Sharing the Load: Human-Robot Team Lifting Using Muscle Activity

Joseph DelPreto† and Daniela Rus†

***Abstract*— Seamless communication of desired motions and goals is essential for enabling effective physical human-robot collaboration. In such cases, muscle activity measured via surface electromyography (EMG) can provide insight into a person's intentions while minimally distracting from the task. The presented system uses two muscle signals to create a control framework for team lifting tasks in which a human and robot lift an object together. A continuous setpoint algorithm uses biceps activity to estimate changes in the user's hand height, and also allows the user to explicitly adjust the robot by stiffening or relaxing their arm. In addition to this pipeline, a neural network trained only on previous users classifies biceps and triceps activity to detect up or down gestures on a rolling basis; this enables finer control over the robot and expands the feasible workspace. The resulting system is evaluated by 10 untrained subjects performing a variety of team lifting and assembly tasks with rigid and flexible objects.**

I. Introduction

Robots have the potential to provide humans with valuable assistance and greatly increase productivity, yet there is often a communication barrier when trying to collaborate on physical tasks. To facilitate natural interactions and efficient teamwork, an interface is needed that allows the robot to accurately interpret the person's intentions or commands in a way that minimally distracts from the ongoing task.

Since a person naturally generates muscle activity during physical interactions, detecting these signals via surface electromyography (sEMG) could provide valuable information about desired motion or stiffness. However, such signals are typically dominated by noise and can be difficult to map to limb motions due to the highly nonlinear relationships between electrical neuron activity, muscle activation, joint impedances or torques, and limb dynamics. In addition, predicting effective robot actions in response to known human actions can be difficult and highly task-dependent.

An approach to address these challenges could be blending motion prediction from natural muscle activity with a muscle-based control interface for explicitly commanding adjustments. The method presented in this work creates such a controller for team lifting tasks by using EMG signals from the biceps and triceps as the only inputs, as depicted in Figure 1. A coarse estimation of the person's upward or downward motion is calculated from biceps activity and used to control the collaborating robot; the person can then adjust their muscle activity to quickly increase or decrease this setpoint. In addition, a plug-and-play neural network classifier detects up or down gestures at any time to offer finer control and facilitate more complex lifting scenarios where the person may want the robot to hold its end of the object at a significantly different height than their own.

†MIT Distributed Robotics Lab, Cambridge, MA 02139
{delpreto, rus}@csail.mit.edu

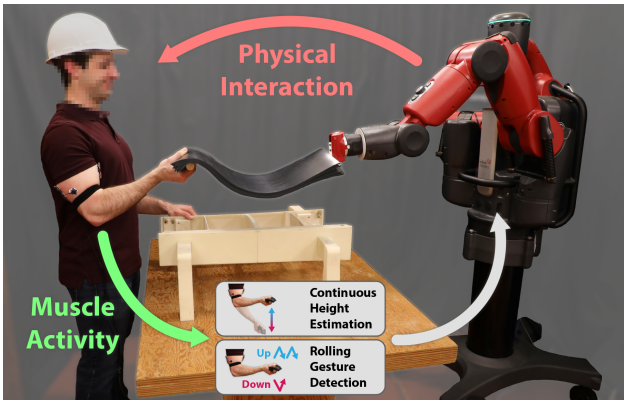


Fig. 1: A human and robot lift an object together, using muscle activity as the sole communication channel. Two pipelines process EMG signals to estimate continuous height adjustments and detect up/down gestures.[1]

This framework provides a human-robot communication channel that is embedded within the motions of the task itself. Muscle activity associated with the desired motions is used to grant the person control over the robot and guide it towards providing effective assistance according to their intentions and physical intuition. The system can then be applied to a variety of lifting tasks.

In particular, this work presents the following:

- an algorithm to continuously estimate a lifting setpoint from biceps activity, roughly matching a person's hand height while also providing a closed-loop control interface for quickly commanding coarse adjustments;
- a plug-and-play rolling classifier for detecting up or down gestures from biceps and triceps activity, allowing the user to explicitly command fine adjustments or move the robot to targets farther from their own hand height;
- an end-to-end system integrating these pipelines to collaboratively lift objects with a robot using only muscle activity associated with the task;
- experiments with 10 subjects to evaluate the setpoint and classification pipelines and to demonstrate collaborative assembly with rigid or flexible objects.

II. Related Work

This paper builds upon research exploring biosignals for robot control and frameworks for human-robot collaboration.

A. Human-Robot Interaction

There have been numerous approaches to understanding a person's intention and determining appropriate robot actions

[1] Videos are available at http://people.csail.mit.edu/delpreto/icra2019

based on models of task and team dynamics. Studies have explored understanding the user's perspective [1] and predicting human intention or plan adaptation [2], [3]. Effective team dynamics and cross-training have also been investigated [4], [5], [6], although the human-robot interface is often a bottleneck for implementation [7].

