Towards Provably Safe Control for Smart Wheelchairs*

Meeko Oishi † **and Nikolai Matni** ‡

Abstract

We propose verification techniques from hybrid control theory to address safety issues in the indoor operation of powered wheelchairs. Verification through hybrid system reachability can provide a mathematical guarantee of safety, where safety is defined as the ability of the system to remain within a desired subset of the state-space. Current efforts are in developing a general algorithm for verification in semi-automated systems, general methods for implementation of control laws for safety, and in the development of an instrumented smart wheelchair testbed at UBC.

Introduction

In Canada, approximately 176,000 seniors with disabilities reside in long-term care facilities as of 2001 (Can 2005). This number will continue to grow as Canada's population inexorably ages. While many of these residents would benefit from access to powered wheelchairs, which can provide more mobility, greater independence, and significantly improve quality of life, unsafe operation can be hazardous to the operator, other residents, staff, and the facility itself.

Driving conditions are a significant factor in many accidents involving powered wheelchairs (Mortenson et al. 2006). Powered mobility users must navigate precisely around fixed obstacles (e.g., carts, custodial equipment) in narrow hallways, through narrow doorways, and around tight corners. In addition, powered mobility users face moving obstacles which include other wheelchair users and walkers, each traveling at a wide range of speeds. Smart wheelchairs that can to reduce or even eliminate collisions or near-collisions with fixed and mobile obstacles have the potential to reduce the liability associated with powered wheelchairs in residential care facilities (Pineau et al. 2003; Cooper 2008; Parikh et al. 2007; Rotenstein et al. 2007; Simpson 2005; Mihailidis et al. 2007).

We aim to improve the safety of powered wheelchair use through verification techniques that can provide mathematical guarantees of safety. Computational techniques for verification (e.g., reachability analysis) can create a new level of confidence and reliability in safety-critical systems such as smart wheelchairs, by predicting where failures might occur, and how human operators can predict them (Oishi et al. 2008). These methods are based in hybrid control theory, in which both continuous dynamics (from physical processes) and discrete dynamics (from the automation's mode-logic) are modeled. Verification provides a mathematical guarantee of safety, where safety is defined as the ability to remain within a desired set in the state-space, despite bounded control authority. These techniques involve quantifying "bad" behavior, then identifying configurations that could lead to the bad region of operation. By eliminating those configurations, safety is ensured. In previous applications to aircraft flight management systems, bad regions represented speeds below which an aircraft could stall (Tomlin et al. 2003).

While repeated simulation can provide intuition about how a given system will behave, results are highly dependent on the particular initial conditions chosen. It is impossible to simulate system trajectories from all initial conditions and for all inputs, and therefore impossible to determine all possible outcomes through simulation. However, reachability analysis can provide information about outcomes from an infinite number of initial conditions within a bounded set. The reachability computation is accomplished by evolving the boundary of the set backwards in time according to the system's dynamics. We draw upon level set methods that can handle general nonlinear systems, such as powered wheelchair dynamics, with bounded inputs and disturbances (Mitchell, Bayen, and Tomlin 2005).

In *semi-automated hybrid systems*, such as smart powered wheelchairs, it is vital that the automation and the user effectively share control of the system. Depending on the specific wheelchair setup, the user can provide either discrete inputs (e.g. pushing buttons, toggling levers) and/or continuous inputs (e.g. joystick control). The user-interface both provides information to the user about the underlying automation, and allows the user to issue input commands to the system. Any potential conflicts between the automation's input and

*This work is supported by an NSERC Discovery Grant, an NSERC USRA, by the CFI Leaders Opportunity Fund, and by the BC Knowledge Development Fund.

†M. Oishi is an Assistant Professor with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada moishi@ece.ubc.ca.

‡N. Matni is a student with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada matni@interchange.ubc.ca.

the user's input must be resolved, and communicated to the user such that the user understands the state of the automation and can easily predict its evolution (Sheridan 1999). Human-automation interaction problems can be difficult to detect through simple inspection (Sarter and Woods 1995; Degani and Heymann 2002; Oishi et al. 2002; Umeno and Lynch 2007), hence our work is in extending verification techniques for standard, fully-automated systems to accommodate semi-automated hybrid systems.

While a highly trained, specialized group of users, such as pilots, may exhibit a great deal of uniformity in their interaction with automation, users of assistive technologies will have significantly more variation in their interaction with automation, depending on their particular physical and/or cognitive disabilities. Our approach therefore is to model the user's intent by specifying broad bounds on what the user can do, rather than trying to model an individual user's or prototype user's dynamics. This allows us to capture the user's intent in a way that is mathematically tractable for hybrid system verification techniques. Standard control techniques assume unlikely extremes (that the human is acting as precisely as an automation, or alternatively, that the human is a disturbance actively driving the system to unsafety). Neither of these captures the types of inadvertent mishaps that occur in many systems.

