Optimal Human-In-The-Loop Interfaces Based on Maxwell's Demon

Kathleen Fitzsimons, Emmanouil Tzorakoleftherakis, and Todd D. Murphey

Abstract— **Interactions with complex systems require safe and reliable interfaces. However, such interfaces must be able to account for substantial uncertainty resulting from the unpredictable nature of human behavior. In this study we demonstrate that a controller/filter unit operating as a "Maxwell's Demon" can be used to synthesize human-machine interfaces that effectively filter user input in real time. Our Maxwell's Demon Algorithm (MDA) was applied in mechanical and software filtering of user actions for the cart-pendulum inversion task. Software filtering was implemented and tested using a custom Android application. Additionally, a haptic device was employed to create a mechanical filter. Results from nine healthy subjects show that both software and mechanical filters increased the success rate of subjects in the swing-up task. This result suggests that the MDA may be applied to design reliable human-machine interfaces for rehabilitation, training, teleoperation and other shared control tasks.**

I. Introduction

Human-machine interfaces have become necessary to ensure safe, reliable interaction with complex systems in activities ranging from driving a car to rehabilitation therapy. Effective interfaces can reduce the cognitive load on a human operator by planning efficient routes, automating obstacle avoidance, managing low level controls of robotics, and filtering input signals [1]–[3]. Interfaces may also provide feedback aimed to improve task performance and training ([3], [4]), even after neurological injury [5]–[7]. However, such interfaces must be able to manage substantial noise and uncertainty that stems from the unpredictable nature of human behavior [8]–[11].

One approach to manage noise and uncertainty has been proposed in [12] on the basis of a philosophical demon suggested by Maxwell to contradict the second law of thermodynamics. The demon selectively opens a molecule-size door in the center of a partitioned box filled with gas in thermal equilibrium. Maxwell's demon allows faster or slower gas particles to move from one side of the box to the other until the two sides of the box are filled with hot and cold gases, respectively [13], [14]. It has been shown that this Maxwell's Demon Algorithm (MDA), can use Gaussian noise to provide sufficient control authority to accomplish dynamic swing-up tasks [12]. This concept can be applied as a controller/filter unit, "a demon", which accepts or rejects user input on the basis of an optimal controller.

This "demon" can be realized through software or by means of mechanical impedances such as a haptic interface. Haptic interfaces are useful for both teleoperation and virtual training, because of the ability to realize many virtual environments on the same device and the ability to easily realize force cues that are not physically realizable. In [15], virtual fixtures were used as perceptual overlays to improve the performance of a pick and place task. Virtual fixtures constrain the position of the operator in the workspace. In [3], this was accomplished by defining virtual walls, modeled as a spring and damper in parallel, providing guidance on the shortest straight line trajectory from one target to the next. Assistive haptic feedback has been shown in many cases to not only enhance task performance, but also to enhance motor training and learning of tasks, including walking, reaching, and path following [16]–[18]. Optimal controllers have also been used to generate haptic feedback signals; results in [19] show that using the output of a Linear Quadratic Regulator (LQR) to create a vibrotactile "teacher" signal improved training outcomes for a dynamical balancing task.

The contribution of this paper is two-fold: we show that the Maxwell's Demon Algorithm in [12] can be effectively used as a means to synthesize reliable human-machine interfaces through a) software and b) mechanical filtering of user interactions with a complex system. Both implementations of the algorithm described in this paper are tested by nine healthy subjects in a cart-pendulum swing-up task. As a software-filtering platform we use a touchscreen, Android phone where the Maxwell's Demon filter is strictly applied in software (in real time). The second implementation (mechanical filtering) utilizes the concept of virtual fixtures from haptics to create a unit that filters subject input mechanically by generating transient virtual walls. Updates of the software and mechanical fixtures are performed in real time according to a receding horizon optimal controller known as sequential action control (SAC) [12], [20]–[23]. Both implementations maximize user effort by providing assistance only when user inputs are incongruous with the optimal controller signal. The use of SAC in this context has the additional benefit of not requiring a reference trajectory such that the user is allowed to find a "good enough" solution.

