CO-CONTRACTILE DIFFERENCES DURING ADAPTATION TO ABRUPT AND GRADUAL DYNAMIC PERTURBATIONS

By

David W. Ptashnik

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Kinesiology - Master of Science

2019

ABSTRACT

CO-CONTRACTILE DIFFERENCES DURING ADAPTATION TO ABRUPT AND GRADUAL DYNAMIC PERTURBATIONS

By

David W. Ptashnik

Adaptation to a novel motor task has been shown to be facilitated through co-contraction as well as exposure to a gradual, rather than an abrupt, dynamic perturbation. In this study, participants performed center out reaching movements in a velocity dependent force field. One group was exposed to a perturbing force of 20 $N{\cdot}m^{-1}{\cdot}s$, another group was exposed to 5 $N{\cdot}m^{-1}{\cdot}s$ -step increments of perturbing force up to a total force of 20 $N{\cdot}m^{-1}{\cdot}s$. Adaptation measured by randomly interspersed channel trials was found to be better in the group exposed to the gradually increasing perturbation. However, no significant differences were observed in levels of muscle activity during the exposure period. These results suggest that despite both co-contraction, and exposure to a gradual perturbation facilitating dynamic motor adaptation, these two mechanisms are independent of one another

ACKNOWLEDGMENTS

I would like to thank the members of the Michigan State University Motor Neuroscience Lab for their support throughout this project. I would like to particularly thank Dr. Florian Kagerer, Alex Brunfeldt, and Phil Desrochers for their guidance in preparing this manuscript and the corresponding presentations. I would also like to thank Dr. David Ferguson and Dr. Rajiv Ranganathan for providing comments and edits to both my manuscripts and presentations throughout the process. I would like to thank my parents, Brian and Martha Ptashnik for their unwavering support throughout my entire experience in this graduate program. Finally, I would like to thank Claire Dinsmore for remaining by my side and keeping me in a positive state of mind even when difficulties arose.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The ability for the motor system to adapt to intrinsic and extrinsic changes, or external environment, is a true marvel of the nervous system. One of the main theoretical approaches trying to understand the mechanisms underlying motor adaptation uses the concept of an internal model, or internal representation. Such a representation is postulated to mimic the input and output variabels of the human motor system. These representations are formed in order to compensate for forces and instability which may arise from our interactions with the environment (Kawato 1999; Burdet et al. 2006). This approach also provides an in-depth illustration of how the motor system can learn new skills or adapt to different destabilizing dynamics that are applied to previously learned skills (Heald et al. 2018; Milner 2002).

One way to probe the function of a system is to perturb it and observe how it recovers from the perturbation. In this context, a perturbation is a disruption which causes the system to deviate from its normal state. Studies have been done examining different forms of perturbations to learned movements and how motor system adaptation occurred in regard to these different destabilizing dynamics (Kagerer et al. 1997; Klassen et al. 2005; Milner 2002). Two such studies compared adaptation to both a gradual (incremental) perturbation over the course of the learning period and a sudden, abrupt perturbation which provided a larger force, constantly applied over the course of the learning period. These studies were done examining both adaptation to an altered visual perception of the direction of movement (Kagerer et al. 1997) and a dynamic, velocity-dependent force-field (Klassen et al. 2005). Both studies showed evidence that a more complete adaptation happens with the gradual perturbation as compared to the abrupt perturbation which was seen through stronger after effects in the groups exposed to the gradual perturbation during the post-adaptation phases of the studies. This provides support to the notion that the sensorimotor representation of a movement is updated during the process of destabilization, and the newly introduced task constraint is accounted for in the new representation of that specific movement.

Other experiments have been done examining the role of muscular co-contraction on various aspects of motor control, such as the acquisition on an internal model, as well as aiding in the error reduction components of the internal model by increasing the mechanical impedance of the limbs, leading to enhanced movement accuracy and stability (Gribble et al. 2003; Rosa 2015; Heald et al. 2018). Co-contraction is simply defined as the simultaneous activation of opposing muscles around a joint. As the motor system encounters task and environmental constraints, it aims to stabilize and accommodate its behavior to produce the most successful movements possible given the present environmental conditions (Hogan 1985). Co-contraction can indeed enhance the stiffness of a joint, which would lead to an increase in a limb's impedance to any form of destabilizing dynamics (Osu and Gomi 1999; Burdet el a. 2001). The goal of this study is to examine the differences in con-contraction during the process of adaptation to a gradual and abrupt dynamic perturbation.

In accordance with the findings from several studies, examining the role of co-contraction in the motor system, the proposed experiment will shed light into how co-contraction aides in the adaptation to novel dynamics. It has been observed that co-contraction can accelerate the rate of dynamic motor learning (Heald et al. 2018) and that gradually increasing a perturbation allows for a more complete updating of the internal model (Kagerer et al. 1997; Klassen et al. 2005). However, it is not currently known whether these findings are connected or independent of one another, and that is what this study aimed to examine.