Recently we have explored how the verification algorithm for semi-automated systems differs depending on the type of user-input (Matni and Oishi 2008). We are currently designing generic algorithms for user-interface design based on our modified application of reachability analysis for safety verification in semi-automated systems, and implementing our computational results onboard actual physical testbeds. We describe our efforts to implement the reachability calculation on a physical system (discretized over space and time) in a manner which upholds the desired safety guarantees.

Reachability analysis for safety verification

We identify "safe" regions of the state space by computing the *reachable set* $\mathcal{W}(t)$, which is the largest controlled invariant set within a given constraint set $\mathcal{W}_0$. For a continuous system $\dot{x} = f(x,u), u \in \mathcal{U} \subseteq \mathbb{R}^m$, $x \in \mathbb{R}^n$, we compute $\mathcal{W}(t)$ in backwards time. The constraint set is encoded implicitly as a level set function $J_0 : \mathcal{X} \to \mathbb{R}$

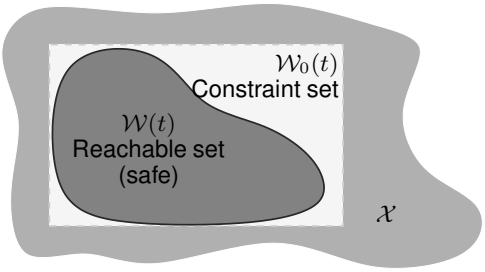
$$\mathcal{W}_0 = \{x \in \mathcal{X} \mid J_0(x) \geq 0\}. \tag{1}$$

then the boundary of the constraint set is propagated backwards in time according to the system dynamics. Finding the backwards reachable set $\mathcal{W}(t)$ requires solving the terminal value time-dependent modified Hamilton-Jacobi (HJ) partial differential equation (PDE) (Tomlin et al. 2003)

$$\begin{aligned} \tfrac{\partial J(x,t)}{\partial t} + \min\left[0, H\left(x, \tfrac{\partial J(x,t)}{\partial x}\right)\right] &= 0 \quad \text{for } t < 0 \\ J(x,0) &= J_0(x) \quad \text{for } t = 0 \end{aligned} \tag{2}$$

with

$$H\left(x, \frac{\partial J(x,t)}{\partial x}\right) = \max_{u \in U} \frac{\partial J(x,t)}{\partial x}^T f(x,u). \tag{3}$$



As shown in (Tomlin et al. 2003), we obtain an implicit representation of the reachable set $\mathcal{W}(t) = \{x \in \mathcal{X} \mid J(x,t) \geq 0\}$ as well as a control law which, if enforced along the boundary of the reachable set, will ensure that the state of the system will never exit $\mathcal{W}(t)$ and therefore never exit $\mathcal{W}_0$. The complete algorithm for hybrid system reachability analysis and controller synthesis is detailed in (Tomlin et al. 2003).

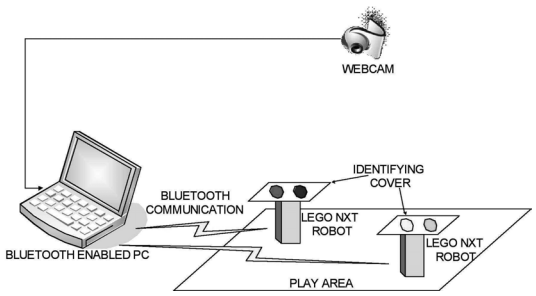
In practice, the computation is accomplished by discretizing the state space into intervals of width Δx and evolving (2) on each of the grid points according to standard numerical integration routines.

Towards implementing safe control laws

The reachability calculation provides a guarantee of safety for states that remain within the reachable set. Once the state exists the reachable set, safety can no longer be guaranteed. Hence in implementation, we cannot allow the system to cross the boundary of the reachable set. In addition, as it is rare to have an analytic solutions for $\mathcal{W}(t)$ we must rely on the computational solution, which is defined only at grid points used in the reachability calculation. In implementing the reachable set for values of x which do not lie on the grid, we choose $\tilde{J}(x,t) = \min\{x_k \in \mathcal{N}(x)\} J(x_k,t)$, with $\mathcal{N}(x)$ the set of neighboring grid points of x, which is the most conservative approximation (e.g., closest to unsafety) of $J(x,t)$. Measurements of the physical system occur every T seconds.