II. Preliminaries

A. Maxwell's Demon Algorithm (MDA)

The MDA algorithm was proposed in [12] for noise-driven swing-up problems based on the hypothesis that noisy inputs can be a rich source of control authority if filtered in a meaningful way. Our MDA filter was implemented by combining a controller and a filter into a single computational unit that cancels noise samples not driving the system to

Authors are with the Neuroscience And Robotics Laboratory (N×R) at the department of Mechanical Engineering, Northwestern University, Evanston, IL. Email: k-fitzsimons@u.northwestern.edu, man7therakis@u.northwestern.edu, t-murphey@northwestern.edu

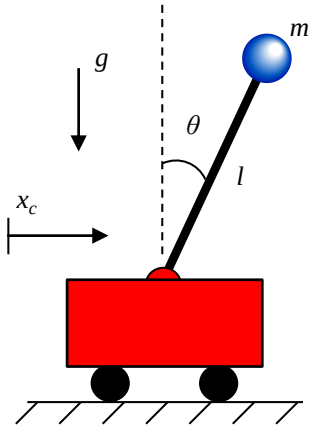

Fig. 1. Schematic of the cart-pendulum system.

the desired control direction. In this section we present a modified version of the MDA algorithm that allows filtering of *user input*. We show that, using this modified version, we can synthesize more reliable interfaces between human operators and complex machines/tasks.

We will now describe Algorithm 1. Given a system and an operator, assume that a user command is generated every t_s seconds. Further, suppose that at each time instance a vector of potential control inputs is computed based on a controller (in this paper we use SAC). Note that the controller should be capable of driving the system by itself according to the desired specifications. If the inner product between the control and the user command vector is positive, and the corresponding angle of the vectors is small, then the effect of user input on the system should be similar to that of the control vector. In that case, if the magnitude of the user command is within the allowed limits, the command is applied to the system. Otherwise, saturation may be applied[1]. On the contrary, if the inner product or angle condition is violated, one of two alternatives can be followed: a) the system input can be set equal to zero (user command is "rejected") or b) the system input can be set equal to the nominal control value. The latter case would result in potentially never-failing interfaces, serving both training and safety purposes. Note that in our experimental setup we followed the first approach; the rationale behind this choice is that being allowed to fail in the task should provide clear indications as to whether the MDA algorithm has any effect on performance. In theory, human-machine interfaces filtered by this approach should drive the system towards the desired control direction; in case the MDA is based on an optimal controller in particular, these directions will improve the selected objective. This process is illustrated in Algorithm 1.

[1]Saturation limits may correspond to physical constraints e.g. angle or torque/force limits etc.

Algorithm 1 MDA approach for filtering user input

- Initialize current time t_0, sampling time t_s, final time t_f, input saturation u_{sat} and angle tolerance γ.

while $t_0 < t_f$ **do**
  Get user input u_{user}
  Compute nominal controller value u_c
  Calculate inner product $\langle u_c, u_{user} \rangle$
  Calculate angle ϕ between u_c and u_{user}

  if $\langle u_c, u_{user} \rangle > 0$ **and** $|\phi| \leq \gamma$ (1)
    if $|u_{user}| < u_{sat}$
      Use u_{user} as current input, $u_{curr} = u_{user}$
    else
      Apply saturated user input $u_{curr} = u_{sat}$
    end if
  else
    Completely "reject" u_{user} $(u_{curr} = 0)$
    or
    Use nominal control instead $(u_{curr} = u_c)$
  end if
  Apply u_{curr} for $t \in [t_0, t_0 + t_s]$
  $t_0 = t_0 + t_s$
end while

B. Sequential Action Control

In this study we use sequential action control (SAC), a recently formulated model-based algorithm for optimal control of nonlinear systems, as a *filter* that rejects user inputs not driving the system to the desired control direction. To achieve the latter, we use the inner product condition described in Algorithm 1. SAC enables rapid, closed-loop constrained control synthesis for broad and challenging classes of systems and objectives. A discussion on local stability is provided in [20]. Our current work focuses on establishing global stability guarantees by applying a terminal cost/region approach as in [24]–[27]. Finally, potential challenges in hardware implementation are discussed in [23]. For a detailed description of SAC, the reader is encouraged to consult [12], [20]–[23].

