—through a LAN). Data latency estimates were derived over a 10 min period. Tests were performed using

TABLE I
MERCURYLIVE VIDEO LATENCY

Latency (ms) Frame Rate (fps)		Video Compression		
		80%	40%	none
Available Bandwidth (kbps)	80	2098 ± 788 4.7 ± 2.1	2930 ± 795 3.3 ± 1.6	4709 ± 3494 1.1 ± 0.4
	800	235 ± 30 13.2 ± 1.0	209 ± 22 13.5 ± 1.1	556 ± 211 3.2 ± 0.9
	Unlimited	229 ± 27 13.1 ± 0.9	225 ± 33 13.3 ± 1.1	511 ± 68 3.0 ± 0.6

nine sensor nodes. The results showed a data latency of 373 ± 182 ms/packet (each packet contained 30 data samples). It is worth emphasizing that using fewer sensors would have led to a reduction in back-off time, thus resulting in a lower latency value. Back-off time is the time a sensor waits before sending a packet after a collision occurs.

Video latency was defined as the time between when a frame is generated at the patient's end and when it appears at the clinician's end. We measured the video latency of our system using vDelay [11]. vDelay is based on transmitting EAN-8 barcodes with an embedded timestamp. At the receiving end, the timestamp is decoded and compared with the local clock. For this set of experiments, we used a computer with an Intel Core Duo i7 processor, a Dell 1907FP flat panel monitor, and a Logitech Quickcam Pro 9000 webcam. To test the effect of different bandwidth conditions, we used Traffic Shaper XP [12]. Traffic Shaper XP achieves bandwidth management by controlling the network stack and introducing delays in the transmission of outgoing packets. As a result, the total bandwidth use cannot exceed the configured maximum rate. We set different video compression values (80%, 40%, and 0%) to test for video quality. By performing these simulations, we determined how video latency and frame rate are affected by bandwidth conditions and video compression levels. It is worth emphasizing that flash adjusts video frame rate automatically according to the available bandwidth and desired video quality.

Table 1 summarizes the results of our tests. As the video compression increased, the video latency time decreased and the frame rate increased. A simulated available bandwidth of 800 kb/s with a 40% video compression level provided acceptable values of latency and frame rate. This result indicates that any home broadband Internet service with 1 Mb/s two-way bandwidth can easily support MercuryLive. It is worth emphasizing that bandwidth requirements for transmitting the decimated version of the sensor data (4 kb/s), raw data sampled at 100 Hz (43 kb/s), and data features derived from 5 s segments of accelerometer data [7] (1 kb/s) are all negligible compared to the bandwidth requirement associated with video conferencing (800 kb/s).

C. Recovery Latency

System recovery latency was defined as the time required by the system to start functioning again after a system failure. To estimate system recovery latency, we connected a sensor node to a PC via its serial port. We reset the sensor node and associated a timestamp with this event. We then waited until the sensor node completed reinitializing at which time the sensor node sent a command to the PC and the PC recorded a timestamp to mark the event. Latency values were estimated as the difference between the two timestamps associated with the reset command and with completion of the sensor node reinitialization process.

System recovery latency values are determined by the configuration of Mercury. With our current implementation, if a failure causes a sensor to reboot, the recovery time for its live data stream is about 13 s and the maximum full resolution data loss is about 63 s. Maximum data loss (63 s) is determined by the watchdog timer (which checks for system failures every 1 s), the firmware reboot delay (2 s), and a checkpoint mechanism that stores information such as amount of samples on flash and the battery charge. This mechanism is set to run every 60 s. These numbers are for a single sensor node. In case multiple sensor nodes go down, the recovery time is only slightly increased since the recovery of each node occurs in parallel. Also, the recovery of these services is completely automatic and does not require any intervention by the subject.

Live data streams can also be interrupted due to several system failures: 1) Mercury BSN network failure; 2) SHIMMER sensor failure; and 3) Internet connection failure. Factors that influence live stream recovery time are: 1) real-time sample refresh period (10 s); 2) watchdog timer (which checks for system failures every 1 s); 3) firmware reboot delay (2 s); and 4) autoreconnection delay from patient's host to the central server (1 s). The real-time sample refresh period is the time interval between consecutive commands sent to the SHIMMER sensors by the Mercury BSN daemon to start decimated live data streams. All in all, it takes up to 10 s after a Mercury BSN network failure to restart the live data stream. The watchdog timer and firmware reboot time also contribute to the delay if the failure involves SHIMMER sensor reboot. In the case that an Internet connection failure from the patient's host occurs, the live stream will resume within 1 s (after Internet connection is re-established) as the autoreconnect timer is set to 1 s. There will be no full resolution data loss as long as the SHIMMER sensors remain alive during this period.