Physical interaction tasks have also been achieved using modalities such as vision, speech, force sensors, and gesture-tracking datagloves [8], [9], [10], [11], [12]. These may be difficult to generalize to complex tasks though, and may be hindered by interactions with the environment [13] or occlusions and ambient noise. Load-sharing policies have also been developed for jointly manipulating objects within simulation [14], and planning methods are explored for safety and efficiency [15], [16] or formation control [17].

B. Using Muscle Signals for Robot Control

While models to predict appropriate actions given a perceived world state are powerful when they can be constructed, directly conveying a user's physical intentions can be useful for interactive tasks. Many EMG devices have been developed for assistive robotics [18], including EMG-based exoskeletons for the hand [19], [20] or the upper-limb. Upper-limb exoskeletons have effectively used approaches such as parameterized muscle models [21], fuzzy logic controllers [22], or impedance controllers [23]. Models can also provide insights into muscle dynamics to facilitate the development of associated controllers [24], [25], [26], [27], [28]. Remote control or supervision of non-collaborating robots has also been explored via gesture-based control [29], [30], [31] or continuous trajectory estimation [32], [33]. In addition, muscle signals have been used to convey stiffness information for dynamic physical collaboration tasks [34].

Such studies have shown that EMG can yield effective human-robot interfaces, but also demonstrate associated challenges such as noise, variance between users, and complex muscle dynamics. Examples of addressing such challenges include redundant switching models [33] or leveraging the human within an exoskeleton control loop [35].

III. Experimental Setup and System Design

An experimental setup and a closed-loop system were designed to explore team lifting scenarios and evaluate performance. The principal components and information flow are illustrated in Figure 2.

A. Team Lifting Tasks

Two pillars of LEDs indicate target lifting heights or cue training gestures. During lifting tasks, the user lifts an object and controls the robot to achieve the desired height while the LEDs are lit, then lowers the object and robot back to the table when they turn off. A dumbbell weighing up to 10 lb is used for non-collaborative trials, while a shared object is used for trials with physical robot interaction.

In addition to these structured lifting tasks, collaborative assembly tasks are performed in which the user and robot install an object on a base structure. As seen in Figure 5, a variety of rigid and flexible objects are used. The user can command the robot through any trajectory they deem appropriate based on the task and the object's properties.

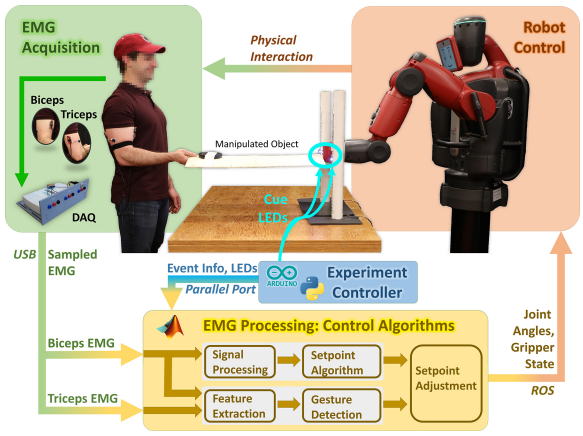


Fig. 2: The system consists of an experiment controller that coordinates the paradigm, EMG acquisition, two pipelines for computing a robot pose, the Baxter robot, and a human subject. Visual and/or physical feedback closes the loop.

B. Experiment Controller and Robot

A centralized control program coordinates the experimental paradigms. It controls event timing, decides target heights, and controls the cue LEDs. It also relays commands from the EMG subsystem to the robot and sends event triggers to the EMG subsystem. A humanoid Rethink Robotics Baxter robot was used for these experiments.

C. EMG Hardware and Data Acquisition

Muscle signals are acquired from the right Biceps Brachii short head and the right Triceps Brachii long head using differential pairs of adhesive 24 mm Covidien electrodes. MyoWare Muscle Sensors provide amplified raw and filtered signals, which are then sampled at 1 kHz by an NI USB-6216 data acquisition device and processed within Simulink 2017b. The setpoint algorithm uses the biceps signal, while the classification pipeline uses the biceps and triceps signals.

D. Subject Selection

A total of 10 subjects participated in the experiments (90% male, 80% right-handed). No previous experience using EMG interfaces was required, and subjects were not screened based on EMG signals. All subjects provided written consent for the study, which was approved by MIT's Committee on the Use of Humans as Experimental Subjects.

IV. Continuous Setpoint Estimation

The setpoint algorithm aims to estimate changes in the person's hand height while also creating a task-based control interface. Rather than model the muscle and limb dynamics to attempt an accurate prediction of pose and appropriate robot response, it leverages the person's ability to close the loop based on visual and physical feedback. The robot roughly lifts to the same height as the person using only their natural muscle activity, then the user can consciously control the robot by adjusting their muscle activity.