Review of the Literature

The Concept of the Internal Model

As previously mentioned, the concept of the internal model describes a theoretical framework which acts as a sensorimotor representation of the inputs and outputs of the motor system. This provides a simplified way for us to explain the complexity behind purposeful movement production. This concept also proposes that there is a component of the internal model called the forward model which predicts the next state of the motor system given the current state of the system and the current motor command (Wolpert et al 1995; Maill and Wolpert 1996; Kawato 1999; Ostry and Feldman 2003). The predicted sensory consequences from the forward model are compared to the actual sensory consequences from the internal model, which allows for online error correction to take place.

The motor system can perform this online error correction by increasing the mechanical impedance (resistance to imposed motion) of the specific limb involved. This can be achieved by co-contracting the muscles around the joint in response to the requirements of the task (Burdet et al. 2001; Mitrovic et al. 2010; Franklin et al. 2003). As a consequence of this increase in co-contraction, both the accuracy and stability of movements may be enhanced even when destabilizing dynamics are present.

Co-Contraction as a Mechanism of Error Reduction- Movement Accuracy

Changes in co-contraction effect limb impedance in response to the requirements of a task (Hogan 1984; Miler 2002; Osu et al. 2002) and increased precision demands may coincide with increased co-contraction (Viser et al. 2004). Paul Gribble and colleagues demonstrated how co-contraction can be used to increase endpoint accuracy in multi-joint movements (Gribble et al. 2003). In their study, participants reached towards four targets of different sizes and locations from a central starting point. Targets were either 5, 30, or 45 mm in radius with a boundary of 20, 35, and 50 mm respectively for each target. Their findings showed that there were increased levels of co-contraction when the participants reached to the smaller of the targets (Osu et al. 2004). This confirmed their hypothesis that the central nervous system does modify the specific muscle activations in the face of accuracy constraints placed on the task and is in line with previous findings which showed that co-contraction and limb stiffness are scaled globally as a function of the target size (Milner 2002; Selen et al. 2005; Lametti et al. 2007). Osu et al. also demonstrated that as a target size decreased, so did the endpoint deviation of the reach due to the increased levels of co-contraction present during the higher accuracy demands (Osu et al. 2004).

Co-Contraction as a Mechanism of Error Reduction- Movement Stability

Aside from being a way by which movement endpoint accuracy can be controlled, muscular co-contraction can also be used as a mechanism to increase the stability of a movement when novel or purposefully destabilizing dynamics are introduced (Milner 2002; Thoroughman and Shadmehr 1999; Imamizu et al. 2000). The achievement of stable interaction requires that the mechanical behavior of the human arm become adapted to the current task and dynamics involved (Milner and Cloutier 1993). Both Thoroughman and Shadmehr (1999) and Milner (2002) examined changes in co-contraction during the introduction of perturbations to the movement. In one study, participants performed reaching movements using a robotic arm manipulanda and were perturbed during the movement by a velocity-dependent force field. A second study examined this phenomenon using a torque motor to perturb wrist flexion and extension movements via a position dependent instability. Electromyography (EMG), which was the variable of interest, and the results show that EMG could be used as a measure of neural output that provides insight into how forward internal models are formed (Thoroughman and Shadmehr 1999).

That study showed that initially during a movement there was an increase in the EMG activity of the bicep mid-way through the movement, corresponding to a correction of a hand path back to the desired trajectory, when the destabilizing perturbation was introduced. The motor system uses the forward internal model to generate a predictive representation of what the next state of the system will be (Miall and Wolpert 1996; Kawato 1999), but when these perturbations (the velocity-dependent force field in this situation) are introduced, the motor system is unable to accurately predict its next state. This resulted in an increase in muscular activity earlier in the movement, as seen through elevated EMG levels and a hand path which more closely resembled that of the desired trajectory for the movement. This increase in muscle activity aided in maintaining movement accuracy and stability by increasing the initial impedance of the arm in the presence of the force field (van Galen and Schomaker 1992; Gribble and Ostry 2000; Gribble et al. 2003; Franklin et al. 2008).

Milner's study also investigated the role of muscular co-contraction in the production of movement stability. The goal of this study was to see how patterns of agonist-antagonist muscular co-contractions adapt in the face of destabilizing task dynamics when the speed of the movement was taken into account (Milner 2002). Participants were instructed to move a cursor across a screen, from right-to-left, using wrist flexion and extension movements. Again, they were perturbed by a position dependent instability during the movement and the resulting EMG activity during the movement was recorded. Participants were cued to perform these movements either fast or slow. Movements performed both slowly and quickly showed increased levels of muscle activity, but those performed slowly showed a sustained high level of muscle activation over the entire movement (Bennett 1993, Gribble et al. 2003). This elevation in co-contraction continued during the post-stabilization period of the movement and was able to provide the motor system with more information regarding its current state than the movement performed quickly. Thus, the slow movement was able to better resist the perturbations and be subject to less variability than the fast movements (Fitts 1954, Harris and Wolpert 1998). The author of that study suggested that this means that this co-contraction is necessary to achieve stability during and at the endpoint of the movement (Gribble et al. 2003; Osu et al. 2002; Wolpert et al. 1995; Kawato 1999; Thoroughman and Shadmehr 1999; Milner 2002).