It must be noted that the MDA algorithm is not SAC-specific; any controller that can successfully control the system of interest can be used instead. For the purpose of this paper, the reader should view SAC simply as a controller that leads to the desired swing-up behavior when directly applied to the tasks in Section III.

III. Materials and Methods

A. Overview

In this experiment we used two testing platforms, i.e. a haptic and a touchscreen platform, to test the proposed MDA approach. Each test platform included an assistance mode,

where the MDA interface was applied, and a no-assist mode. Trials utilized a simulated two-dimensional cart-pendulum system, which subjects were instructed to swing up to the unstable equilibrium (the system was initially resting at the downward stable equilibrium). The equations that describe the underactuated cart-pendulum system (Fig. 1) are given by:

$$\dot{x} = f(x,u) = \begin{pmatrix} \dot{\theta} \\ \frac{g}{l}\sin\theta + u\cos\theta - \frac{b}{ml^2}\dot{\theta} \\ \dot{x}_c \\ u \end{pmatrix} \quad (2)$$

where the state vector x consists of the angular position and velocity of the pendulum and the position and lateral velocity of the cart, $x = [\theta, \dot{\theta}, x_c, \dot{x}_c]$, the input u is the lateral acceleration of the cart, g is the acceleration due to gravity, b is the damping coefficient, l is the pendulum length and m the mass at the tip.

On the haptic platform, subjects held the stylus of a haptic device and kinematically controlled the cart acceleration (and thus position) by moving the stylus in the horizontal plane. The touchscreen platform used a custom Android application running on a Samsung Galaxy Note 4 to generate the cart-pendulum simulation and filter the input signal provided by the subject. Subjects used either a stylus or their index finger to kinematically control the acceleration (and thus position) of the cart in the horizontal plane. The experimental setup is illustrated in Fig. 2. All subjects were tested on both platforms in both no-assist and assistance modes and in a counterbalanced fashion. Because there is only one control input in the task, the control and user input vectors were always parallel and their relative angle, ϕ was either $180°$ or $0°$; thus the parameter γ of Algorithm 1 was set to 0 for both platforms.

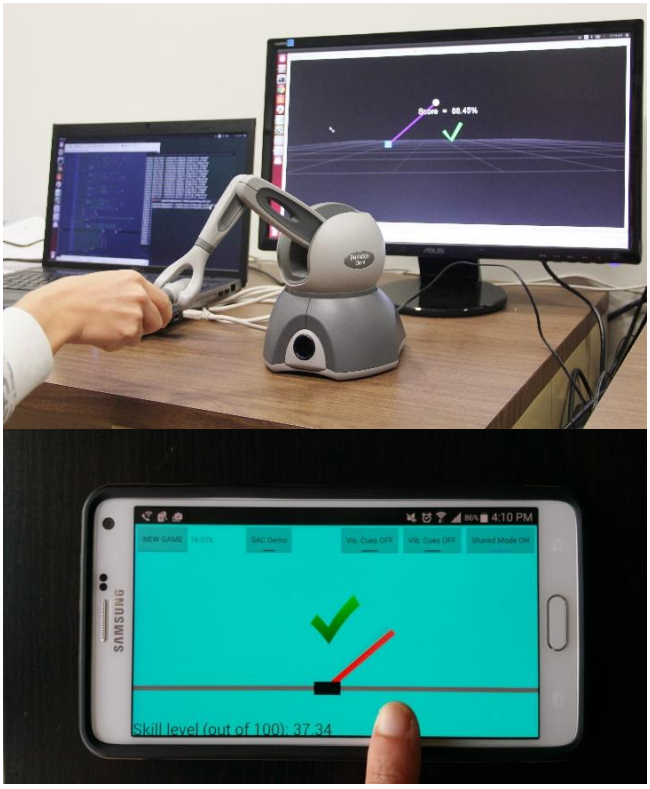

Fig. 2. Experimental setup used for the haptic and touchscreen platforms.

