The 2016 AAAI Fall Symposium Series:
Artificial Intelligence for Human-Robot Interaction
Technical Report FS-16-01

From Bot to Bot: Using a Chat Bot to Synthesize Robot Motion

Michael Fischer*, Samir Menon*, Oussama Khatib
Department of Computer Science
Stanford University, Stanford, California 94305
* Authors contributed equally

Abstract

We present Bot to Bot, a system for developers to write voice controlled applications in a high-level language while retaining portability over a variety of different robot hardware platforms. In this paper we describe how Bot to Bot leverages advances in natural language processing and robotic control to take a user's voice command and translate it into a structured intent for the robot through the following intermediate representations: verbal bites, robot assembly, and robot control primitives. Our long-term goal is to find a verbal instruction set for human-robot interaction. We provide our software as open source to encourage future research.

Introduction

An outstanding problem in human-robot interaction is enabling articulated robots with many degrees-of-freedom to accept abstract human commands and react fluidly in real-time (Goodrich and Schultz 2007; Siciliano and Khatib 2008). A variety of input devices have been developed to simplify how humans communicate with such robots. Most, however, require direct human attention to continuously parse sensor feedback for robotic control (Massie and Salisbury 1994; Conti and Khatib 2005; Bark et al. 2008; Jiang et al. 2009; Maheu et al. 2011). Few systems (House, Malkin, and Bilmes 2009) have leveraged a primary human communication modality: natural language. Advances in voice recognition and parsing have led to commercial chat bots (Lyons et al. 2016), which provide an unprecedented opportunity to enable natural language based human-robot interactions at scale and in real-time. Natural language communicated through voice, presents a use case to control the robot without needing to find a remote control device, enabling the off-loading of tasks to the robot (Nielsen 1994; Miller 1968).

Demonstrating that it is feasible to use chat bots–programs that accept natural language commands as input–for fluid real-time human-robot interaction requires overcoming three challenges while guaranteeing human safety: *(i)* creating a software system for bidirectional real-time chat bot to robot communication, *(ii)* selecting a robot control system that can execute a variety of motor tasks while keeping humans safe, and *(iii)* determining a vocabulary for human-robot communication.

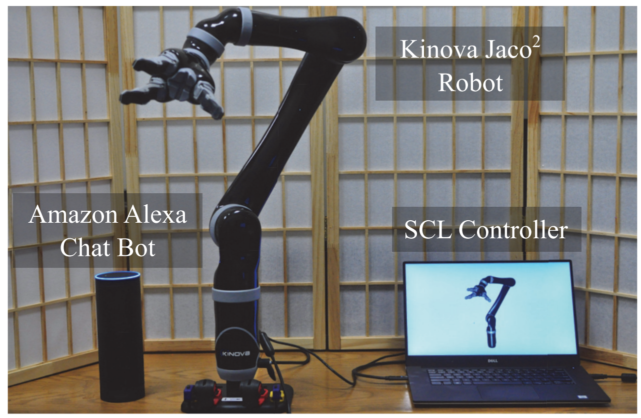

Figure 1: *Voice-Activated Robot Motion Synthesis:* An Amazon Alexa device translates vocal natural language commands to a structured text representation, which is transmitted over the internet to a Redis database. A Standard Control Library (SCL) controller maps text into analog trajectories in task space, and then computes appropriate control torques to actuate a Kinova Jaco^2 robot.

In this paper we present a software system that supports real-time human to chat bot to robot communication with intuitive verbal commands that abstract the complexity of robot control from the developer. The goal of the system is to allow researchers to study voice controlled human-robot interactions at scale by abstracting away the complexity of programming the robots and providing scaffolds to overcome the ambiguity in understanding natural language.

Related Work

Human-robot interaction is a diverse field, from which we review a subset of work that is relevant to our contribution, which is the development of a real-time software system to enable human-robot natural language interaction.