These are just a few examples of how the motor system can enhance the mechanical impedance of the limbs, by increasing co-contraction when confronted with mechanical instability (Milner and Cloutier 1993). Thus, a stable movement can be inferred to be the result of an accurate forward sensorimotor representation as the motor system is able to correctly predict its next state, given that the system is provided with accurate information regarding its current state (Wolpert et al. 1995; Kawato 1999; Thoroughman and Shadmehr 1999; Milner 2002).

Co-Contraction and the Internal Model

It has been examined that, using muscular co-contraction as a mechanism to increase limb impedance, the motor system is able to increase the accuracy of movements as well as increase the stability of movements, when disruptive dynamics are introduced. This increase in limb impedance by co-contraction can also be used as a mechanism by which the nervous system facilitates the learning of a new motor task (Rosa 2015). The hypothesis behind the form of learning, generated by a forward internal model states, that humans learn a new task as a result of repeated exposure to sensory signals from the limbs as they move through the environment. These signals are then integrated with the motor commands to produce purposeful movement (Bizzi and Mussa-Ivaldi 1998; Rosa 2015). As these movements occur, the system may encounter error signals that imply a discrepancy between the original motor command and the movement that was produced. The presence of these error signals allows for the system to compensate for the new demands placed on it, facilitating adaptation (Kagerer et al. 2006).

When a new task or skill is being learned, it is not uncommon for an individual to appear rigid during the initial learning process, as resistance to the disturbing effects of the novel dynamics are increased, via co-contraction (Bennet et al. 1991; Heald et al. 2018; Milner and Franklin 2005). As the individual continues practicing the task, the stiffness of the limb decreases (as defined by limb impedance) causing the levels of co-contraction to decrease and the specific internal model that encodes for the newly learned movement to become more stable (Milner and Cloutier 1993). One study that examined this was done by James Heald, David Franklin, and Daniel Wolpert. They posed the question: can increasing the level of co-contraction increase the rate at which an internal model is acquired. In this study, participants made forward reaching movements while a velocity-dependent force field was applied to their limb. Participants were assigned to three groups: one which was cued to relax their muscles in the presence of the perturbation, one which was cued to stiffen their arm (using co-contraction) during the perturbation, and one which was not cued either way (Heald et al. 2018).

Increased levels of EMG were seen in all groups when the force field was applied during the reach. The group which was cued to stiffen its muscles prior to the reach showed a greater increase in global EMG than the relaxed group and the control group, as well as greater adaptation to the task during the earlier period of the movement. Thus, the internal model that takes into account the force field perturbation being present during the movement was updated more quickly when participants were told to co-contract (stiff group) than in the group which remained relaxed throughout the movement and perturbation, and the group which was not cued (Bennet et al. 1991; Bizzi and Musa-Ivaldi 1998). These findings were in line with results that previous studies have reported under similar circumstances, in that, co-contraction increases when an error signal is produced from a disruption to the movement (Meulenbroek et al. 2005; Kagerer et al. 2006), increased levels of co-contraction provide greater amounts of sensory feedback to the system which allows for a more accurate sensorimotor representation of the movement and a more accurate movement itself (Gribble et al. 2003; Desmurget and Grafton 2000), and that as the feedback response to the error signals improve during the period of adaptation, the levels of muscle activity will respond accordingly as the sensorimotor representation of the movement becomes more fine-tuned (Thoroughman and Shadmehr 1999; Osu et al. 2002; Milner and Franklin 2005).

Hypothesis

Previous studies have shown that motor adaptation can be facilitated by 1) co-contraction, and 2) exposure to gradual (vs. abrupt) perturbations. Based on this, we hypothesize that if there is a direct relationship between these two mechanisms, co-contractile response should be more pronounced in a gradual than an abrupt dynamic perturbation. Alternatively, if co-contraction is not higher despite better adaptation it would indicate independence between these mechanisms.

MATERIALS AND METHODS

Participants

Sixteen, healthy, college students with normal or corrected-to-normal vision (3 female; 13 right-handed and 3 left-handed; age 20.25 ± 2.13 yr., mean ± SD) were recruited from the student population at Michigan State University and were randomly assigned to either a group which was exposed to a gradual or an abrupt dynamic perturbation. All participants reviewed and signed an informed consent statement and filled out a copy of the Edinburgh Handedness Questionnaire in order to verify hand dominance (Oldfield 1971). All experimental protocols were approved by the Michigan State University Institutional Review Board.