B. Haptic Platform

The haptic platform was comprised of the state sensing and simulation of the cart-pendulum system, and the computation of SAC control actions used to update the haptic feedback law. The position of the cart was given by the horizontal position of the haptic device stylus. The sensed position of the cart along with the simulated angle of the pendulum were used by SAC to compute an optimal action. The feedback law was the addition of a saturation component and a virtual wall, which updated according to the calculated optimal SAC action. The major components of the haptic platform include:

1) *Robot Operating System:* ROS [28] provides the services of a typical operating system such as device drivers and communication between processes. ROS uses distributed framework of process which communicate by passing messages. It is used to integrate the software and hardware components of this experiment and to record trials.
2) *Phantom Omni Haptic Device*: The Phantom Omni (Sensable) is a compact robotic arm with a stylus end-effector. It has six degrees of freedom – three of which are actuated. The Omni has a device driver compatible with ROS, through which the position of the end-effector can be sensed and the force at the end effector can be controlled.
3) *`trep`:* A simulation package called `trep` (available at `nxr.northwestern.edu/trep`) which follows the variational integrator approach [29] to simulate arbitrary mechanical systems was used to simulate the dynamics of the cart-pendulum system.
4) *Haptic Feedback Law:* The amount and direction of force generated by the Omni was determined by summing two feedback terms, i.e. a saturation term and a virtual wall term, aiming to mechanically implement u_{sat} and the inner product/angle conditions (1) of Algorithm 1 respectively. In particular, the saturation term given in (3) creates a zone where no opposing force is generated by the Omni when $\dot{x}_c \in [-4m/s, 4m/s]$, i.e. when the velocity of the cart, as controlled by the subject, is within the specified limits. If the subject moved out of that region, an opposing force of up to $4N$ was applied. The feedback law as given in (4) creates transient virtual fixtures based on the cart/stylus position x_c and velocity $\dot{x}_c$.

$$F_{sat} = -\frac{15}{(1+e^{-1.25(\dot{x}_c-4)})} + \frac{15}{(1+e^{1.25(\dot{x}_c+4)})} \quad (3)$$

$$F_{wall} = \begin{cases} K_p(x_{wall} - x_c) + K_d\dot{x}_c & \text{if (1) is false} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The position of the fixtures, x_{wall}, was determined by the latest SAC value, in agreement with the MDA. In our study we used $K_p = 300\frac{N}{m}$ and $K_d = 50\frac{N}{m/s}$. The graphical interface and state information of the system was updated at 30Hz, while the optimal control action from SAC was computed at 5Hz, updating the position of the virtual wall. The force feedback loop ran at 300Hz, using the most current position and the velocity of the cart/stylus.

C. Touchscreen Platform

The touchscreen platform used a custom Android application running on a Samsung Galaxy Note 4 in landscape mode (the app is available for download at `nxr.northwestern.edu/sites/default/files/files/SACGames.apk`). Subjects use their index finger or the phone's included stylus to move the virtual cart along a line on the screen. The application simulates the system, updates the visualization and calculates the optimal control values using SAC at 30Hz. Unlike the haptic platform, where the MDA filtering was implemented mechanically with the motor-driven force profiles in (3) and (4), Algorithm 1 in this case was implemented in software. In particular, if condition (1) or the saturation limits were violated, user input was completely rejected or saturated respectively[2]. Hence, the Android application captures a kind of infinite actuation scenario, where any action can be rejected. Finally, the u_{sat} value for this platform was set equal to $15 m/s^2$ (recall that subjects kinematically move the cart by controlling its lateral acceleration).