Algorithm 1 Setpoint Control Algorithm

SIGNAL PROCESSING
$raw \leftarrow$ amplified biceps EMG signal from MyoWare board, sampled at 1 kHz
$filtered \leftarrow$ band-pass filter 5-400 Hz
$envelope \leftarrow$ rectify, low-pass filter 5 Hz, amplify by 1.5
$envelopeScaled \leftarrow$ normalize using MVC-based $G_{normalization}$
UPDATE BUFFERS
Circular $baselineBuffer$ and $enabledBuffer \leftarrow envelopeScaled$
Non-circular $changesBuffer \leftarrow envelopeScaled$
DETERMINE ENABLE BIT
$enabled \leftarrow$ mean$(enabledBuffer) > L_{enable}$
if $enabled$ **then**
ESTIMATE AND FILTER SETPOINT ADJUSTMENTS
  if $changesBuffer$ is full **then**
    $rawChange \leftarrow$ integrate $[changesBuffer - \overline{baselineBuffer}]$
    if $|rawChange| < L_{change}$ **or** ($\geq 75\%$ of $actionHistory \neq 0$ **and** majority of those $\neq$ sign$(rawChange)$) **then**
      $rawChange \leftarrow 0$
    end if
    $setpointChange \leftarrow rawChange \times G_{setpoint}$
    $robotHeight \leftarrow robotHeight + setpointChange$
    Circular $actionHistory \leftarrow$sign$(rawChange)$
  end if
end if

Algorithm 1 outlines the pipeline, which processes a single EMG channel from the biceps to produce two outputs: a continuously adjusted robot height, and whether the person is engaged in the lifting task. Figure 3 shows sample results.

A. Signal Processing

The amplified raw biceps signal from the MyoWare processing board is conditioned and processed in software to extract a signal envelope. A bandpass filter preserves the useful frequency content of the EMG signal [36] while removing high-frequency noise and low-frequency offsets or motion artifacts. The signal envelope is then detected to indicate muscle activation levels. Example filtered and envelope-detected signals are shown in Figure 3.

B. Parameterized Setpoint Algorithm

The central principle of the setpoint algorithm is to focus on short-timescale changes in muscle activation around a long-timescale baseline level of activation. By incrementally updating the setpoint based on relative changes in muscle activity instead of mapping muscle activations to poses, the algorithm is more robust to EMG variation across users and time. In addition, changes in activation are intuitively informative for lifting tasks since moving a weight between two poses requires a spike in torque.

There are 8 main parameters, which will be introduced throughout this section then optimized in the next section.

The algorithm first applies a normalization gain $G_{normalization}$ based on maximum voluntary contraction (MVC) to the detected EMG envelope. This signal then populates a circular buffer of duration $D_{baseline}$ and a shorter noncircular buffer of duration $D_{integration}$. The mean of the long buffer represents a baseline level of activation, which is subtracted from the shorter buffer

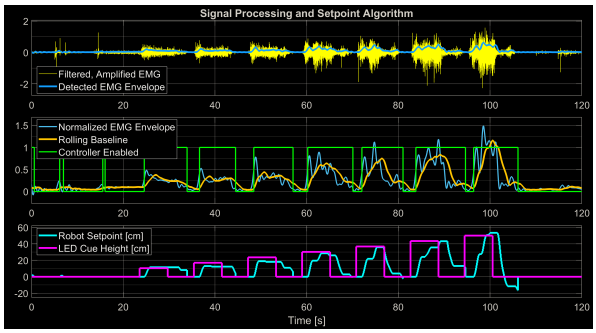


Fig. 3: These sample traces were recorded while the human-robot team lifted an object to 7 target heights. The pipeline filters, amplifies, and envelope-detects the EMG signal (top), computes a rolling baseline to determine whether the controller should be enabled (middle), and uses variations around the baseline to adjust the robot setpoint (bottom). The targets (magenta) were achieved.

TABLE I: Setpoint Algorithm Parameters

Parameter	Tuning Type	Optimization Bounds	Optimization Result	Values Used (mean ± SD)
$D_{baseline}$	One-time	[0.25, 5.00] s	2.38 s	2.40 s
$D_{integration}$	One-time	[0.05, 0.20] s	0.10 s	0.10 s
$D_{changes}$	One-time	[0.20, 2.00] s	1.90 s	1.90 s
D_{enable}	One-time	[0.38, 2.25] s	2.00 s	1.00 s
L_{change}	One-time	–	–	0.01
L_{enable}	Per-Subject	[0.01, 0.15]	0.15	0.28 ± 0.27
$G_{normalization}$	Per-Subject	–	–	11.50 ± 10.48
$G_{setpoint}$	Per-Subject	[0.20, 20.00]	0.75	0.38 ± 0.20

to yield changes around the baseline. These changes are integrated when the short buffer fills. If the result is above a threshold level L_{change}, it is amplified by a gain $G_{setpoint}$ to adjust the robot's setpoint.

