A Shared Control Architecture for Human-in-the-Loop Robotics Applications

Velin Dimitrov[1] and Taşkın Padır[1]

Abstract— **We propose a shared control architecture to enable the modeling of human-in-the-loop cyber physical systems (HiLCPS) in robotics applications. We identify challenges that currently hinder ideas and concepts from cross-domain applications to be shared among different implementation of HiLCPS. The presented architecture is developed with the intent to help bridge the gap between different communities developing HiLCPS by providing a common framework, associated metrics, and associated language to describe individual elements. We provide examples from two different domains, disaster robotics and assistive robotics, to demonstrate the structure of the architecture.**

I. Introduction

It is evident that over the last decade, the emergence of research in human-in-the-loop cyber physical systems (HiLCPS) in a very diverse set of application domains has encouraged innovative approaches to complex problems and challenges. Robots act as physical agents within a HiLCPS and they enable complex interaction with the environment. As a result there is a need to investigate, design, implement and validate novel shared control techniques to enable reliable, robust, and sufficiently agile interaction, communication, and operation in human-in-the-loop robot systems. We posit that there are certain tasks humans are (and will be) superior to robots such as perception, intuitive control, and high-level decision-making, on the other hand, there are tasks robots can (or should) perform such as precise low-level motion planning, solving an optimization problem, and operating in dirty, dull and dangerous situations. Therefore, the investigation of new control interfaces and shared control methods that can effectively delegate tasks and blend the control between the robot and human operator will enable us to field robot systems that act in direct support of humans.

It is possible to classify most robots that are currently deployed in applications in two categories: (i) fully autonomous robots performing specific tasks [1]–[3], and (ii) tele-operated robots with little to no intelligence [4]–[6]. We acknowledge that not all human-robot interaction fall into these two categories but they represent a wide majority of systems that are currently in use. As we attempt to close the gap in between these two classes, new control techniques are needed to dynamically shift the level of control between the human operator and the intelligent robot using dynamical system modeling, stochastic control, probabilistic robotics, physical human-robot interaction, and systems engineering tools.

The development of HiLCPS has also led to the emergence in new challenges when trying to apply traditional tools to solve the associated problems. HiLCPS tend to involve collaborative development efforts with contributions from a wide variety of disciplines. Each discipline brings it own associated set of tools, metrics, and language which make collaboration more difficult than within a discipline. Understanding a key concept in one domain will not be sufficient for researcher in HiLCPS. The tools to encourage cross-domain collaboration are either non-existent or insufficient today.

In addition to a variety of disciplines, HiLCPS commonly span several different scales, from fine granular details at the lowest level to abstract coupling and interactions at the highest levels, at the same time. The flow of information and interfaces between the levels need to be well understood to grasp the operation of the HiLCPS, but without a formal methodology to analyze those interactions the complexities of the systems are lost or ignored as disturbances.

The paper is organized as follows: Section II outlines the challenges we have identified developing HiLCPS, Section III describes the shared control architecture in detail along with interactions between each element, Section IV shows an example application of the architecture for disaster robotics, and Section V highlights developments towards implementing the architecture for assistive robotics.

II. Challenges in HiLCPS

Because of their nature, with abstract elements in the cyber sphere and concrete objects in the physical environment, interactions within the human-in-the-loop CPS require design and analysis through different methods than traditional systems. An additional complexity arises from researchers using an entirely different language to describe the same core principle in a different domain. In [7], the lack of communication between different communities was directly tied to a lack of common language. Ultimately, one of the goals of research in HiLCPS should be to focus on an architecture and associated methodologies to enable and encourage the cross-pollination of ideas, approaches, and solutions between the domains. Many concepts in HiLCPS are shared across domains, but several challenges exist making the jump across domains more difficult and ultimately limiting the potential impact of novel concepts.