D. Experimental Protocol

Nine subjects (3 males, 6 females) consented to participate in this study which was approved by Northwestern University's Institutional Review Board[3]. Only healthy adult subjects with no prior knowledge of the experimental procedure were allowed to participate. All subjects were tested on the haptic and touchscreen platforms in both no-assist and assistance modes. The order in which each platform and mode combination was tested was randomized to account for any learning effects.

Subjects were introduced to each interface, and the task was demonstrated to them. They also completed a practice "game" on each platform before data collection began. Ten trials were recorded on each platform/mode combination. Trials began with the pendulum resting at the downward equilibrium and subjects were instructed to attempt to swing up the pendulum to the upward unstable equilibrium. Each trial automatically ended when the pendulum angle was between $-0.15 rad$ and $0.15 rad$ and the angular velocity of the pendulum was less than $0.6 rad/s$. This region of the state space was selected on the grounds of being a region of attraction for a linear quadratic regulator (LQR), when applied to this system. If the pendulum did not reach this region within 50 seconds, the trial ended and was recorded as unsuccessful.

[2]In the haptic platform the efficiency of the virtual fixtures depends on the amount of force that can be generated by the haptic device.

[3]This study was reviewed by the NU IRB and approved, though the study was deemed to not be a human subject study since it does not examine any scientific questions about people.

IV. Results

A. Sample Response

The haptic platform force feedback as calculated by the MDA forced changes in the user input. Figure 3 shows a sample response on the haptic platform in assistance mode. A virtual wall was triggered on five occasions in the depicted trial. For instance, around t=4s, when the virtual wall is triggered according to (4), it causes the position of the cart to remain almost constant, while the slope of the velocity curve changes from negative to positive as indicated by the SAC reference signal. It is also evident that the cart velocity remains bounded by the damping like feedback of the saturation constraint in (3). For example observe the generated forces at approximately t=0.55s and t=2.5s when the cart velocity was close to -4m/s and 4m/s respectively. This exhibits how MDA can effectively yield to skilled users while assisting unskilled users.

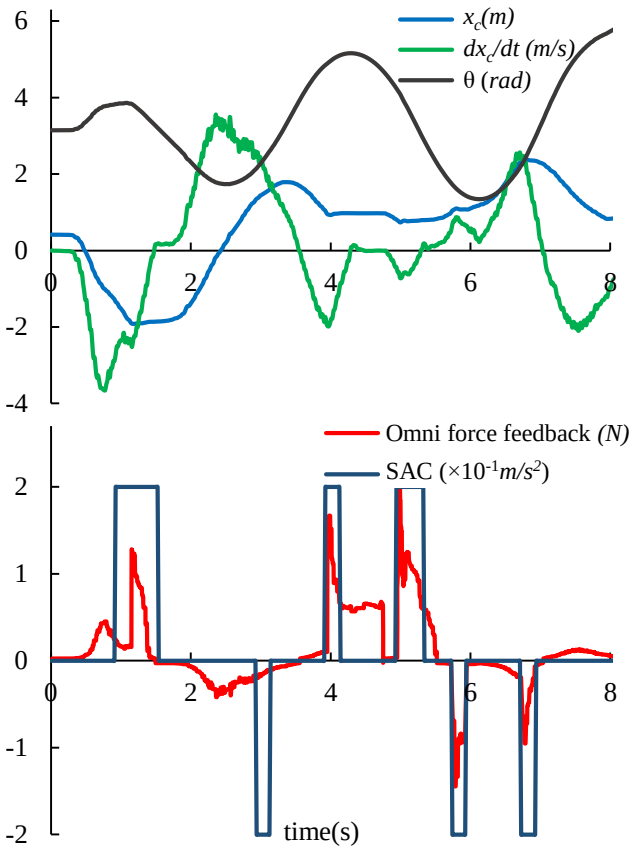

Fig. 3. Sample response of a subject using the haptic platform in assistance mode. Top: the states of the cart-pendulum system. The subject kinematically controls the cart position x_c (and $\dot{x}_c$) through the cart's lateral acceleration. Bottom: The SAC reference signal used in (1) and corresponding force feedback generated by the haptic device (Omni) based on (3) and (4). The impulse-like force response correspond to activation of the virtual wall.