A naive method of preventing boundary crossings, and therefore ensuring safety, is to create a subset of the reachable set $\mathcal{W}(t) \supseteq \mathcal{W}_\epsilon(t) = \{x \in X \mid J(x,t) \geq \epsilon\}$ and to enforce the control law for safety immediately after detecting a boundary crossing of $\mathcal{W}_\epsilon(t)$. However, this requires trial-and-error to find an appropriate value of ϵ, and provides no guarantee that the boundary crossing will be detected for $x \in \mathcal{W}(t) \cap \overline{\mathcal{W}_\epsilon(t)}$ (e.g., it is possible that the boundary crossing might not be detected until $x \in \overline{\mathcal{W}(t)}$). It is also possible that the system could exit and then re-enter $\mathcal{W}(t)$ between time-steps.

Our approach is to detect boundary crossings before they actually occur, by determining restrictions on the spatial discretization used in the reachability calculation, given a fixed sampling interval T. Consider a forward-difference model of integration, in which $x\left((n+1)T\right) = x\left(nT\right) + Tf(x(nT), u(nT))$. In order to ensure that no boundary crossings are missed during sampling intervals, we must choose a spatial discretization Δx for the reachability cal-



culation such that

$$\max_{x \in X, u \in \mathcal{U}} f(x,u) \cdot T \leq \Delta x \tag{4}$$

for all possible values of states x and control inputs u. Such a choice of Δx will prevent boundary crossings of the reachable set in the physical system.

Experimental testbed

A vision-based LEGO NXT robot testbed has been built to test and validate general algorithms for verification of semi-automated systems and implementation of safe control laws. The system operates as either a fully-autonomous or semi-autonomous platform. Users can interact with a variety of GUIs to mimic discrete and continuous user-inputs. This testbed provides an inexpensive, low-risk platform for testing new algorithms.

In addition, we are building a two-wheelchair testbed with fully instrumented powered wheelchairs and indoor GPS for ground-truthing. The wheelchair testbed will enable physical testing and validation of reachability-based algorithms for provably safe control of semi-autonomous powered wheelchairs.

References

2005. Advancing the inclusion of people with disabilities. Technical Report SD23-4/2005E, Government of Canada, Canada.

Cooper, R. 2008. Quality-of-life technology. *IEEE Engineering in Medicine and Biology* 27(2):10–11.

Degani, A., and Heymann, M. 2002. Formal verification of human-automation interaction. *Human Factors* 44(1):28–43.

Matni, N., and Oishi, M. 2008. Reachability-based abstraction for an aircraft landing under shared control. In *Proc of the American Control Conf.*

Mihailidis, A.; Elinas, P.; Boger, J.; and Hoey, J. 2007. An intelligent powered wheelchair to enable mobility of cognitively impaired older adults: An anticollision system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15(1):136–143.

Mitchell, I.; Bayen, A. M.; and Tomlin, C. J. 2005. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control* 50(7):866–872.

Mortenson, W.; Miller, W.; Boily, J.; Steele, B.; Crawford, E.; and Desharnais, G. 2006. Overarching principles and salient findings for including in guidelines for power mobility use within residential care facilities. *Jrnl of Rehabilitation Res. & Dev.* 43(2):199–208.

Oishi, M.; Mitchell, I.; Bayen, A.; Tomlin, C.; and Degani, A. 2002. Hybrid verification of an interface for an automatic landing. In *Proc IEEE Conf. on Decision and Control*, 1607–1613.

Oishi, M.; Mitchell, I.; Bayen, A.; and Tomlin, C. 2008. Invariance-preserving abstractions of hybrid systems: Application to user interface design. *IEEE Trans Control System Technology* 16(2):229–244.

Parikh, S.; Grassi, V.; Kumar, V.; and Okamoto, J. 2007. Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform. *IEEE Intelligent Systems* 22(2):33–41.

Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; and Thrun, S. 2003. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems* 42(3-4):271–281.

Rotenstein, A.; Andreopoulos, A.; Fazl, E.; Jacob, D.; Robinson, M.; Shubina, K.; Zhu, Y.; and Tsotsos, J. 2007. Towards the dream of intelligent, visually-guided wheelchairs. In *Int'l Conf. on Technology and Aging*.

Sarter, N., and Woods, D. 1995. How in the world did we get into that mode? Mode error and awareness in supervisory control. *Human Factors* 37(1):5–19.

Sheridan, T. 1999. Human supervisory control. In Sage, A., and Rouse, W., eds., *Handbook of Systems Eng. & Management*. John Wiley and Sons. 591–628.

Simpson, R. 2005. Smart wheelchairs: A literature review. *Jrnl Rehabilitation Res. & Dev.* 42(4):423–436.

Tomlin, C.; Mitchell, I.; Bayen, A.; and Oishi, M. 2003. Computational techniques for the verification of hybrid systems. *Proc of the IEEE* 91(7):986–1001.

Umeno, S., and Lynch, N. 2007. Safety verification of an aircraft landing protocol: A refinement approach. In Bemporad, A.; Bicci, A.; and Buttazzo, G., eds., *Hybrid Systems: Computation and Control*, LNCS 4416. Springer Verlag. 557–572.