Robot Control Advances in control theory led to the formulation of the multi-task operational space control system, which offers the ability to program human-like motions for arbitrary robots (Khatib 1987; Khatib et al. 2004;

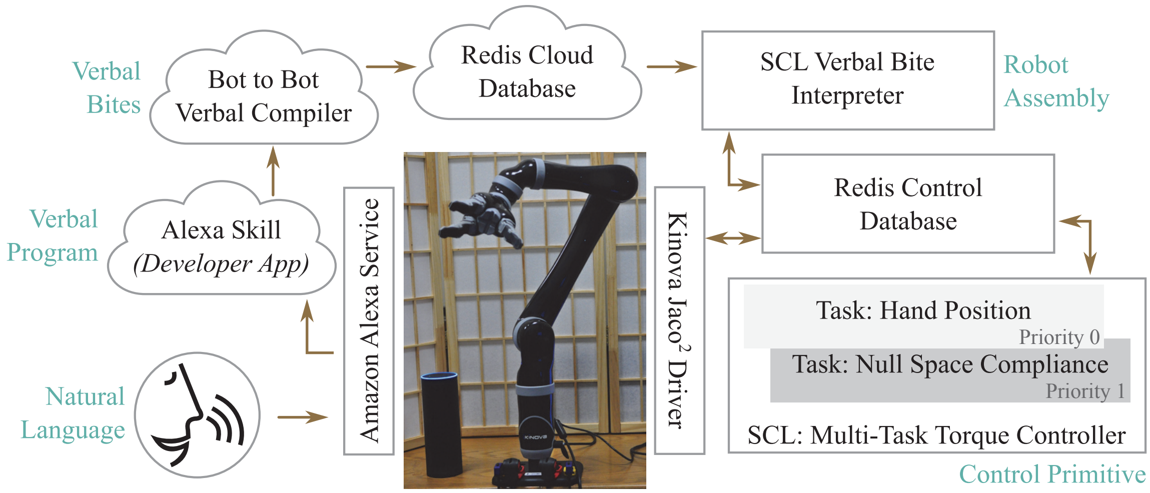

Figure 2: *Compiling Natural Language to Robot Motion:* Information flow in the Bot to Bot system starts with human natural language (bottom left), which is parsed by Amazon's Alexa Service and an Alexa Skill application into a *verbal program*. A Bot to Bot verbal compiler transforms the program to lower level *verbal bites*, which are further transformed to yet lower level *robot assembly* commands by the SCL verbal bite interpreter. Finally, a closed-loop multi-task controller runs a *control primitive*–a set of prioritized tasks with specific goal conditions–that executes the robot assembly code.

Sentis and Khatib 2005; Demircan et al. 2010; Menon et al. 2014). Moreover, a wide variety of control primitives allow autonomous task control for many day-to-day motor tasks in dynamic environments (Ude et al. 2010; Kazemi et al. 2014; Khansari, Klingbeil, and Khatib 2016). In doing so, it eliminates the requirement of humans to explicitly specify robot actuator commands joint-by-joint and thus makes it feasible to engineer robot-agnostic human-robot verbal commands. While operational space control requires torque controlled robots, such robots are now commercially available (for instance, the Kuka LBR iiwa and Kinova Jaco^2) and are poised to become affordable in the near future.

Voice Controlled Articulated Robots An existing system, the VoiceBot (House, Malkin, and Bilmes 2009), uses human non-verbal audio cues including varying pitch, vowel quality, and amplitude, to control an articulated robot. While the VoiceBot's continuous commands are useful for short-range maneuvering, they require constant human attention. In addition, the robot's inverse kinematic control system leads to stiff robot configurations, which are not human-safe in the event of a collision. A different system worked towards basic voice driven motion commands for articulated robots (Chatterjee et al. 2005), and did not demonstrate human safety either. Other systems focused on mobile robots (Liu et al. 2005), avoiding the complexity of articulated robots. These systems, however, did not demonstrate an ability to scale to complex natural language communication. In addition, they did not leverage recent advances in control that promise to dramatically improve robot autonomy and skill.