The user inputs on the touchscreen platform were only filtered in software by the MDA. As a result, the real-world motion was not directly affected by the interface.

B. Statistical Results

The primary performance metric used to evaluate subject performance was their success rate, shown in Fig. 4. A trial was considered successful if the pendulum was inverted within the 50s time window. In assistance mode, every subject had a success rate greater than or equal to their unassisted success rate regardless of platform or order of testing. In particular, two paired-sample t-tests were performed on the success rate data of all nine subjects. The assistance mode ($mean$ = 0.96, SD = 0.05) on the touchscreen platform outperformed the no-assist mode ($mean$ = 0.76, SD = 0.26), $t(8)$ = -2.35, $p < 0.05$. On the haptic platform the assistance mode ($mean$ = 0.92, SD = 0.11) was also significantly better than the no-assist mode ($mean$ = 0.84, SD = 0.14), $t(8)$ = -2.80, $p < 0.05$. These results demonstrate that both implementations increased the reliability of the interface, despite the fact that the unassisted success rate was already very high in the no-assist mode (76% and 84% for the two touchscreen and haptic platform respectively).

The time to success (TTS) was a secondary performance metric we used to evaluate the MDA interface. The no-assist and assistance TTS on each platform were compared using two paired-sample t-tests. The mean TTS of the no-assist mode ($mean$ = 23.49, SD = 15.93) and assistance mode ($mean$ = 22.77, SD = 14.48) on the haptic platform were not statistically different, $t(89)$ = 0.34, p = 0.734. However, the assistance mode of the touch screen platform ($mean$ = 16.45, SD = 12.53) produced TTS significantly less than in the no-assist mode ($mean$ = 23.65, SD = 17.59), $t(89)$ = 3.47, $p < 0.005$. These results are illustrated in Fig. 4.

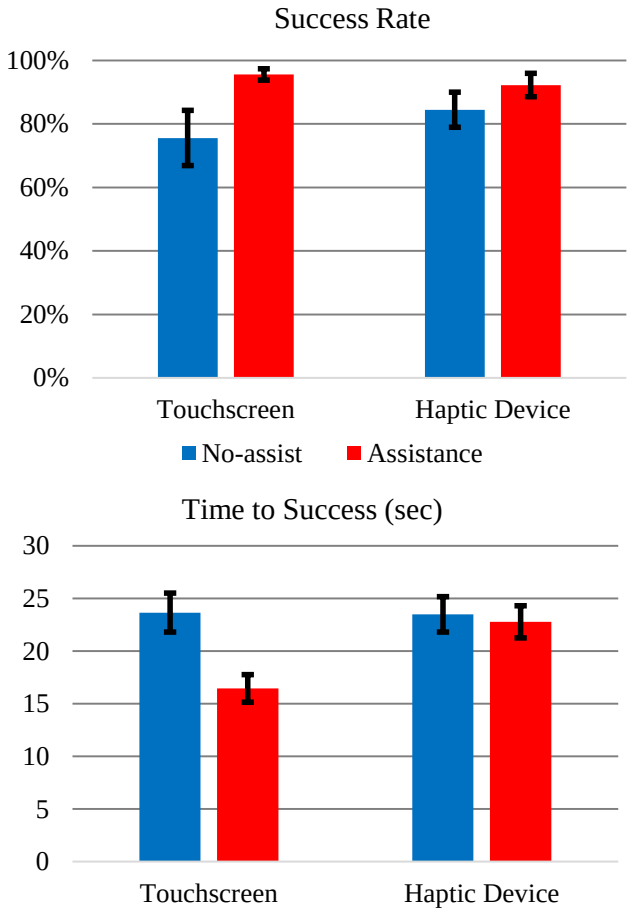

Fig. 4. Summary of results; error bars indicate standard error. Top: Success rate, the primary metric, was significantly higher on both platforms in the assistance mode, even though the average success rate of unassisted trials was already very high (76% and 84% for the two platforms). Bottom: The time to success was significantly lower in the assistance mode of the touchscreen platform. The same was not true on the haptic platform, possibly due to the limited power capabilities of the haptic device which led to less strict enforcement of the MDA as a mechanical filter.