Before applying this change to the robot, however, the computed adjustments are slightly filtered. A rolling history is stored spanning $D_{changes}$ seconds of whether the computed setpoint adjustments were positive, negative, or zero. A new setpoint adjustment is only applied if its sign agrees with the majority of the stored history (ignoring zeros) or if at least three-quarters of the history is zero. This filtering helps smooth the robot motion, which can be especially desirable when the user moves rapidly or when muscle signals become erratic due to fatigue.

To determine whether the controller should be enabled, a rolling buffer of the normalized envelope spanning D_{enable} seconds is compared to a threshold L_{enable}. If this indicates the user is not lifting, the robot waits in a neutral position.

C. Parameter Optimization

A summary of the parameters required by the model is shown in Table I. Most of the parameters are held constant for all users, and were optimized based on initial open-loop experiments where a user was cued to lift a dumbbell to each of 7 target heights. A genetic algorithm was used for all parameters except $G_{normalization}$ and L_{change}, which were set based on MVC and manual preference, respectively. Using the Matlab 2017b Global Optimization Toolbox, the algorithm was run for 66 generations with a uniform creation function, rank scaling, scattered crossover fraction 0.8, elite

ratio 0.05, and stochastic uniform selection. The objective function simulated the setpoint algorithm with new parameters, then extracted the root mean square (RMS) error of the computed setpoints at the end of each trial.

Table I presents the results along with the values ultimately used during online experiments. The computed durations are reasonable given the cadence of a typical lifting task; the baseline buffer spans a few seconds, the history of changes allows a reversal of lifting direction approximately once per second, and the integration buffer is short yet long enough to smooth spurious EMG fluctuations. Note that thresholds and gains act on normalized signals.

The three subject-specific parameters are based on brief calibration routines. The subject is cued to repeatedly tense and relax their arm, then to lift a dumbbell to 7 target heights without robot feedback. $G_{normalization}$ and L_{enable} are computed by comparing relaxation and contraction segments, then $G_{setpoint}$ is computed by simulating the setpoint algorithm and minimizing its error at each target.

D. Human-in-the-Loop Control

Although the model is optimized by minimizing position error between the user and robot, the system is not designed to track the human's position accurately in online trials; the user may not want the robot to mirror their own position. Instead, the algorithm provides a coarse height estimation and creates a framework with which the user can communicate desired motions. Biceps are naturally used to lift an object, then further tensing or relaxing the muscle will further raise or lower the robot. By using the task's primary muscle in a way that is consistent with the task, the system aims to reduce necessary training and behavioral modification.

Conceptually, the algorithm achieves this by leveraging the independence between joint position and stiffness. A person can lift an object using mainly their biceps, but then significantly change their muscle activation without moving the object by co-activating their antagonistic triceps. The controller effectively repurposes this stiffness degree of freedom as a communication channel to the robot.

V. Rolling Gesture Classification

While the setpoint algorithm estimates changes in the user's hand height and enables active control over the robot, it can be difficult to achieve fine-grained or persistent height adjustments. In addition, commanding robot positions that are significantly higher than the user's hand may be tiring.

To address these cases, a classification pipeline was implemented that operates in parallel with the setpoint algorithm. It continuously classifies EMG envelopes from the biceps and triceps to detect up and down gestures; the robot then briefly moves slowly in the desired direction. This pipeline is also plug-and-play, only trained on data from prior subjects, so new users can immediately control the robot via gestures.

A. Training Gestures and Feature Extraction

Training data was collected by cueing up or down gestures during specified time windows using the LED pillars. As illustrated in Figure 1, an up gesture consists of two brief upward hand motions while a down gesture consists of a single brief downward hand motion. As users become more experienced with the system, they can learn to produce the required muscle activations while minimizing object motion. Subjects were instructed to match their gesture duration to the LEDs, which remained lit for 1 s. Figure 4 visualizes collected EMG signals and extracted training data, demonstrating some common characteristics but also significant variability between gesture examples and across subjects.

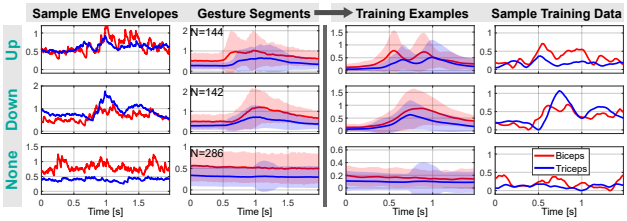


Fig. 4: EMG signals are segmented and processed into gesture training data by normalizing, centering, shifting down to 0, and downsampling. The center columns represent 286 trials from 6 subjects, with mean traces in bold and one standard deviation shaded on each side. Synthetic augmentation examples are not included.