Natural Language Processing in Robotics Efforts to realize natural language human-robot interaction have focused on mitigating the effect of ambiguity in human speech. Humans, for instance, might interact differently with robots when compared to humans (Strait, Canning, and Scheutz 2014), which could necessitate structured communication that relies on multiple interaction cues (Schermerhorn et al. 2006). Other research attempted to sidestep such problems by parsing formal as well as indirect human requests (Briggs and Scheutz 2013), for which chat bot programs (Fischer and Lam 2016) could be beneficial. Yet other research identified when robots can not execute human commands, and studied responses beyond a robot stall (Scheutz et al. 2007).

Brain-Computer Interfaces for Robot Control Brain-controlled interfaces have been explored as a means of robot control for people with disabilities and degenerative diseases (see (Mak and Wolpaw 2009) for an overview). Non-invasive control methods use electrodes positioned on the scalp to measure electrical activity in the brain (EEG) (Shenoy et al. 2006; Li et al. 2008). Invasive methods, in contrast, use surgical implants to record electrical activity from within the brain (Santhanam et al. 2006; Hochberg 2012; Collinger et al. 2013; Wodlinger et al. 2015; Aflalo et al. 2015). While non-invasive methods are preferable, they often produce low signal-to-noise measurements. Both non-invasive and invasive methods, however, presume that users have limited mobility, robots have limited autonomy, and bulky equipment is acceptable. Both approaches stand to benefit from general advances in natural language driven robot control, which could improve the efficacy of human-robot interaction by augmenting the former's granular control with high-level intentions.

System for Bot to Bot Communication

The Bot to Bot system aims to simplify human-robot natural language interaction by following a principled approach to software engineering (Fig. 2). The set of abstraction layers (Fig. 3) it creates rely on principles that we will now define.

Instantiation	*Stage*
"Alexa tell the robot to wake only me up at 7:00 am."	Natural Language
`{ "intent": "wake", "slots": { "time": "07:00" }, "code": ...setTimeout()... }`	Verbal Program
`{ "intent": "wake", "action": ["poke", "poke"] }`	Verbal Bite
`{ "action": "poke", "axis": "x", "motion": "sin(t)" }`	SCL Robot Assembly
`{ "Pri0": "Hand Position Task", "Pri1": "Null Space Compliance" }`	Control Primitive
Robot Arm Movement	Multi-task Controller

Figure 3: *Instance of Robot Motion Synthesis:* An example demonstrates a possible sequence of transformations starting with a natural language sentence and ending in an articulated robot arm's movement. Intermediate objects are specified using JavaScript Object Notation (JSON). The figure uses pseudocode in places to simplify the JSON.

Robot Motion Synthesis Pipeline

With the Bot to Bot software system, we abstracted computations into distinct layers with explicitly defined intermediate interfaces. Building upon a specific instance that exemplifies how the software operates (Fig. 3), we now discuss the computations in detail.

Natural Language to Verbal Program We translated natural language to verbal programs using Amazon's Echo device. Echo's far-field voice recognition allows it to clearly detect voices and parse natural language. Programming interfaces provided by Amazon allow developers to use Alexa, the standard voice recognition interface, and engineer programs–Alexa skills–that realize real-time natural language interaction. Using Amazon's toolchain, we specified an interface to map one or more users' spoken input into an intent that could be translated into robot actions.

Since natural language provides users many ways to convey identical intent, we programmatically mapped each intent to a set of sample *"utterances"* (for exemplars, see Table. 1) that capture different ways to convey the same intent (for exemplars, see Table. 2). The mapping was then used to train a representational linguistic model. The utterances thus provide a structured method for accepting numerous natural language sentences that relate to a given intent. As such, increasing the set of utterances improves the linguistic model.

```
WAKE wake me up at {Time}
WAKE push me out of bed at {Time}
WAKE make sure I'm awake at {Time}
```

Table 1: *Utterance Exemplars* Utterances are provided by the developer to train the linguistic model. When the user speaks they don't have to exactly match a sample utterance. However, the more utterances provided the more accurately the system is able to determine the user's intent. Utterances have slots, {Time}, in the case above. A slot is a placeholders where Alexa can find a parameter needed to run the verbal program.

```
{
    "intent": "WakeHuman",
    "slots": [{
        "name": "Time",
        "type": "AMAZON.TIME"
    }]
}
```

Table 2: *Intent Schema Exemplars* Utterances are provided by the developer to train the linguistic model. When the user speaks they do not have to exactly match a sample utterance. However, the more utterances that are provided the more accurately the system is able to determine the user's intent. Utterances have slots, {Time}, in the case above. A slot is a placeholders within the utterance where Alexa can find a parameter needed to run the verbal program.