V. Discussion and Concluding Remarks

In this study we used two implementations of MDA as a user input filter, i.e. software filter and mechanical filter, to improve the subjects' performance on a dynamic swing-up task. The touchscreen results with the MDA software filter indicate higher success rate and lower time to success. It was immediately evident during testing that subjects found the task on the touchscreen platform difficult and that the MDA was particularly effective in preventing over excitation of the cart-pendulum system. These results support the findings in previous work where Gaussian Noise provided the input [12].

The second implementation combined the MDA notion of shared control with haptic feedback strategies to synthesize a mechanical filter for control inputs. Although the assistance mode on the haptic platform did increase the success rate, there was no significant difference in time to success between the no-assist mode and the assistance mode. This is likely due to the fact that the haptic interface does not generate enough force to strictly enforce the Maxwell's Demon Algorithm. The efficiency of the transient virtual walls could be improved in future work by utilizing a higher power robotic system. It is also possible that some subjects employed a different strategy than the optimal controller we used, and that these alternative strategies may have also been sufficient to complete the task. However, note that in certain situations, subjects have to follow a specific set of rules. For example, in an actual therapy session, subjects have to adhere to the therapist's understanding of a task. It may be reasonable to expect the same thing, e.g. in software-enabled therapy.

Both platforms demonstrate the utility of MDA in synthesizing reliable human machine interfaces. Perhaps the most important feature of these MDA interfaces is their adaptability in real-time. They require no predefined trajectory, run on an indefinite time horizon, and automatically adapt to operator skill. With only the current state information, MDA can meaningfully reject or impede unhelpful inputs or be completely transparent to operators with significant skill. For complex tasks that require human operators (e.g. walking with an exoskeleton, operating a crane, or flying an aircraft), these features can alleviate or minimize the need for training with virtual simulators by ensuring safety of the physical

system and the operator. For therapeutic applications, an MDA interface may prevent slacking by patients, can relieve frustration, and utilize intentioned but noisy signals (e.g. tremor and spasticity) of patients with neuromotor disorders [5], [8]–[11].

ACKNOWLEDGMENT

This work was supported by the National Science Foundation Graduate Research Fellowship program and Grant CNS 1329891. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] T. Ross and G. Burnett, "Evaluating the human–machine interface to vehicle navigation systems as an example of ubiquitous computing," *International Journal of Human-Computer Studies*, vol. 55, no. 4, pp. 661–674, 2001.

[2] S. Payandeh, "Application of shared control strategy in the design of a robotic device," in *American Control Conference (ACC)*, vol. 6, 2001, pp. 4532–4536.

[3] M. K. OMalley, A. Gupta, M. Gen, and Y. Li, "Shared control in haptic systems for performance enhancement and training," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 1, pp. 75–85, 2006.

[4] M. Yoneda, F. Arai, T. Fukuda, K. Miyata, and T. Niato, "Assistance system for crane operation with haptic display operational assistance to suppress round payload swing," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1999, pp. 2924–2929.

[5] L. Marchal-Crespo and D. J. Reinkensmeyer, "Review of control strategies for robotic movement training after neurologic injury," *Journal of Neuroengineering and Rehabilitation*, vol. 6, no. 1, pp. 20–35, 2009.

[6] R. Sigrist, G. Rauter, R. Riener, and P. Wolf, "Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review," *Psychonomic bulletin 'I&' review*, vol. 20, no. 1, pp. 21–53, 2013.

[7] E. Tzorakoleftherakis, M. Bengtson, F. Mussa-Ivaldi, R. Scheidt, and T. Murphey, "Tactile proprioceptive input in robotic rehabilitation after stroke," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 6475–6481.