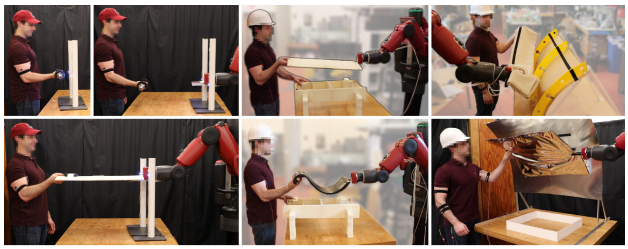
1) Segmentation and Signal Processing: Envelopes of biceps and triceps muscle activity are acquired from the MyoWare processing boards then normalized based on MVC. From each trial, one labeled gesture segment and one baseline segment are extracted according to the cue LEDs.

The segmented muscle signals are smoothed by a moving mean with duration 75 ms, and downsampled to 50 Hz. They are then independently shifted so their minimum values are at 0 to reduce the impact of signal drift, inter-subject variations, and the weight or height of the object.

2) Data Augmentation: To achieve reliable predictions on a rolling basis despite training on time-locked examples, a data augmentation approach can be employed [29]. Each extracted training segment is first centered; up gestures are centered to the average location of two prominent biceps peaks, while down gestures are centered to a single prominent triceps peak. Four positively labeled copies of each gesture are then synthesized by randomly shifting left and right between 0 and 100 ms. Four negatively labeled copies are also synthesized by shifting farther left and right between 150 and 450 ms. Together, these encourage the network to prefer gestures centered in its classification window within a certain tolerance. For each original baseline segment, two randomly shifted copies are also synthesized.

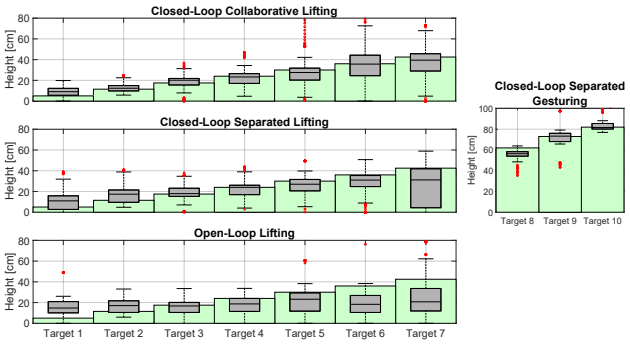
In addition to augmenting via time shifts, synthetic examples were generated based on magnitude. While the signals are normalized via MVC, there can still be variation across subjects, object weights, or gesture speeds. After synthesizing time-shifted examples, every example in the corpus is copied and scaled by a random factor between 0.6 and 1.4.

As a result of this augmentation, each gesture segment yields 10 positive and 8 negative examples while each baseline segment yields 6 negative examples. The center 1.5 s of each example is then extracted, and the two EMG channels are concatenated to yield a 150-element feature vector.



(a) (b) (c)

Fig. 5: Various team lifting and assembly tasks were performed. Column (a) uses the setpoint algorithm with cued target heights in open-loop, separated closed-loop, and interactive closed-loop scenarios. Column (b) performs assembly with rigid and flexible objects. Column (c) extends the system to use stiffness and additional degrees of freedom.



(a) (b)

Fig. 6: For each target lifting height (green), the achieved robot setpoints are aggregated across all users. All trials used the setpoint algorithm, but only those in (b) used gestures. There is significantly less error with closed-loop feedback, and using gestures further increases accuracy while reaching higher targets.

B. Neural Network Training and Online Classification

These labeled vectors are used to train a feed-forward neural network using the Pattern Recognition functionality of Matlab's Neural Network Toolbox (2017b). The network has a single hidden layer of size 50 using a hyperbolic tangent sigmoid activation function, and an output layer of size 3 using a softmax activation function. The three outputs are used to indicate whether the segment was classified as baseline, a down gesture, or an up gesture.

For each online experiment, a new classifier was trained using data from previous subjects. Streaming EMG envelopes are then normalized, smoothed, downsampled, and used to populate two 1.5 s rolling buffers in Simulink (2017b). With each new sample, the buffers are shifted down to 0, concatenated, and classified. Network outputs are slightly filtered to reduce spurious predictions; a rolling buffer of 40 classifications (800 ms) is maintained, and a final gesture prediction is declared if its mode occurs at least 4 times and is on average within 40 ms of the buffer's center.

If an up or down gesture is detected, the robot moves in that direction at 3 cm/s. It stops when the opposite gesture is detected, 5 s have elapsed without another gesture detected, or the robot's height limits are reached.

VI. Experimental Results and Discussion

Experiments were conducted with 10 subjects to evaluate the algorithms and system efficacy.[1]

A. Setpoint Algorithm

To evaluate the setpoint algorithm, users lifted an object to a sequence of 7 increasing heights while the system estimated an appropriate robot position. This was done in an open-loop scenario without robot motion, then in two closed-loop scenarios: one where the robot moved but did not interact with the user, and one where the human and robot jointly lifted a rigid object. Figure 5a shows the task setups. Targets were equally spaced between 10.0 cm and 50.0 cm above a 76.5 cm table. All 10 subjects participated in these scenarios, totalling 20 open-loop sequences (excluding 1 per subject used for calibration), 22 separated closed-loop sequences, and 39 collaborative sequences.