At the 2013 IEEE Systems, Man, and Cybernetics Conference held in Manchester, UK, a workshop was held on

[1]Velin Dimitrov is a graduate student with Robotics Engineering Program, and Dr. Taşkin Padir is a faculty member in the Robotics Engineering Program and Assistant Professor of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA, {vdimitrov, tpadir}@wpi.edu

shared control where almost every system presented would be considered a HiLCPS [8]. Application domains included haptic control for more efficient deep sea exploration, control of multiple rovers in a space environment, assistive curve negotiation in vehicles, and assistive robotics to improve quality of life. Modeling these HiLCPS within each domain is a challenge because the systems are complex, information travels through many pathways within the system, information interfaces are not traditional, and the systems tend to be hybrid in nature with very asynchronous communication. As highlighted in the research literature, comparison of these systems is possible within the same application domain, but currently very difficult or impossible in some cases across different domains.

The haptic feedback that works to enable more intuitive curve negotiation in a vehicle [9] may also be beneficial to an operator controlling a space rover worlds away hampered by communication delays and bandwidth limitations. Many of these systems are employing concepts of model-based control and design, where operators are directly interacting with a virtual *model* of the system as opposed to directly with the system. This is a powerful concept, and each domain currently has its own approach to implementing such systems. The concepts are tantalizingly similar between the domains, but divergent enough in implementation to limit their applicability.

In addition, many application domains have similar challenges with respect to performance metrics, optimization, and scalability that can be addressed in a common architecture since the goals are similar across the domains. Performance metrics for traditional systems, such as transient response, bandwidth, settling time, and disturbance rejection have been well studied, but are not necessarily as effective describing HiLCPS because of the challenges outlined above. A set of performance metrics that works well for describing HiLCPS, especially the coupling within the system, would be extremely beneficial. They would allow more direct comparison of HiLCPS across application domains.

The performance metrics would have an added benefit of enabling a better understanding of optimization in HiLCPS. Traditional optimization methods from control theory work well, but are sometimes difficult to implement in HiLCPS because most rely on good system models or good observability in the system. Many HiLCPS, because of complex coupling, are difficult to model and the system may not be fully observable or deterministic, especially with respect to the human operators, and will likely be highly nonlinear. Good performance metrics of the system can encourage the development of new control and optimization methods for HiLCPS.

Finally, most of the HiLCPS systems need to integrate the concept of scalability: both in terms of the quantity of systems within the HiLCPS and the capabilities of those systems. Conditions in many HiLCPS change quickly and the systems need to dynamically adapt to continue operating in the desired mode. The modeling and control methods need to be able to dynamically scale with respect to the number of individual systems that compose the HiLCPS and their capabilities. Systems need to be able to come online and offline within minimal disruption to the HiLCPS. A common architecture can enable the comparison of scaling strategies in different HiLCPS.

According to [10] the future of shared control HiLCPS leverages transparent input interfaces, good context-aware models to enable human intent inference, a well-executed shared governance of the system, modularity/re-configurability, and distributed architectures. We posit that the lack of an architecture that can effectively model cross-domain HiLCPS implementations limits the ability of shared control HiLCPS to leverage the concepts outlined. Such an architecture needs to be developed and designed to be generic enough to be functional in different application domains, but at the same time have specificity allowing the intricacies of specific implementations to be captured.

III. Shared Control Architecture

A shared control architecture needs to be easily adaptable and extensible to grow as research thrusts push in different directions. The architecture should be able to accommodate standard and non-traditional metrics that can be used to evaluate human-in-the-loop HiLCPS and are relevant in cross-domain applications. Concepts such as trust between agents in the system, the performance of the system, the control effort required within the system, and efficiency of control need to be easily quantified and described within the architecture. In addition to these interface-level metrics (metrics between subsystems within the HiLCPS), metrics internal to the subsystems should be available to encourage system-level optimization and model-based control of these HiLCPS. Complex intricacies in the systems can then be easily evident as opposed to being buried in the details and coupling of the individual subsystems.