To simplify generalization, Alexa delineates intent from the parameters that instantiate a specific action. For instance, from our example (see Fig. 3), the intent *"wake"* also requires a parameter *"07:00"* to instantiate an executable verbal program. We associated utterances with *"slots"*, or placeholders where Alexa can expect to find parameters for utterances. Each slot has a type and is specified with a list of possible *"slot entities"* that can fill the slot, allowing rich natural language communication. Furthermore, the developer does not have to list all the possible values that a slot can accept; Alexa can decipher synonyms and associate them with slot entities. Inputs to a slot are, however, biased towards being the values that the developer provides. A number of slot types such as date, duration, number, time, US city, and US state come predefined, and this set is poised to grow with Alexa's growing popularity.

Having defined utterances and slots, and built a linguistic model, we could receive spoken instructions and convert them to JSON to be passed to the next layer. Future efforts will build upon on our preliminary implementation for robots and expand our repertoire of human-robot natural language skills.

Verbal Program to Verbal Bites Bot to Bot applications allow software developers to write robot-independent software by abstracting away calls to hardware. Translating abstract verbal programs into verbal bites, however, does require ensuring that the programs can be feasibly realized

"Alexa tell the robot to wave to visitors at the door."
"Alexa tell the robot to poke me when I get a friend request."
"Alexa tell the robot to wave when the car is here."
"Alexa tell the robot to wake me at 7:00 am."
"Alexa tell the robot to scratch my back now!"

Table 3: *Verbal Programs Exemplars* Possible programs that a developer could build and run without needing detailed knowledge of motor control or the robot specific hardware of the robot.

on the lower level robotic hardware. Since different robots possess different capabilities, we require verbal bites to be drawn from a set that is feasible to realize. As such, the verbal bites must always directly map to robot capabilities. To keep the process robot-agnostic, a suitable compiler would issue errors on a mismatch.

Verbal Bites to SCL Robot Assembly While verbal bites can be translated into lower level robot capabilities, the existence of classes of robots with similar abilities–six degree-of-freedom manipulators, for instance–led us to create an intermediate layer to specify robot abilities in a robot-agnostic manner. These abilities, specified as *"robot assembly"* commands, can be used to categorize robots into capability categories instead of analytical kinematic and dynamic ability based categories. We believe this distinction will be valuable for software systems that program robots. Formalizing robot assembly for commercially available robots will constitute valuable future research that promises to allow high-level language programs that retain portability over a variety of different robots.

SCL Robot Assembly to Control Primitives Translating robot assembly commands into controllers that can actuate a robot requires selecting control tasks that can correct errors associated with the assembly commands. The problem of identifying one or more suitable motion and force tasks, however, remains an open challenge for robotics. Recent developments in identifying control primitives (see related work for a detailed discussion), however, afford a finite but steadily growing set of capabilities. While many existing control primitives are not robot-agnostic, it is feasible to reduce them to robot-agnostic tasks using the operational space control formulation. However, since this is an active research area, we do not require control primitives to be completely robot-agnostic and instead allow them to be engineered for specific robot capability categories.

Control Primitives to Multi-task Controller The final stage in our robot motion synthesis pipeline is to construct a multi-task operational space controller that controls a given robot to perform the desired action. Having defined the action with one or more control primitives, we may yet decide to add robot-specific tasks and filters to the final controller. These are expected to be designed by control engineers who are familiar with specific robots and should not need intervention from higher level programmers.