Figure 6a illustrates the results. Across all targets, the mean and standard deviation of the RMS setpoint error during the last 1.0 s of each trial for the open-loop, separated closed-loop, and collaborative closed-loop cases were 14.1 ± 15.2 cm, 10.0 ± 11.1 cm, and 9.0 ± 16.0 cm, respectively. Users visually compared the object or the robot's fingers to the LED targets, so some variability is expected. Each closed-loop case was significantly more accurate than the open-loop case ($p < 0.01$ for each one), with little difference between the two closed-loop cases ($p < 0.49$). This suggests that including closed-loop feedback, either visual or physical, allows the human to significantly improve performance by adjusting muscle activity to control the robot.

The object was also level during the final 1.0 s of collaborative trials, with a pitch of -1.4° $\pm$ 5.4°. Together with the increased accuracy between open-loop and closed-loop cases, this indicates that users successfully leveraged arm stiffness to control the robot without affecting their own pose.

1) Assembly Tasks: Two assembly tasks illustrated in Figure 5b were also performed. The human and robot jointly lifted an object, waited while the human manipulated a base structure with their free hand, then lowered the object onto the structure. This was done with a rigid object (8 subjects, 41 trials) and a flexible rubber sheet (4 subjects, 20 trials). The rigid-object assembly lasted 16.4 ± 8.1 s, during which the object pitch averaged -0.3° $\pm$ 5.3°. The flexible-object assembly averaged 8.8 ± 3.8 s. Assembly was successful in both cases, demonstrating applicability of the system to basic team lifting tasks with a variety of object material properties.

B. Gesture Classification

To evaluate whether the gesture classification pipeline facilitates finer control and higher targets, users were cued to gesture the robot towards 3 targets from 62.5 cm to 82.5 cm above the table while keeping their elbow at approximately 90°. This was designed to mimic lifting a flexible carbon fiber sheet to a specific angle before pressing it against a vertical mold. Figure 6b visualizes the results. Across all 3 targets, spanning 38 trials from 5 subjects, the mean RMS error during the final 1.0 s of each trial was 6.8 ± 7.6 cm. Users achieved lower error than in the separated closed-loop case without gestures ($p < 0.05$). And while the setpoint

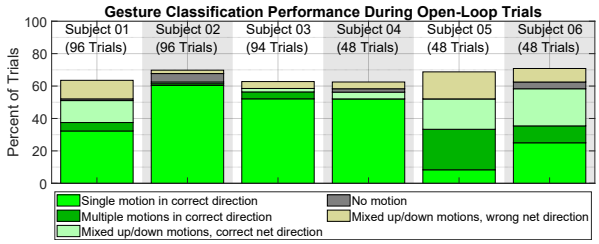


Fig. 7: An EMG classifier trained on previous subjects operated continuously during open-loop session of cued up or down gestures. The results indicate the breakdown of trials during which the robot would have moved in the direction indicated by the users' gestures.

algorithm's error generally increased with increasing height, the gesture control framework enabled a larger workspace.

1) Classification Performance: The classification pipeline operated during the open-loop gesture training sessions as well as the closed-loop gesture task. Each experiment used a classifier trained on the open-loop sessions of previous subjects. 6 subjects participated in these sessions, 5 of whom also participated in the closed-loop sessions. The first classifier was trained on 160 gestures from a separate subject.

Figure 7 summarizes the performance during open-loop trials by determining whether the robot would have moved in the gestured direction. Although the networks were trained on relatively small collections of time-locked examples, they were able to classify gestures on a rolling basis for new users that had neither feedback from the system nor direct knowledge about how prior subjects performed the gestures.

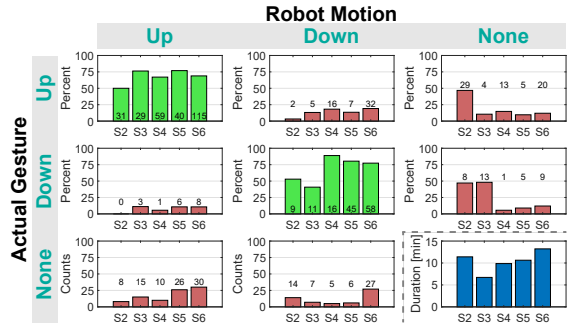
The same networks were used for the closed-loop control task described above, when users could make either gesture at any time. Ground truth for the gestures was obtained by post-processing videos to estimate robot motion in response to user gestures. Figure 8 depicts per-experiment and overall confusion matrices. Across all subjects, the classifier operated at 50 Hz for over 51 min and the robot responded appropriately to 68.8% of gestures. Note that if the robot ignored a gesture, the person could simply repeat it. Comparing open-loop and closed-loop results also suggests that users can subtly adjust their gestures based on feedback. Together with the accuracy shown in Figure 6b, these results indicate a successful plug-and-play gesture control interface.