The structure of the architecture is split into quadrants related to *where* (cyber and physical) each component resides and *who* (human and robot) contributes the information in each component. The components in the cyber realm operate in an abstract area that could be in the internet (on a cloud platform for example), could be on the robot, or could be distributed across several different areas. The cyber realm is not characterized by physical location but the availability of significant computation power and bandwidth within the realm, where heavy computing and intelligence can be easily implemented. The components in the physical realm are associated with the tangible objects of the system such as the on-board computing of the systems and interfaces the operators utilize.

A second categorization splits the architecture between components that directly utilize mostly human-centric information or robot-centric information. The components base their outputs on the information that is available, so this is a natural dichotomy based on how the control within the system is split between the autonomous agents and the human-driven ones. The human robot interface provides an abstraction for the information transfer between the two

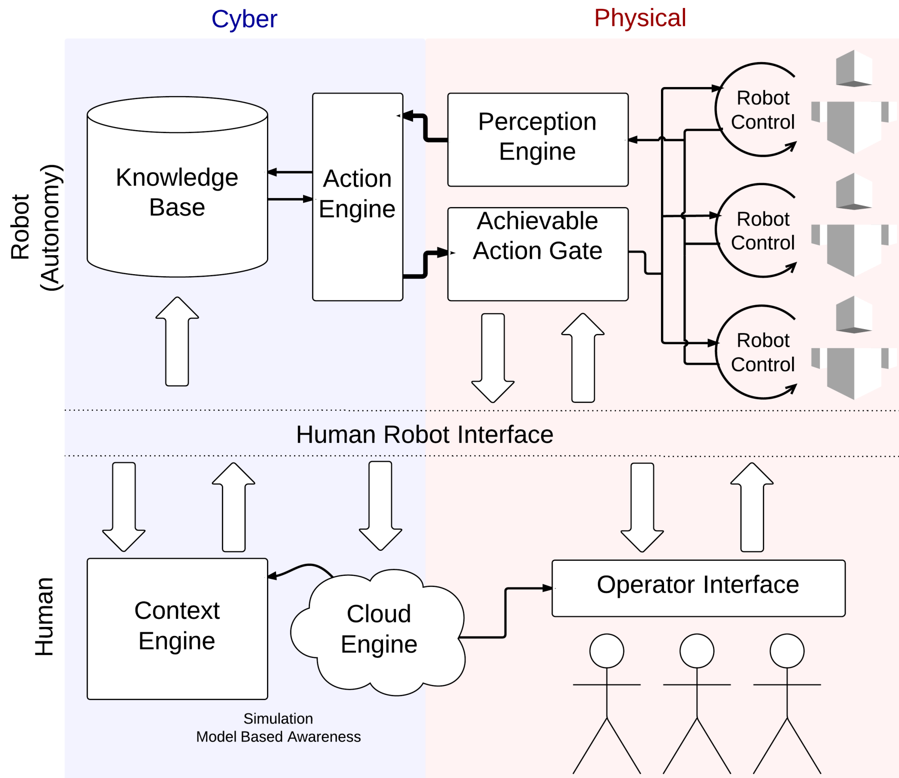

Fig. 1. The shared control architecture consisting of the knowledge base, action engine, achievable action gate, perception engine, low-level robot control, human robot interface, context engine, cloud engine, and operator interface. Not all HiLCPS will necessarily implement all blocks. The architecture is split in both cyber and physical domains, and also human and robot domains depending on *where* elements are located and *who* contributes information to each element.

realms encompassing information generated by the operators and also automatic contextual information from the other elements. It is separate, but includes information from the operator interface. In reality, these boundaries are fuzzy and not clear-cut defined in a real system, but such a categorization helps with the comprehension of information sources and flow within the system.

Figure 1 shows the proposed architecture that corresponds to the features outlined above. The architecture is composed of several elements that we have identified to be common to HiLCPS we are developing in space robotics, disaster response robotics, and assistive robotics at Worcester Polytechnic Institute (WPI): a knowledge base, action engine, perception engine, achievable action gate, robot control, human-robot interface, context engine, cloud engine, and operator interface. The architecture is by no means complete, not all HiLCPS will necessarily include each element, and will expand, consolidate, and change as research development progresses. It should be noted that the robot does not need to be a robot in the traditional meaning of the word. Any autonomous agent would fit the architecture as well.