Apparatus

For data collection, a free moving, robotic manipulanda was used (KINARM by BKIN Technologies, Kingston, Ontario, Canada) (Fig 1). A monitor projected onto a reflective surface to provide visual feedback to the participants; vision of the workspace below was blocked by the reflective surface. During use of the device, the locations of the participant's hands were represented by white dots; data were acquired at a sampling rate of 1,000 Hz.

Figure 1: Picture of KINARM apparatus by BKIN Technologies

Experimental Design - Kinematics

Participants made horizontal reaching movements using their dominant arm to one of three targets, located at 10 cm from the center point and oriented at 45, 90, and 135 degrees from the center starting position (Fig 2) and were instructed to move and straight and accurately towards the target as possible. If the time it took to complete the movement was between 800 ms and 1250 ms, the target would turn green, providing feedback to the participant that the time to perform the movement was in the accepted range. If the time it took to complete the movement was less than 800 ms or greater than 1250 ms, the target would change a different color in order to provide feedback to the participant. The target turning blue denoted that the movement time was too long and the target turning red denoted that the movement time was too short. If participants did not perform the reach within the specified time interval, they were verbally instructed on how to adjust the speed of their reach for subsequent movements. Both groups performed 28 unperturbed reaches to the three targets in a null field (no perturbing force present) in order to establish a baseline of muscle activity during the task. During this phase the only force present was a viscous field spread across the entire workspace designed to dampen the movements. This viscous force was present during each of the three phases and is given as:

$$\begin{bmatrix} Fx \\ Fy \end{bmatrix} = \begin{bmatrix} -b & 0 \\ 0 & -b \end{bmatrix}$$

Where *Fx* and *Fy* are the forces present on the workspace in the x and the y directions, respectively, and b is equal to 7 $N \cdot m^{-1} \cdot s$. Once the exposure period began, the abrupt group experienced a perturbing force of 20 $N \cdot m^{-1} \cdot s$ for the subsequent 208 trials give as:

$$\begin{bmatrix} Fx \\ Fy \end{bmatrix} = \begin{bmatrix} -b & -20 \\ 20 & -b \end{bmatrix} \begin{bmatrix} Vx \\ Vy \end{bmatrix}$$

Where *Fx*, *Fy, Vx,* and *Vy* are the forces and velocities at the handles of the device in the x and the y directions. The gradual group experienced perturbations of 5, 10, 15, and 20 $N{\cdot}m^{-1}{\cdot}s$ in increments of 52 trials before increasing to the next level given as, for each of the levels of perturbation:

$$\begin{bmatrix} Fx \\ Fy \end{bmatrix} = \begin{bmatrix} -b & -5 \\ 5 & -b \end{bmatrix} \begin{bmatrix} Vx \\ Vy \end{bmatrix}$$

$$\begin{bmatrix} Fx \\ Fy \end{bmatrix} = \begin{bmatrix} -b & -10 \\ 10 & -b \end{bmatrix} \begin{bmatrix} Vx \\ Vy \end{bmatrix}$$

$$\begin{bmatrix} Fx \\ Fy \end{bmatrix} = \begin{bmatrix} -b & -15 \\ 15 & -b \end{bmatrix} \begin{bmatrix} Vx \\ Vy \end{bmatrix}$$

$$\begin{bmatrix} Fx \\ Fy \end{bmatrix} = \begin{bmatrix} -b & -20 \\ 20 & -b \end{bmatrix} \begin{bmatrix} Vx \\ Vy \end{bmatrix}$$

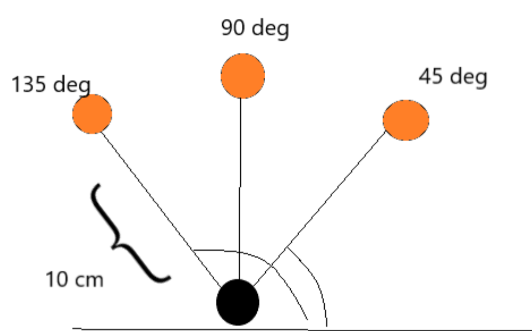
The target positions were displayed at random. During the entire experiment, channel trials were randomly inserted to aid in the assessment of the participant's adaptation to the task. These channel trials forced the participant to reach in a perfectly straight line towards the target oriented at 90 degrees through stiffness (2000 Nm) and dampening (5 $N{\cdot}m^{-1}{\cdot}s$) perpendicular to the movement direction, and no perturbing forces will be acting on them during these trials. Following the exposure period, 28 more unperturbed trials were administered in order to assess for any after effects that may have persisted.