D. Data Upload Latency

MercuryLive opportunistically uploads data from the sensor nodes to a database that resides on the central server. Due to bandwidth and battery life limitations, MercuryLive does not provide real-time access to the full resolution data. Instead, raw data and/or data features are logged to the onboard flash memory and uploaded to the central server when possible. Consequently, data are uploaded (from the sensor nodes to the central server) with latency.

To estimate the data upload latency, we performed tests in which data features and raw data were uploaded onto the central server from the patient's host. These experiments were performed using the same configuration of computers used to estimate the command latency (see the related section

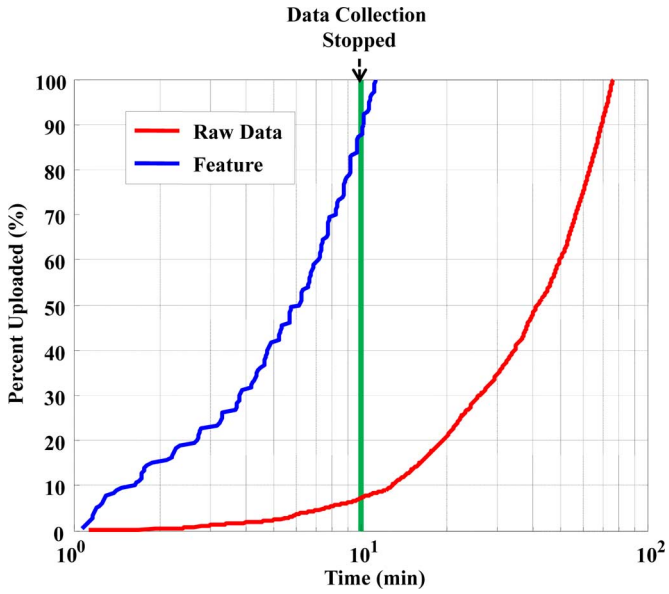

Fig. 3. Plot of the time required to upload 10 min worth of full resolution data (red) at 100 Hz and related features (blue) for nine sensor nodes each sampling three axes of accelerometer data.

aforementioned). The patient's host was located in an urban home. The central server and clinician's host were located in a research laboratory with a 100 Mb/s LAN connecting the clinician's host and the central server. Raw data were sampled at 100 Hz and three accelerometer channels were recorded per sensor node. Also, from each accelerometer channel, a set of six features were extracted (for a total of 18 features for each sensor node) using a 5 s sliding window with an 80% overlap (4 s). Fig. 3 shows the background uploading service progress over time during and after a 10 min interactive session. In this experiment, the full-resolution data recorded during the 10 min of interactive session are available on the MercuryLive portal about 80 min after the interactive session ends. In contrast, data features are available within 1 min from the end of the data collection. These numbers are for nine active sensor nodes. For an application that requires real-time decision making, transmitting features is the most feasible approach with currently available technology. Real-time decision making is not a requirement when monitoring patients with PD. Therefore, it is acceptable to allow a delay between the time when data are collected on the sensor nodes and the time when it is made available to clinical personnel for analysis.

IV. Pilot Deployment

MercuryLive is currently undergoing testing in patients with PD in the home setting. During the tests, we are placing sensors on the patient and we are giving him/her a laptop (patient's host) with webcam to interact with the clinician. The clinician monitors the patient from a clinical site and instructs the patient to perform motor tasks selected from the unified Parkinson's disease rating scale (UPDRS) [13]. Testing of the platform includes the use of algorithms [7] that estimate clinical scores capturing the severity of tremor, bradykinesia, and dyskinesia based on the

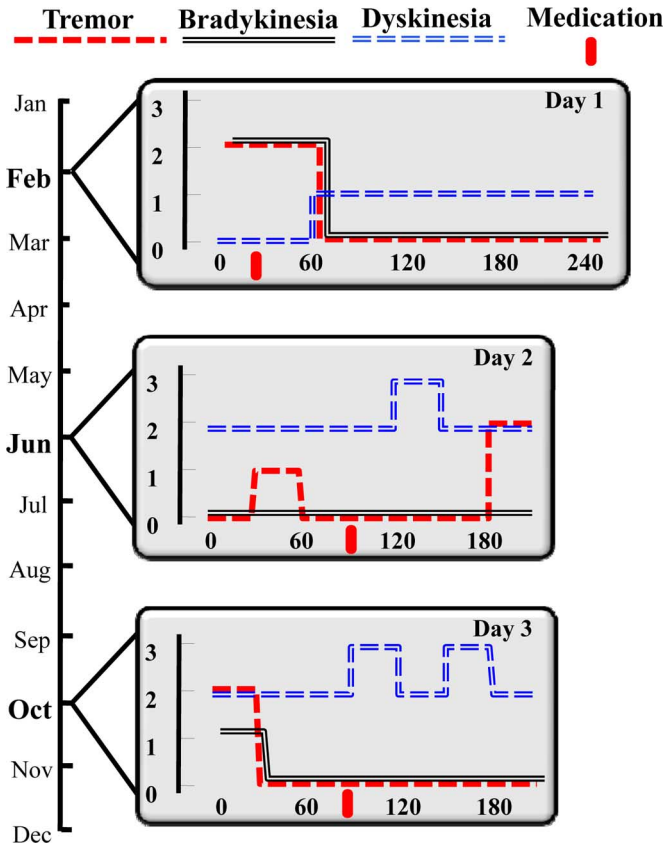

Fig. 4. Representation of clinical scores to be provided to clinicians at the end of the monitoring period.