The *Knoweledge Base* stores global strategies and approaches for the system, providing options to the action engine further down the line. The knowledge base is aware of the history of the action engine, and provides the high-level abstract goal that the system should try to achieve along with which control modality is most appropriate to accomplish the given task. For most systems, the knowledge base operates on an infrequent and asynchronous time frame, changing goals and approach on an event-based principle. Because of this, the knowledge base can store an immense amount of information and has the time to search through it to create a relevant plan for the action engine. In addition to the history of the action engine, the knowledge base receives information from the context engine to adjust global strategy based on the context of the environment, human operator, and system model. As the system operates and encounters various scenarios, the knowledge base stores information on which approach performed well based on the HiLCPS metrics so the system dynamically adapts to changing conditions.

The *Action Engine* takes the global strategy and plan given by the knowledge base and generates a set of potential actions that the robots can take to achieve the tasks. The action engine has access to the information from the perception engine, which provides limited information on the state of the robot, and passes its desired actions to the achievable action gate. The action engine, depending on the HiLCPS, may be completely in the cyber realm or may be infringing in the physical realm as well, especially if it is distributed in terms of computing. The action engine is where the different

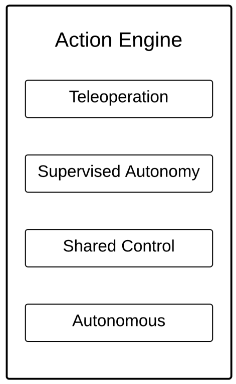

Fig. 2. The action engine provides desired actions for the robots to complete. Various control schemes can be implemented here including teleoperated control, supervised or guarded autonomy, shared control, and autonomous control.

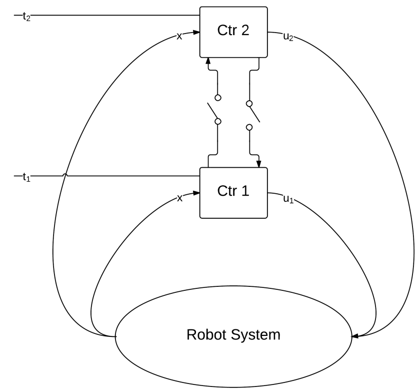

Fig. 3. A proposed architecture for HiLCPS robot controllers. Each robot may have different controllers providing different inputs at different time-scales. In additon, there may be coupling between the controllers as well. The architecture allows for both parallel and series combinations of controllers to be modeled.

control modalities are implemented, as shown in Figure 2. The HiLCPS can be teleoperated, controlled through supervised or guarded autonomy, shared controlled, or fully autonomous. In many situations, the context engine (through the knowledge base) will affect which control modality is most appropriate in a given scenario.

The *Perception Engine* has a high-bandwidth, low-latency link with the robot sensors collecting and sorting the information from those sensors. The primary purpose of the perception engine is to interpret the sensor information and pass along only the information relevant to generating a set of desired actions to the action engine. Any system-wide state estimation (not robot-specific state estimation) should occur in the perception engine since it is the first place that information from all the robots is aggregated. In addition, the perception engine should be able to detect failures in the individual robot so the desired actions can by dynamically adjusted. Methods to detect soft failures, where sensor information is still available but is of questionable credibility, should also be present to arbitrate discrepancies between the individual robots.

The *Achievable Action Gate* takes the desired actions provided by the action gate and checks if they are actually achievable and relevant for the system. For example if the action engine requests a set of actions that are out-of-bounds for a given robot, the action gate will modify those actions so they are as close to the desired ones as possible but within the capabilities of the robot. The action gate is especially important for the teleoperation control modality since the action gate will limit the ability of the operator to accidentally command the robots into potentially dangerous situation either for the environment or the robots themselves. The operator can change the desired actions through the action gate depending on the control modality.