[8] B.-C. Chen and H. Peng, "Differential-braking-based rollover prevention for sport utility vehicles with human-in-the-loop evaluations," *Vehicle System Dynamics*, vol. 36, no. 4-5, pp. 359–389, 2001.

[9] J. Tang, Q. Zhao, and R. Yang, "Stability control for a walking-chair robot with human in the loop," *International Journal of Advanced Robotic Systems*, vol. 6, no. 1, pp. 47–52, 2009.

[10] H. Kazerooni, J.-L. Racine, L. Huang, and R. Steger, "On the control of the berkeley lower extremity exoskeleton (bleex)," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005, pp. 4353–4360.

[11] M.-S. Ju, C.-C. K. Lin, D.-H. Lin, I.-S. Hwang, and S.-M. Chen, "A rehabilitation robot with force-position hybrid fuzzy controller: hybrid fuzzy control of rehabilitation robot," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 13, no. 3, pp. 349–358, 2005.

[12] E. Tzorakoleftherakis and T. D. Murphey, "Controllers as filters: Noise-driven swing-up control based on maxwell's demon," in *IEEE Conf. on Decision and Control (CDC)*, 2015, pp. 4368–4374.

[13] A. B. Pippard, *Elements of classical thermodynamics: for advanced students of physics*. Cambridge University Press, 1957.

[14] H. Leff and A. F. Rex, *Maxwell's Demon 2 Entropy, Classical and Quantum Information, Computing*. CRC Press, 2002.

[15] L. B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *IEEE Virtual Reality Annual International Symposium*, 1993, pp. 76–82.

[16] D. S. Reisman, R. Wityk, K. Silver, and A. J. Bastian, "Locomotor adaptation on a split-belt treadmill can improve walking symmetry post-stroke," *Brain*, vol. 130, no. 7, pp. 1861–1872, 2007.

[17] J. L. Patton, M. Kovic, and F. A. Mussa-Ivaldi, "Custom-designed haptic training for restoring reaching ability to individuals with poststroke hemiparesis," *Journal of Rehabilitation Research and Development*, vol. 43, no. 5, pp. 643–656, 2006.

[18] J. Lee and S. Choi, "Effects of haptic guidance and disturbance on motor learning: Potential advantage of haptic disturbance," in *IEEE Haptics Symposium*, 2010, pp. 335–342.

[19] E. Tzorakoleftherakis, F. Mussa-Ivaldi, R. Scheidt, and T. Murphey, "Effects of optimal tactile feedback in balancing tasks: A pilot study," in *American Control Conference (ACC)*, 2014, pp. 778–783.

[20] A. Ansari and T. D. Murphey, "Sequential Action Control: Closed-form optimal control for nonlinear systems," *IEEE Trans. Robot.,* http://nxr.northwestern.edu/publications, In Review.

[21] A. Mavrommati, A. Ansari, and T. D. Murphey, "Optimal control-on-request: An application in real-time assistive balance control," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 5928–5934.

[22] A. Ansari and T. Murphey, "Control-on-request: Short-burst assistive control for long time horizon improvement," in *American Control Conference (ACC)*, 2015, pp. 1173–1180.

[23] E. Tzorakoleftherakis, A. Ansari, A. Wilson, J. Schultz, and T. D. Murphey, "Model-based reactive control for hybrid and high-dimensional robotic systems," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 431–438, 2016.

[24] H. Chen and F. Allgöwer, "Nonlinear model predictive control schemes with guaranteed stability," in *Nonlinear model based process control*. Springer, 1998, pp. 465–494.

[25] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[26] L. Grüne and J. Pannek, *Nonlinear model predictive control*. Springer, 2011.

[27] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[28] (2014) Robot Operating System. Willow Garage. [Online]. Available: http://www.ros.org/

[29] E. R. Johnson and T. D. Murphey, "Scalable variational integrators for constrained mechanical systems in generalized coordinates," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1249–1261, 2009.