analysis of accelerometer data. An initial implementation of a GUI to display such results is shown in Fig. 4. Clinical analysis of the information presented in Fig. 4 indicates, for instance, that the response to medication is good on Day 1 but it is not satisfactory during Days 2 and 3. Most noticeably, the plots suggest that the medication timing on Day 3 should have been delayed to reduce the amount of dyskinesia. Also the response appears to worsen during Day 3. This calls for a change in the timing and/or dosage of medication. Based on these preliminary results, we speculate that results from frequent longitudinal observations provide clinicians with information that could be used to adjust patients' medication regimen more effectively.

V. Conclusion

To our knowledge, the one presented herein is the first system that allows one to remotely monitor patients with PD in the home setting using wearable sensors. The system is an integrated platform that includes: 1) wearable sensors used to gather accelerometer data and 2) a web-based application that allows for two-way communication between patient and clinician. Via the proposed platform, clinicians can access sensor data, upload feature datasets, and estimate UPDRS scores. The system is currently undergoing extensive field testing, but preliminary results support its suitability to monitor patients with PD in the home setting and to gather information to facilitate the titration of medications in late-stage PD. Future study will be focused

on exploring the use of mobile devices as a gateway to collect data. Furthermore, we plan to explore the use of personal health records to store and display the results of the analyses of the data gathered in the home setting.

ACKNOWLEDGMENT

The authors would like to thank C. Mancinelli for her help with the data collections of patients with PD.

REFERENCES

[1] D. G. Standaert and A. B. Young, "Treatment of CNS neurodegenerative diseases," in *Goodman and Gilman's Pharmacological Basis of Therapeutics*, J. G. Hardman and L. E. Limbird, Eds. New York: McGraw-Hill, 2001, pp. 549–620.

[2] S. Fahn, "Levodopa in the treatment of Parkinson's disease," *J. Neural. Transm. Suppl.*, vol. 71, pp. 1–15, 2006.

[3] P. Bonato, "Wearable sensors and systems. From enabling technology to clinical applications," *IEEE Eng. Med. Biol. Mag.*, vol. 29, no. 3, pp. 25–36, May–Jun. 2010.

[4] N. L. Keijsers, M. W. Horstink, and S. C. Gielen, "Ambulatory motor assessment in Parkinson's disease," *Mov. Disord.*, vol. 21, pp. 34–44, Jan. 2006.

[5] S. T. Moore, H. G. MacDougall, J. M. Gracies, H. S. Cohen, and W. G. Ondo, "Long-term monitoring of gait in Parkinson's disease," *Gait Posture*, vol. 26, pp. 200–207, Jul. 2007.

[6] A. Salarian, H. Russmann, C. Wider, P. R. Burkhard, F. J. Vingerhoets, and K. Aminian, "Quantification of tremor and bradykinesia in Parkinson's disease using a novel ambulatory monitoring system," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 2, pp. 313–322, Feb 2007.

[7] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato, "Monitoring motor fluctuations in patients with Parkinson's disease using wearable sensors," *IEEE Trans. Info. Tech. Biomed.*, vol. 13, no. 6, pp. 864–873, Nov. 2009.

[8] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: A wearable sensor network platform for high-fidelity motion analysis," presented at the 7th ACM Conf. on Embedded Networked Sensor Systems, CA, 2009.

[9] P. Alexandros and G. B. Nikolaos, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. Syst. Man. Cyber. Part C*, vol. 40, no. 1, pp. 1–12, Jan. 2010.

[10] K. J. O'Donovan, B. R. Greene, D. McGrath, R. O'Neill, A. Burns, and B. Caulfield, "SHIMMER: A new tool for temporal gait analysis," in *Proc. IEEE Eng. Med. Biol. Soc.*, 2009, pp. 3826–3829.

[11] O. Boyaci, A. Forte, S. Abdul Baset, and H. Schulzrinne, "vDelay: A tool to measure capture-to-display latency and frame rate," presented at the 11th IEEE Int. Symp. on Multimedia, San Diego, CA, 2009.

[12] Traffic Shaper XP, Bandwidth Controller Inc. (2007). [Online]. Available: http://bandwidthcontroller.com/trafficShaperXp.html.

[13] S. Fahn and R. L. Elton, "Unified Parkinson's disease rating scale," in *Recent Developments in Parkinson's Disease*, S. Fahn, Ed. Florham Park, NJ: MacMillan Healthcare Information, 1987, pp. 153–163.