C. Extended Assembly Tasks

Two assembly tasks with increased complexity were also performed to demonstrate extensibility of the system. Each scenario was completed at least 10 times by a single subject.

In the top task of Figure 5c, the human and robot lift a flexible canvas and press it against a base structure; this mimics pressing carbon fiber resin against a mock fuselage. The system was augmented with thresholded EMG activity from the forearm to detect when the user pushes the cloth forward. The robot then moves horizontally and holds its position while the user completes the dexterous task of shaping the cloth against the structure.

In the second task of Figure 5c, the human and robot install a flexible sheet onto an overhead structure; this mimics installing a fire blanket on a fuselage. The system is



(a)

Robot Motion

Actual Gesture	Up Count	Up Percent	Down Count	Down Percent	None Count	None Percent	Total Count
Up	274	67.32%	62	15.23%	71	17.44%	407
Down	18	9.33%	139	72.02%	36	18.65%	193
None	89		59		Total Experimental Duration:		
Total	381		260		51.8 min		

(b)

Fig. 8: Online gesture classification performance during closed-loop tasks is summarized by confusion matrices for each experiment (a) and across all subjects (b). Percents are based on the counts of actual gestures, and numbers on each bar count associated instances.

augmented to detect the user's arm stiffness by multiplying the biceps and triceps envelopes. If it is below a threshold, the robot remains compliant around its setpoint so the user can position the sheet. Otherwise, the robot stiffens so the user can pull the sheet taught and install it.

D. Workload and Usability

All 10 subjects also completed a survey including the NASA-TLX workload assessment [37]. The raw TLX score averaged 44.7 ± 16.16 out of 100, with lower numbers indicating reduced workload. When asked to also rate system learnability and intuitiveness, responses averaged 38.6 ± 27.3 and 37.7 ± 26.9 out of 100 with lower scores indicating faster learning and increased intuitiveness. While a larger sample size would be needed to generalize these results, they suggest that the system required a reasonable amount of workload and can be intuitive to use after brief learning sessions.

VII. Conclusion

Working effectively with a robot on physical tasks requires a natural and unobtrusive communication channel to convey desired motions. The presented system moves towards this goal by providing a real-time interface for team lifting tasks that can use only two channels of upper-arm muscle activity to continuously estimate changes in a person's hand height, create a control interface, and detect up or down gestures. The gesture classification pipeline is also plug-and-play, not requiring retraining for each new user. The system allows the human's understanding of the task and environment to guide behavior, so the robot can provide effective assistance in a variety of team lifting scenarios. With such steps, the vision of humans working seamlessly with robots to increase safety and productivity moves ever closer towards reality.

Acknowledgments

This work was funded in part by the Boeing Company, for which the authors express gratitude.

REFERENCES

[1] J. G. Trafton, N. L. Cassimatis, M. D. Bugajska, D. P. Brock, F. E. Mintz, and A. C. Schultz, "Enabling effective human-robot interaction using perspective-taking in robots," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, vol. 35, no. 4, pp. 460–470, 2005.

[2] S. J. Levine and B. C. Williams, "Concurrent Plan Recognition and Execution for Human-Robot Teams," *Twenty-Fourth International Conference on Automated Planning and Scheduling*, no. 2001, pp. 490–498, 2014.

[3] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6175–6182.

[4] E. E. Entin and D. Serfaty, "Adaptive Team Coordination," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 41, no. 2, pp. 312–325, 1999.

[5] C. E. Volpe, J. A. Cannon-Bowers, E. Salas, and P. E. Spector, "The Impact of Cross-Training on Team Functioning: An Empirical Investigation," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 38, no. 1, pp. 87–100, 1996.

[6] S. Nikolaidis and J. Shah, "Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 33–40.

[7] R. Roberson, "Human–Robot Interaction in Rescue Robotics," *Ieee Transactions on Systems, Man, and Cybernetics*, vol. 32, no. 2, pp. 138–153, 2004.

[8] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: Object-lifting case study with a humanoid," *IEEE-RAS International Conference on Humanoid Robots*, pp. 399–404, 2009.

[9] D. J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar, "Collaborative human-humanoid carrying using vision and haptic sensing," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 607–612, 2014.

[10] J. R. Medina, M. Shelley, D. Lee, W. Takano, and S. Hirche, "Towards interactive physical robotic assistance: Parameterizing motion primitives through natural language," in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 1097–1102.

[11] C. Lenz, S. Nair, M. Rickert, A. Knoll, W. Rösel, J. Gast, A. Bannat, and F. Wallhoff, "Joint-action for humans and industrial robots for assembly tasks," *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*, pp. 130–135, 2008.

[12] A. Edsinger and C. C. Kemp, "Human-robot interaction for cooperative manipulation: Handing objects to one another," *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*, pp. 1167–1172, 2007.