Experimental Design - Electromyography

Surface electromyography (EMG) was recorded from two muscles of the shoulder (posterior deltoid and anterior deltoid), two muscles of the shoulder girdle (trapezius and pectoralis major), and two muscles involved in the shoulder-elbow complex (biceps brachii and triceps). The EMG signal were recorded using a BrainVision ExG 16 channel BrainAmp system (Brain Products, Gilching, Bavaria, Germany). Participants' skin was cleaned with an alcohol wipe and prepared with a light brushing of sand paper in order to increase signal conductance. Electrodes were then placed on the chosen muscles using a belly-tendon montage, maximizing signal strength while minimizing cross talk from nearby muscles. Electrodes were additionally secured in place with medical tape. EMG signals were filtered [cutoff frequency: 20 Hz (low), 500 Hz (high)] and sampled at 1,000 Hz. Prior to the experimental task, participants performed a maximum voluntary isometric contraction (MVC) in order to establish a maximal level of muscle activity for each of the muscles involved in the task for each participant. This was done for individual normalizing of the muscles for each participant since the maximal muscle activity observed will differ between participants.

Figure 2: **A.** The experimental paradigm of the study showing target locations, distance from the start position and direction. **B.** The channel layout used on the EMG system.

A.



B.

Channel	Muscle
1	Biceps
2	Triceps
3	Posterior Deltoid
4	Anterior Deltoid
5	Trapezius
6	Pectoralis Major

Data Analysis

The collected EMG data was sampled at 1,000 Hz for biceps brachii, triceps, trapezius, pectoralis major, anterior deltoid, and posterior deltoid. Collection began 500 ms prior to movement onset and ended 1000 ms after movement onset. The sampled signal was then high pass filtered at 40 Hz, full wave rectified, and low pass filtered at 6 Hz using a 3rd order Butterworth filter. Raw muscle activity for each participant was examined for movement artifacts, which were subsequently noted and removed.

These signals were then segmented blocks of 0.5 seconds. To the quantify the outcomes from our EMG data, we took the integral of the rectified signal, which provided us with a measure of the energy of that signal during each trial.

The kinematic data were analyzed using several different measures: 1) maximum lateral deviation (MLD) during the reach in order to view the farthest lateral point of deviation from the desired, straight line, reach trajectory, 2) root mean square error (RMSE) which provided us with a measure of the linearity of the movement (the area of the space between the desired reach trajectory and the actual reach trajectory) , and 3) during the channel trials, the force in which the participants pressed against the channel wall (peak lateral force) was used as a measure of adaptation to the task. All data analysis was done using BrainVision Analyzer 2 (Brain Products, Gilching, Bavaria, Germany) for the EMG data and MATLAB r2019a (MathWorks, Incorporated, Natick, Massachusetts) for the kinematic data and statistical analysis was done in RStudio.

To identify differences in EMG signal energy, MLD, movement RMSE, and adaptation between the groups, repeated measures ANOVAs were performed. The significance for all statistical tests was set to $P < 0.05$ and all kinematic data were reported as mean ± standard error of the mean (s.e.m). For analysis, kinematic data was calculated for each trial and was blocked into groups of 4 trials to include a reach to each direction and one channel trial. For graphing purposes kinematic data were similarly grouped into blocks of 4 trials and was plotted with the standard error of the mean. EMG data were grouped into four phases: Phase 1 -baseline (initial 28 trials of experiment – the entire pre-exposure period), Phase 2 - early exposure (the initial 32 trials of the exposure period), Phase 3 - late exposure (the final 16 trials of the exposure period), and Phase 4 - early post-exposure (the initial 12 trials of the post-exposure period)

RESULTS

Participants began the experiment by making center-out reaches in three directions without any forces being applied (null field; baseline period). Then, either an abrupt or a gradually increasing force designed to elicit a co-contractile response, and adaptation, was applied.

Kinematics

During the early part of the exposure period there was a marked increase in MLD and RMSE for each of the groups, in response to the onset of the perturbation. A much higher response was observed in the abrupt group than the gradual group for both MLD (abrupt: 1.195 ± 0.227 cm, gradual: 0.759 ± 0.067 cm) and RMSE (abrupt: 12.663 ± 1.03 mm, gradual: 5.967 ± 0.556 mm). A main effect of group ($F_{1,14}$ = 20.24, P = 0.0005) as well as a main effect of time ($F_{1,14}$ = 204.65, P < 0.0001) was found for RMSE, while only a main effect of time was found for the MLD between these two time points ($F_{1,14}$ = 26.83, P = 0.0001).