Pat
Move
Wave
Poke
Scratch

Table 4: *Verbal Bite Exemplars* Verbal bites serve as basis functions that simplify software development and encourage code reuse. By allowing software developers to interact with hardware in a safe and scalable manner, they provide a suitable level of abstraction for many applications.

poke()
rotate(98)
grasp(true)
applyForce(10.2)
insert(usb, port3)
moveHand(9, 2, 7)

Table 5: *Robot Assembly Exemplars* Robot assembly commands are robot-agnostic motion and force control specifications that realize control primitives when combined with a task-space controller. For example, if there was a robot produced by Company A and another by Company B, they'd both run the same robot assembly code.

Control Formulation

Task Control Overview

SCL implementes the operational space formulation for robot control, which realizes dynamically consistent torque control for one or more control tasks. The dynamics of an articulated robot may be described using the equations of motion derived from the Euler-Lagrange equations:

$$A(q)\,\ddot{q} + b(q,\dot{q}) + g(q) = \Gamma. \tag{1}$$

where Γ is the generalized external force, $\ddot{q}$ are the generalized accelerations, A is the generalized inertia matrix, b is the vector of centrifugal and coriolis forces, and g is the gravitational force in generalized coordinates. These dynamics may be projected into task space using the dynamically consistent generalized inverse of the task Jacobian:

$$\begin{aligned} \bar{J_x}^T[A(q)\,\ddot{q} + b(q,\dot{q}) + g(q) &= \Gamma] \quad (2)\\ \Lambda(q)\,F_x^* + \mu(q,\dot{q}) + p(q) &= F_x, \quad (3) \end{aligned}$$

where the Jacobian inverse is given by $\bar{J_x}^T = (J_x A(q)^{-1} J_x^T)^{-1} J_x A(q)^{-1}$ for a specific task space x (Whitney 1972; Khatib 1995). This ensures that any applied control forces will do work only in task coordinates.

Implementation Details

We controlled the Kinova Jaco^2 robot with two prioritized tasks in SCL. At the highest priority, we used a Euclidean-space operational point control task that accepted motion trajectories along the x-, y-, and z-axis:

$$F_x^* = k_p(x_{des} - x_{curr}) - k_v \dot{x}_{curr}, \tag{4}$$

where k_p and k_v are the proportional and derivative gains of a PD controller, x_{curr} and $\dot{x}_{curr}$ are the present end-effector position and velocity, and x_{des} is the desired goal position. The desired goal position was set through voice commands. Since the robot was programmed to avoid high speed motions, we ignored the centrifugal and coriolis forces ($\mu(q, \dot{q})$). We used Kinova's estimates for the gravity vector ($p(q)$). The task's final contribution to the generalized forces was given by:

$$\Gamma_{t0} = J_x^T(\Lambda(q)\, F_x^*). \tag{5}$$

In the null space of the operational point task, the second priority level, we also specified a null-space damping task. This task simulates viscous friction along axes of motion that are irrelevant to the task at hand:

$$\Gamma_{t1} = (I - J_x \bar{J}_x{}^T)(-k_v \dot{q}_{curr}), \tag{6}$$

where Γ_{t1}, the task's contribution to the generalized forces, is projected into the null space of the first task.

The composite robot torque command was:

$$\Gamma_{commanded} = \Gamma_{t0} + \Gamma_{t1}. \tag{7}$$

The two tasks allowed real-time control of the end-effector in Euclidean-space while maintaining compliance along the null space, which improves the robot's robustness to external perturbation and decreases impact forces in the event of a human collision.

Conclusion

Recent advances in natural language processing and robotic control created an opportunity to engineer a system that translates abstract natural language commands into tangible robot motions in real-time. We capitalized upon this opportunity to develop such a system and demonstrate that it works. Future research will generalize this idea to develop a framework for natural language based human-robot interaction.

As computers become ubiquitous, we see a world where human-computer interactions go beyond the touchscreen, where we interface with the devices around us in more human ways. With Bot to Bot, we start the conversation. Bot to Bot gives the next generation of application designers, without needing to expand their skill set, the ability to tackle the complex design problems of tomorrow, today.

Appendix

Robot Hardware and Control System We used a Kinova JACO^2 arm with six degrees-of-freedom (DOF) and a KG-3 gripper using the operational space formulation implemented in the Standard Control Library (SCL) (Menon 2011). The SCL multi-task controller received reference trajectory updates from the verbal bite interpreter asynchronously, and, in turn, computed and sent control torques to the low-level driver asynchronously. A low-level robot driver monitored joint angles and velocities and relayed torque control commands at a rate of 500Hz over USB.