The *Robot Control* block accepts the actions from the action gate and translates them to low-level commands that the motion controllers on each robot can accept and accomplish. We have identified that many HiLCPS have parallel and series control loops, and this is one of the intricacies that should be captured to enable comparison of cross domain HiLCPS. Figure 3 show the proposed architecture for modeling robot controllers. Individual controllers can provide inputs directly to the system or go through another controller, enabling it model both parallel and series combinations of controllers. For simplicity, the diagram shows only two controllers, but it can be scaled depending on the complexity of the system. The switch, which form the parallel series structures, in many cases are set in a permanent state by the specifics of the system, but may be dynamic in certain cases if the system switches architecture or modes. In addition, the controllers do not need to be on the same timescales. t_1 and t_2 provide independent triggers that enable the controllers to run on different timescales.

The *Cloud Engine* receives limited state information about the robot system through the robotic system through the human robot interface and passes the information to the cloud, harnessing the combination of significant human and computational resources, to provide simulations and model-based awareness and control improvements. For example, a simplified model of the robot system can be run in the cloud engine, controlled in simulation through a crowd-sourcing platform, and those results can be passed to the context engine, providing a better sense of the environment, or to the operator interface. The cloud engine, like the action engine, can straddle the grey boundary between the cyber and physical realm with parts in an abstract computation environment and parts implemented on the operator side.

The *Context Engine* receives information from the human-robot interface similar to the cloud engine to understand the environment and situation around the robots. In addition, it can utilize the information from the cloud engine to help decipher the information and context in an accurate and efficient manner. The contextual information is sent to the knowledge base to be recorded and included in the plan later sent to the action engine. This step is critical to enable HiLCPS that are aware of their environment and can dynamically adapt to changing and dynamic conditions.

Finally, the *Operator Interface* serves as the carrier of information between the HiLCPS and the operator or potentially multiple operators. The interface can change based on the information provided by the cloud engine. Model-based awareness algorithms running in the cloud engine can change the perspectives of the information presented on the operator interface. The operator interface provides only a subset of the information flowing through the human robot interface, the context engine can send contextual information directly through as well. All information through the operator interface does not necessarily need to be validated by the operator always, but it is available for introspection if need.

Despite the simple nature of the diagram, we feel this is a preliminary implementation of a very powerful approach to model human-in-the-loop HiLCPS that has potential impacts across the whole range of application domains. It can enable the cross-pollination of ideas and approaches, the ability to compare and contrast ideas in common structure, and allow researchers to describe HiLCPS in a common language. The architecture allows us to compare the approaches we have taken in two different domains at WPI: disaster response robots and assistive robotics. These applications can be described within the proposed architecture, and already implement pieces of the architecture elements described above. We present the current progress of work in these two applications and how the architecture accommodates concepts from both.

IV. DARPA Robotics Challenge

Worcester Polytechnic Institute (WPI) Robotics Engineering C Squad (WRECS) participated in the Defense Advanced Research Projects Agency (DARPA) Robotics Challenge (DRC) Trials as the only Track C team, and became a finalist by ranking 7th out of 16 teams in December 2013. WRECS received the Best-in-Task: Vehicle Award by completing the first phase of the task in about 6 minutes. It is evident that the *driving a utility vehicle* task presents itself as a novel application for humanoid robots, and it is essential to implement reliable supervised autonomy modules on the robot for constrained motion planning and control as well as sufficiently fast perception [11] algorithms. In the overall disaster response mission scenario, *driving a utility vehicle* task comes first. The robot will need to reach the disaster site using a vehicle designed for human-use. As a Track C team, WRECS participated in the DRC Trials with a government furnished ATLAS robot, designed and built by Boston Dynamics specifically for the DRC.

The robot has 28 hydraulically actuated degrees of freedom, so the control modality of direct teleoperation is not feasible. For the driving task, the robot sits in a small utility vehicle and has to actuate the throttle and steering controls. To emulate the restrictions of a real disaster scenario, the communications link between the operators and robot has both severe bandwidth and latency restrictions. We implemented a shared control architecture where the robot autonomously actuates the throttle based on a PI-controller, and the operator commands steering angles implemented through a joint-angle lookup table.