Over the course of the exposure period there is a notable decrease, as seen from figure 3, in both MLD and RMSE for the abrupt group from the first block (reported above) to the last block (MLD: 0.628 ± 0.112 cm, RMSE: 4.702 ± 0.765 mm) of exposure. The gradual stayed relatively the same from the first block of exposure (reported above) to the last block of the exposure period (MLD: 0.415 ± 0.067 cm, RMSE: 3.261 ± 0.486 mm). For RMSE, there was a main effect for group ($F_{1,14}$ = 21.30, P = 0.0004) and time ($F_{1,14}$ = 88.38, P < 0.0001) over the exposure period. There was also a significant group by time interaction ($F_{1,14}$ = 21.44, P = 0.0004) indicating that there were significant differences between the two groups over the exposure period; post-hoc tests showed that the interaction was driven by a significant group difference during the first block of exposure ($t_{11.09}$ = 5.587, P = 0.0001). No significant effects or interactions were found for MLD over the exposure period aside from previously stated notable decrease in the abrupt group and less pronounced decrease in the gradual group.

To assess whether any aftereffects were present once the perturbation was removed in both groups we compared the last block of the exposure period (reported above) to the first block of the post exposure period (MLD; abrupt: 0.651 ± 0.158 cm, gradual: 0.223 ± 0.013 cm, RMSE; abrupt: 7.802 ± 0.671 mm, gradual: 7.872 ± 0.518 mm). A main effect of group ($F_{1,14}$ = 11.69, P = 0.004) was found for MLD between these blocks. This indicates that there was a significant difference between the two groups as they transitioned back to the unperturbed paradigm of the experiment.

Figure 3. Kinematic adaptation to the task over the course of the entire experiment. Data for **A.)** Maximum Lateral Deviation and **B.)** RMSE was calculated on each trial and plotted for the abrupt (red) and gradual (cyan) groups as the average across a block of 4 trials. **C.)** Exemplar hand path traces for a single participant from each group: abrupt (top) and gradual (bottom).

A.

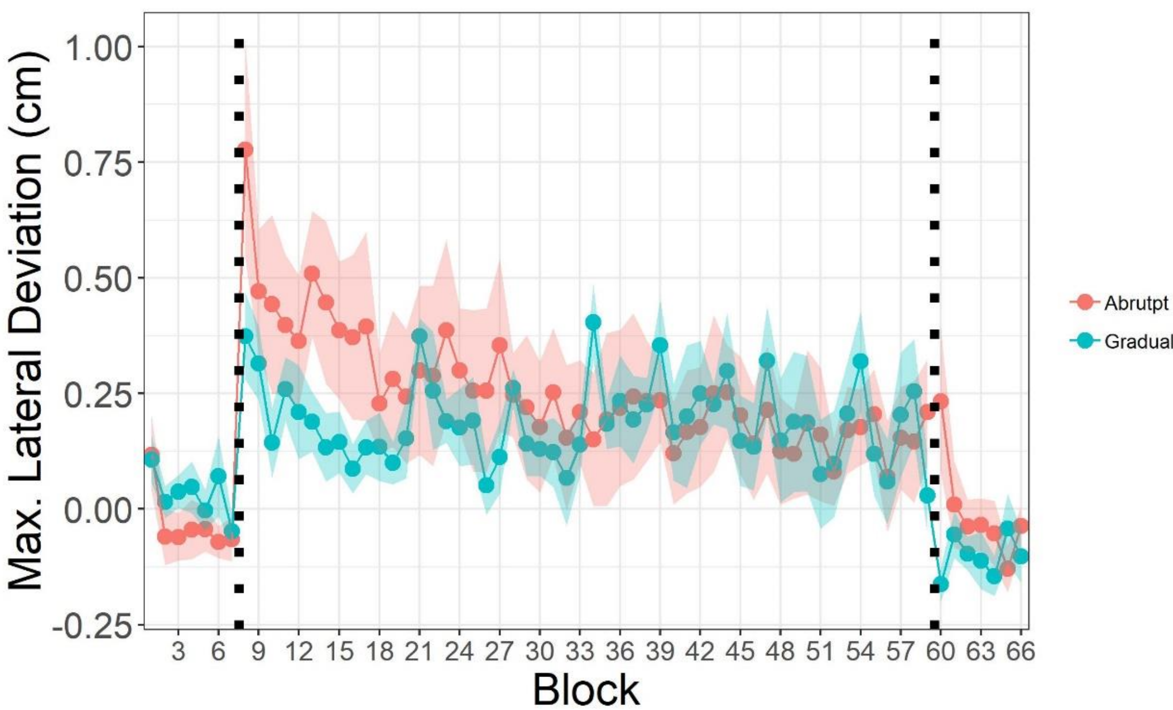
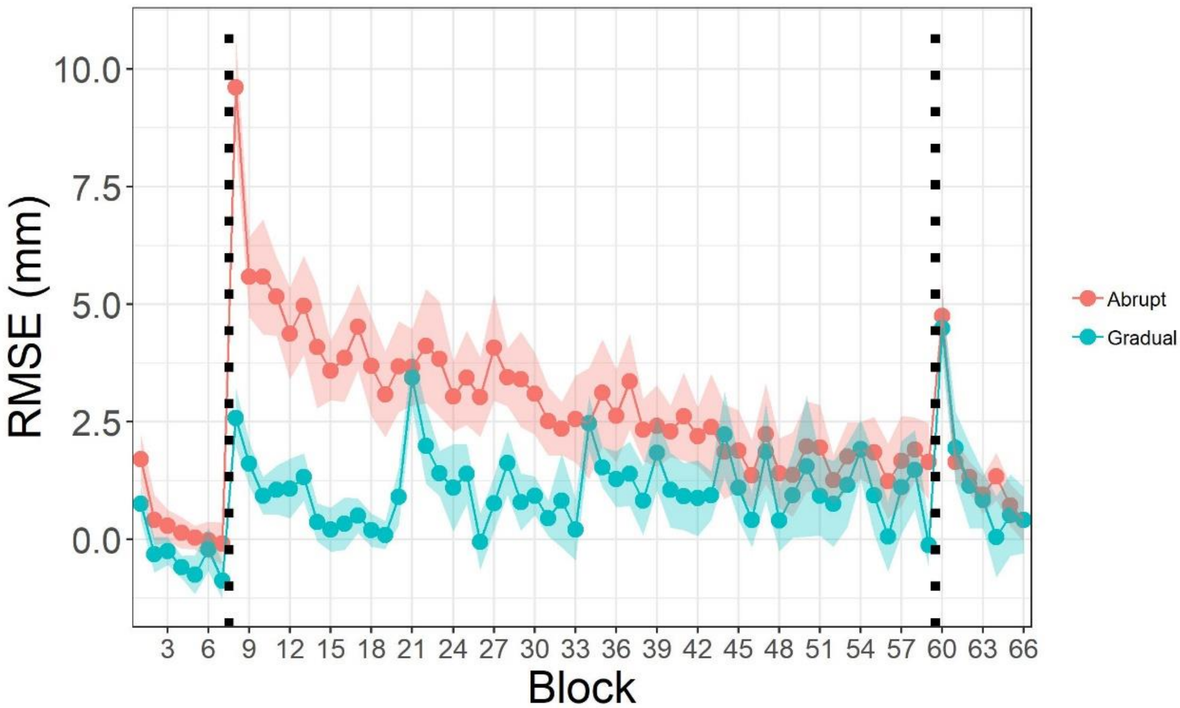