Amazon Echo The Echo hardware includes a Texas Instruments DM3725 ARM Cortex-A8 processor, 256MB of LPDDR1 RAM and 4GB of storage space. Echo has seven microphones and beam forming technology to allow it to have directional listening. We used an Amazon Echo device running firmware version 3389.

Amazon Lambda The Alexa Skills (*Developer Apps*) is run in the cloud without having having to provision or manage servers using a service called Amazon Lambda. The code for the skills is entered into a web browser and an environment is chosen (Python, Node, or Java). Logs are piped back through the browser for testing and debugging.

References

Aflalo, T.; Kellis, S.; Klaes, C.; Lee, B.; Shi, Y.; Pejsa, K.; Shanfield, K.; Hayes-Jackson, S.; Aisen, M.; Heck, C.; Liu, C.; and Andersen, R. A. 2015. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science* 348(6237):906–910.

Bark, K.; Wheeler, J. W.; Premakumar, S.; and Cutkosky, M. R. 2008. Comparison of Skin Stretch and Vibrotactile Stimulation for Feedback of Proprioceptive Information. In *2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 71–78. IEEE.

Briggs, G. M., and Scheutz, M. 2013. A hybrid architectural approach to understanding and appropriately generating indirect speech acts.

Chatterjee, A.; Pulasinghe, K.; Watanabe, K.; and Izumi, K. 2005. A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems. *IEEE Transactions on Industrial Electronics* 52(6):1478–1489.

Collinger, J. L.; Wodlinger, B.; Downey, J. E.; Wang, W.; Tyler-Kabara, E. C.; Weber, D. J.; McMorland, A. J.; Velliste, M.; Boninger, M. L.; and Schwartz, A. B. 2013. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet* 381(9866):557 – 564.

Conti, F., and Khatib, O. 2005. Spanning large workspaces using small haptic devices. *Proc. of the Eurohaptics on Haptic Interfaces for Virtual Environment and Teleoperator Systems* 183–188.

Demircan, E.; Besier, T.; Menon, S.; and Khatib, O. 2010. Human motion reconstruction and synthesis of human skills. In Lenarčič, J., and Stanisic, M., eds., *Advances in Robot Kinematics*. Springer, Berlin, Heidelberg, Germany,. 283–292.

Fischer, M., and Lam, M. 2016. From books to bots: Using medical literature to create a chat bot. In *Proceedings of the First Workshop on IoT-enabled Healthcare and Wellness Technologies and Systems*, IoT of Health '16, 23–28. New York, NY, USA: ACM.

Goodrich, M. A., and Schultz, A. C. 2007. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.* 1(3):203–275.

Hochberg, L. R. e. 2012. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature* 485(7398):372–375.

House, B.; Malkin, J.; and Bilmes, J. 2009. The voicebot: a voice controlled robot arm. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 183–192. ACM.

Jiang, L.; Cutkosky, M. R.; Ruutiainen, J.; and Raisamo, R. 2009. Using Haptic Feedback to Improve Grasp Force Control in Multiple Sclerosis Patients. *IEEE Transactions on Robotics* 25:593–601.

Kazemi, M.; Valois, J.-S.; Bagnell, J. A.; and Pollard, N. 2014. Human-inspired force compliant grasping primitives. *Autonomous Robots* 37(2):209–225.

Khansari, M.; Klingbeil, E.; and Khatib, O. 2016. Adaptive human-inspired compliant contact primitives to perform surfacesurface contact under uncertainty. *The International Journal of Robotics Research*.

Khatib, O.; Warren, J.; Sapio, V. D.; and Sentis, L. 2004. Human-like motion from physiologically-based potential energies. *Advances in Robot Kinematics* 149–163.

Khatib, O. 1987. A unified approach for motion and force control of robot manipulators: The operational space formulation. *Robotics and Automation, IEEE Journal of* 3(1):43–53.