In the architecture, the shared controller is part of the action engine. The controller passes the ankle joint values calculated by a PI controller to actuate the throttle to the achievable action gate. The perception engine analyzes stereo vision and LIDAR data in real-time to estimate the speed of the vehicle. The achievable action gate takes the user desired steering angle, checks that it is valid, and generates the manipulator inputs to move the steering wheel. The knowledge base stores a lookup table of good manipulator trajectories, and the action engine and achievable action gate modify those trajectories to ensure smooth end-effector motions. The arms and feet of the robot are not visible from the sensors during driving. Thus, the cloud engine is used on the operator side to generate a visual model the pose of the robot in the vehicle and provide synthetic situational awareness to the operator. Future work will focus on integrated contextual awareness into the system.

V. Assistive Robotics

A second area of research at WPI that influenced the development of the shared control architecture is the work in assistive robotics and semiautonomous wheelchairs. It is evident through personal interaction with individuals who have physical disabilities, that they are strongly opposed to assistive robotics that are fully autonomous. Trust between the human operator and the robotic system is difficult to cultivate, but it is of paramount importance to bring assistive technologies out of research labs and into the hands of people who need them. Previous work has focused on developing sensor modules to enable semiautonomous control of a wheelchair [12], but further development in higher bandwidth and better fidelity interfaces is required to expand the availability of these systems to people who cannot use traditional interfaces.

One exciting development the architecture can help advance is context-aware navigation and planning algorithms for assistive robots. While this idea has been proposed before, we aim to integrate it into the architecture in the context engine. The interaction between the cloud engine, context engine, and knowledge base is structured to allow the system to infer the users intent before any inputs are even given by the operator. That can mitigate the delays that accumulate with error-prone interfaces such as brain-computer interfaces by providing more tailored options to the operator. In addition, it enables the modeling of operator performance to detect when inputs are deviating from the expected operation in a given scenario. For example, in a typical domestic layout if a user is in the kitchen, the dining room is a very likely next destination, while the guest bedroom is probably not the next destination. The context engine and cloud engine can estimate the probabilities of where the user would like to go next, essentially inferring the next likely action, and tailor the operator interface to make the transition more likely to occur successfully.

VI. Future Work

Future work needs to focus on demonstrating the applicability of the architecture in different application domains, developing a set of metrics specific to HiLCPS, expanding the control modalities, and proving the applicability of the architecture outside the research environment. We have demonstrated the architecture in the applications that inspired its development, disaster robots and assistive robots, but HiLCPS span a vast variety of domains. HiLCPS in each domain will place emphasis on different elements in the architecture and utilize different methodologies which will show the ability to incorporate those differences.

Closely related to expanding to different domains, traditional performance metrics are not necessarily applicable in all HiLCPS, so new metrics that are specific to HiLCPS can be discussed and analyzed using the architecture. Integrating the work from [13], [14] into the architecture would provide an excellent path. For example, traditional controllers are evaluated based on their tracking performance, but with human-in-the-loop systems, humans may be able to handle more tracking error subconsciously in exchange for better intuitive "feel" of a given controller. Some of these metrics will be application-specific, but the ultimate goal is to have a set of metrics that transcends applications so HiLCPS implementations can be easily compared quantitatively. Since human-in-the-loop HiLCPS involve close interaction between humans and robots, a trust metric would be a prime candidate [15].

In Figure 2, we propose four different control modalities that could be part of the action engine. These are control modalities that make sense in disaster robotics and assistive robotics, but there are many other modalities that could be explored. In [16], shared control is broken down into six different modalities. It is conceivable that a system could implement all six modalities and dynamically switch between them depending on the situation. In [17] a similar approach is taken where control is dynamically changed to assist a user. It would be a beneficial extension to the architecture to explicitly model that switching between modalities.

Already proving useful in a research environment, if the architecture is not shown effective in real-world scenarios, that will severely restrict its ability to be adopted across the HiLCPS community.