[13] L. Peternel, T. Petrič, E. Oztop, and J. Babič, "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach," *Autonomous robots*, vol. 36, no. 1-2, pp. 123–136, 2014.

[14] M. Lawitzky, A. Mörtl, and S. Hirche, "Load sharing in human-robot cooperative manipulation," *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*, pp. 185–191, 2010.

[15] J. A. S. Przemyslaw A Lasota, Gregory F Rossano, "Toward safe close-proximity human-robot interaction with standard industrial robots," *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pp. 339–344, 2014.

[16] R. Wilcox, S. Nikolaidis, and J. Shah, "Optimization of Temporal Dynamics for Adaptive Human-Robot Interaction in Assembly Manufacturing," *Robotics Science and Systems*, vol. VIII, pp. 441–448, 2012.

[17] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[18] R. a. R. C. Gopura, D. S. V. Bandara, J. M. P. Gunasekara, and T. S. S. Jayawardane, "Recent trends in EMG-Based control methods for assistive robots," *Electrodiagnosis in New Frontiers of Clinical Research*, pp. 237–268, 2013.

[19] M. Dicicco, L. Lucas, and Y. Matsuoka, "Comparison of Two Control Strategies for a Muscle Controlled Orthotic Exoskeleton for the Hand," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. pp, no. April, pp. 1622–1627, 2004.

[20] M. Mulas, M. Folgheraiter, and G. Gini, "An EMG-Controlled Exoskeleton for Hand Rehabilitation," *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pp. 371–374, 2005.

[21] J. L. a. S. Ramos and M. a. Meggiolaro, "Use of surface electromyography for human amplification using an exoskeleton driven by artificial pneumatic muscles," *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 585–590, 2014.

[22] K. Kiguchi, T. Tanaka, and T. Fukuda, "Neuro-fuzzy control of a robotic exoskeleton with EMG signals," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 481–490, 2004.

[23] R. A. R. C. Gopura, K. Kiguchi, and Y. Yi, "SUEFUL-7: A 7DOF upper-limb exoskeleton robot with muscle-model-oriented EMG-based control," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, no. NOVEMBER 2009, pp. 1126–1131, 2009.

[24] F. E. Zajac, "Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control," *Critical reviews in Biomedical Engineering*, vol. 17, no. 4, pp. 359–411, 1989.

[25] K. Manal and T. S. Buchanan, "A one-parameter neural activation to muscle activation model: estimating isometric joint moments from electromyograms," *Journal of biomechanics*, vol. 36, no. 8, pp. 1197–1202, 2003.

[26] E. Cavallaro, J. Rosen, J. C. Perry, S. Burns, and B. Hannaford, "Hill-based model as a myoprocessor for a neural controlled powered exoskeleton arm - parameters optimization," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 4514–4519.

[27] A. Qashqai, H. Ehsani, and M. Rostami, "A hill-based emg-driven model to estimate elbow torque during flexion and extention," in *2015 22nd Iranian Conference on Biomedical Engineering (ICBME)*, Nov 2015, pp. 166–171.

[28] S. Menon, T. Migimatsu, and O. Khatib, "A parameterized family of anatomically accurate human upper-body musculoskeletal models for dynamic simulation & control," in *Proceedings of the 16th IEEE-RAS International Conference on Humanoid Robots*, Novebmer 2016.

[29] J. DelPreto, A. F. Salazar-Gomez, S. Gil, R. M. Hasani, F. H. Guenther, and D. Rus, "Plug-and-play supervisory control using muscle and brain signals for real-time gesture and error detection," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[30] J. Kim, S. Mastnik, and E. André, "EMG-based hand gesture recognition for realtime biosignal interfacing," *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI)*, vol. 39, p. 30, 2008.

[31] B. Crawford, K. Miller, P. Shenoy, and R. Rao, "Real-Time Classification of Electromyographic Signals for Robotic Control," *Proceedings of AAAI*, pp. 523–528, 2005.

[32] P. K. Artemiadis and K. J. Kyriakopoulos, "EMG-based control of a robot arm using low-dimensional embeddings," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 393–398, April 2010.

[33] N. M. López, F. di Sciascio, C. M. Soria, and M. E. Valentinuzzi, "Robust EMG sensing system based on data fusion for myoelectric control of a robotic arm." *Biomedical Engineering Online*, vol. 8, p. 5, 2009.

[34] L. Peternel, N. Tsagarakis, and A. Ajoudani, "A human–robot co-manipulation approach based on human sensorimotor information," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 7, pp. 811–822, 2017.

[35] T. Lenzi, S. M. M. De Rossi, N. Vitiello, and M. C. Carrozza, "Proportional EMG control for upper-limb powered exoskeletons," *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pp. 628–631, 2011.

[36] C. J. D. Luca, "Surface electromyography: Detection and recording," *DelSys Incorporated*, 2002.

[37] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in Psychology*, 1988, vol. 52, pp. 139–183.