Figure 3. (cont'd)

B.



C.

Abrupt

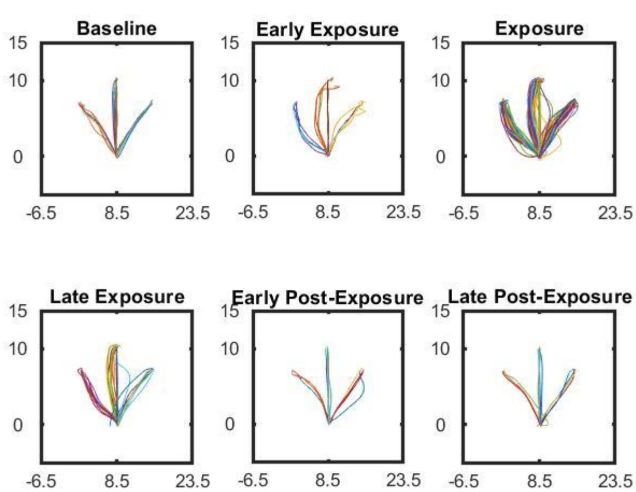
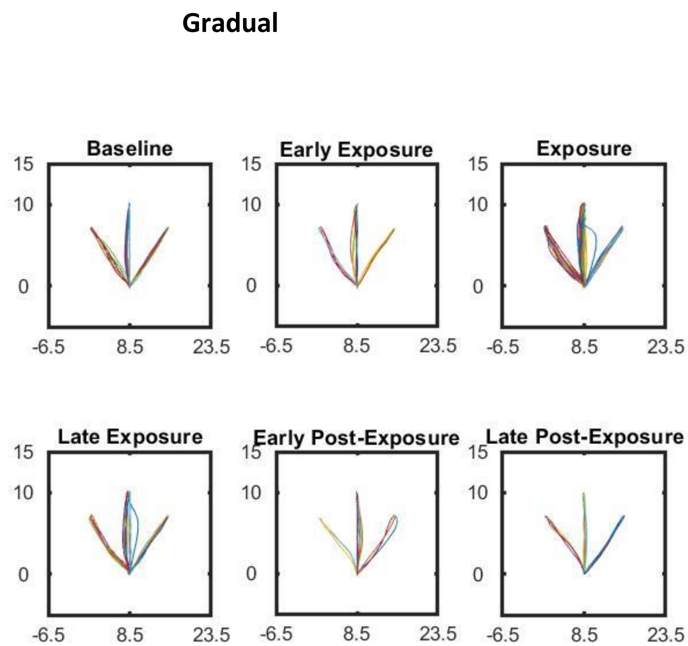