VII. Conclusion

We have presented a new shared control architecture for modeling human-in-the-loop cyber physical systems. Challenges in cyber physical systems are identified, and the proposed architecture is developed to help mitigate these challenges. The elements and information flow within the architecture is described, and two example applications are described within the context of the architecture.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. 1135854, 1264588, and 1355623. The work related to the DARPA Robotics Challenge is sponsored by Defense Advanced Research Project Agency, DARPA Robotics Challenge Program under Contract No. HR0011-14-C-0011.

References

[1] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. Weghe, "Herb: a home exploring robotic butler," *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10514-009-9160-9

[2] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 5, pp. 1105–1118, Sept 2009.

[3] V. Dimitrov, M. DeDonato, A. Panzica, S. Zutshi, M. Wills, and T. Padir, "Hierarchical navigation architecture and robotic arm controller for a sample return rover," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, Oct 2013, pp. 4476–4481.

[4] A. Bejczy, W. Kim, and S. Venema, "The phantom robot: predictive displays for teleoperation with time delay," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, May 1990, pp. 546–551 vol.1.

[5] C. Lundberg and H. Christensen, "Evaluation of mapping with a teleoperated robot with video feedback," in *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, Sept 2006, pp. 164–170.

[6] M. W. Carey, E. M. Kurz, J. D. Matte, T. D. Perrault, and T. Padir, "Novel eod robot design with dexterous gripper and intuitive teleoperation," in *World Automation Congress (WAC), 2012*, June 2012, pp. 1–6.

[7] J. Burke, R. Murphy, E. Rogers, V. J. Lumelsky, and J. Scholtz, "Final report for the DARPA/NSF interdisciplinary study on human-robot interaction," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 2, pp. 103–112, May 2004.

[8] D. Abbink, T. Carlson, and M. Mulder. SMC2013 workshop on shared control. [Online]. Available: http://cnbisrv02.epfl.ch/ tcarlson/smc2013wsc/

[9] L. Profumo, L. Pollini, and D. A. Abbink, "Direct and indirect haptic aiding for curve negotiation," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, Oct 2013, pp. 1846–1852.

[10] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, "The future of human-in-the-loop cyber-physical systems," *Computer*, vol. PP, no. 99, pp. 1–1, 2012.

[11] C. Rasmussen, K. Yuvraj, R. Vallett, K. Sohn, and P. Oh, "Towards functional labeling of utility vehicle point clouds for humanoid driving," in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, April 2013, pp. 1–6.

[12] R. Desmond, M. Dickerman, J. Fleming, D. Sinyukov, J. Schaufeld, and T. Padir, "Development of modular sensors for semi-autonomous wheelchairs," in *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, April 2013, pp. 1–6.

[13] J. Marble, D. Bruemmer, D. Few, and D. D. Dudenhoeffer, "Evaluation of supervisory vs. peer-peer interaction with human-robot teams," in *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, Jan 2004, pp. 9 pp.–.

[14] J. Saleh and F. Karray, "Towards generalized performance metrics for human-robot interaction," in *Autonomous and Intelligent Systems (AIS), 2010 International Conference on*, June 2010, pp. 1–6.

[15] M. Desai, M. Medvedev, M. Vazquez, S. McSheehy, S. Gadea-Omelchenko, C. Bruggeman, A. Steinfeld, and H. Yanco, "Effects of changing reliability on trust of robot systems," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, March 2012, pp. 73–80.

[16] A. Enes and W. Book, "Blended shared control of zermelo's navigation problem," in *American Control Conference (ACC), 2010*, June 2010, pp. 4307–4312.

[17] T. Carlson, R. Leeb, R. Chavarriaga, and J. del R Millan, "Online modulation of the level of assistance in shared control systems," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, Oct 2012, pp. 3339–3344.

[18] V. Dimitrov and T. Padir, “A shared control architecture for human-in-the-loop robotics applications,” in *RO-MAN, 2014 IEEE*, Aug 2014.