Khatib, O. 1995. Inertial properties in robotic manipulation: An object-level framework. *International Journal of Robotics Research* 14(1):19–36.

Li, Y.; Wang, C.; Zhang, H.; and Guan, C. 2008. An eeg-based bci system for 2d cursor control. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2214–2219. IEEE.

Liu, P. X.; Chan, A. D. C.; Chen, R.; Wang, K.; and Zhu, Y. 2005. Voice based robot control. In *2005 IEEE International Conference on Information Acquisition*.

Lyons, G.; Tran, V.; Binnig, C.; Cetintemel, U.; and Kraska, T. 2016. Making the case for query-by-voice with echoquery. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, 2129–2132. New York, NY, USA: ACM.

Maheu, V.; Archambault, P. S.; Frappier, J.; and Routhier, F. 2011. Evaluation of the jaco robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In *2011 IEEE International Conference on Rehabilitation Robotics*, 1–5.

Mak, J. N., and Wolpaw, J. R. 2009. Clinical applications of brain-computer interfaces: Current state and future prospects. *IEEE Rev Biomed Eng* 2:187–199.

Massie, T., and Salisbury, J. 1994. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, volume 55-1, 295–300. IOS Press.

Menon, S.; Fok, S.; Neckar, A.; Khatib, O.; and Boahen, K. 2014. Controlling articulated robots in task-space with spiking silicon neurons. In *Biomedical Robotics and Biomechatronics (2014 5th IEEE RAS EMBS International Conference on)*, 181–186.

Menon, S. 2011. The Standard Control Library version 0.97. http://samirmenon.org/scl.html.

Miller, R. B. 1968. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), 267–277. New York, NY, USA: ACM.

Nielsen, J. 1994. *Usability engineering*. Elsevier.

Santhanam, G.; Ryu, S. I.; Yu, B. M.; Afshar, A.; and Shenoy, K. V. 2006. A high-performance brainâ??computer interface. *Nature* 442(7099):195–198.

Schermerhorn, P.; Kramer, J.; Brick, T.; Anderson, D.; Dingler, A.; and Scheutz, M. 2006. Diarc: A testbed for natural human-robot interactions. In *In Proceedings of AAAI 2006 Robot Workshop*. AAAI Press.

Scheutz, M.; Schermerhorn, P.; Kramer, J.; and Anderson, D. 2007. First steps toward natural human-like hri. *Autonomous Robots* 22(4):411–423.

Sentis, L., and Khatib, O. 2005. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics* 2(4):505–518.

Shenoy, P.; Krauledat, M.; Blankertz, B.; Rao, R. P.; and Müller, K.-R. 2006. Towards adaptive classification for bci-part of the 3rd neuro-it and neuroengineering summer school tutorial series. *Journal of neural engineering* 3(1):R13.

Siciliano, B., and Khatib, O. 2008. *Springer handbook of robotics*. Springer Science & Business Media.

Strait, M.; Canning, C.; and Scheutz, M. 2014. Let me tell you! investigating the effects of robot communication strategies in advice-giving situations based on robot appearance, interaction modality and distance. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, 479–486. New York, NY, USA: ACM.

Ude, A.; Gams, A.; Asfour, T.; and Morimoto, J. 2010. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics* 26(5):800–815.

Whitney, D. E. 1972. The mathematics of coordinated control of prosthetic arms and manipulators. *ASME. J. Dyn. Sys., Meas., Control* 94(4):303–309.

Wodlinger, B.; Downey, J. E.; Tyler-Kabara, E. C.; Schwartz, A. B.; Boninger, M. L.; and Collinger, J. L. 2015. Ten-dimensional anthropomorphic arm control in a human brainmachine interface: difficulties, solutions, and limitations. *Journal of Neural Engineering* 12(1):016011.

Acknowledgments

We thank Vinay Sriram for assisting with programming for the Kinova Jaco^2 redis driver. The project was supported by a National Science Foundation National Robotics Initiative grant (NRI-1427396, O. Khatib and R. Bajcsy), and by a Stanford Neurosciences Institute Big Ideas grant (O. Khatib e.a.).