Figure 3. (cont'd)

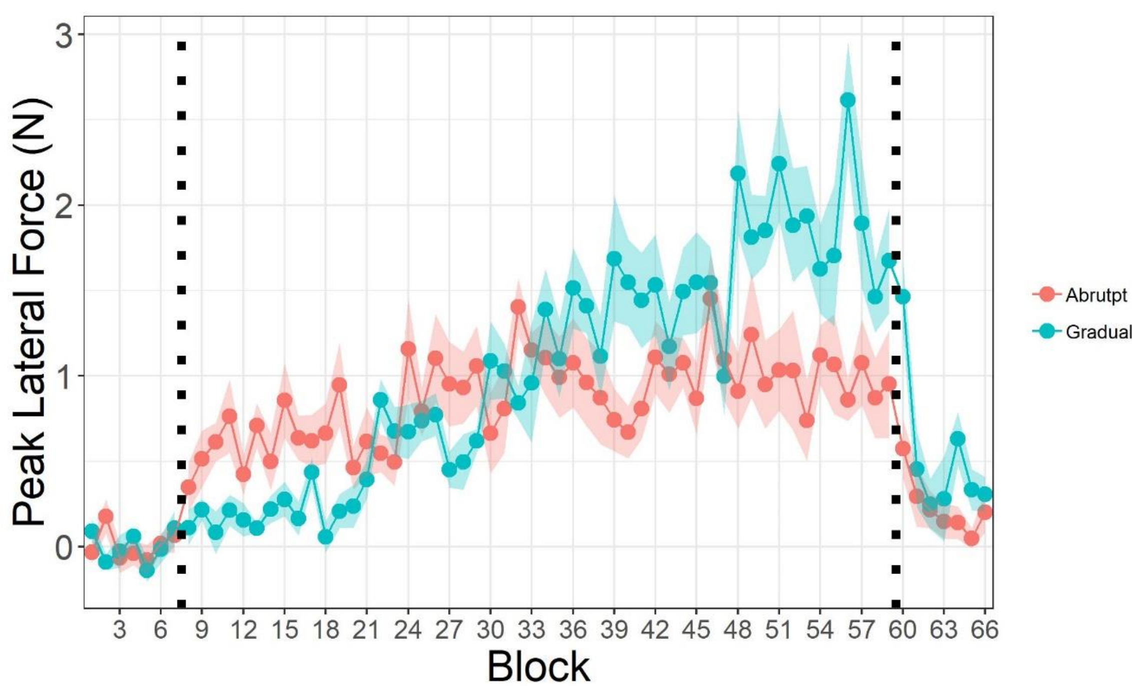


Adaptation

Peak lateral force (PLF) was recorded during channel trials and was used as a measure of the participants' adaptation to the task. PLF was measured very close to zero during the baseline period for both the abrupt and gradual groups. During the exposure period, PLF increased in both groups, with a more noted increase in the gradual group. In order to assess adaptation across the exposure period we compared the initial (block of the exposure phase) adaptation to the final (block of the exposure phase) adaptation between the 2 groups. While both groups performed similarly at the beginning of exposure, (abrupt: 0.886 ± 0.127 N, gradual: 0.655 ± 0.111 N) the amount of force exhibited by each group at final exposure differed much more (abrupt: 0.951 ± 0.314 N, gradual: 2.219 ± 0.312 N). We found a main effect of time ($F_{1,14} = 22.95$, $P = 0.0003$), which was expected for an adaptation study such as this. A marginally significant group by time interaction was also found over the exposure period ($F_{1,14} = 4.51$, $P = 0.05$). Post-hoc comparisons revealed that the groups were not different during the early exposure block (P = 0.2), but differed significantly at the end of exposure ($t_{8.89}$ = -4.905, P = 0.0008) indicating that participants in the gradual group had adapted more completely to the perturbation than those in the abrupt group. Group comparisons were performed on the second block of the post exposure period ($t_{13.43}$ = -3.255, P = 0.006) and revealed that the gradual group retains the force compensation necessary to adapt to the task while the abrupt group does not.

Figure 4 (shown below) shows the PLF over the entire course of the experiment for the abrupt group and the gradual group. As seen in the figure, both groups experienced an increase in the PLF produced over the experiment, with a more noted increase in the gradual group than in the abrupt group over the course of the exposure period.

Figure 4. Peak Lateral Force over the course of the entire experiment. PLF was calculated on each trial and plotted as an average of a block of 4 trials. Data shows the mean ± standard error of the mean across the abrupt (red) group and gradual (cyan) group.



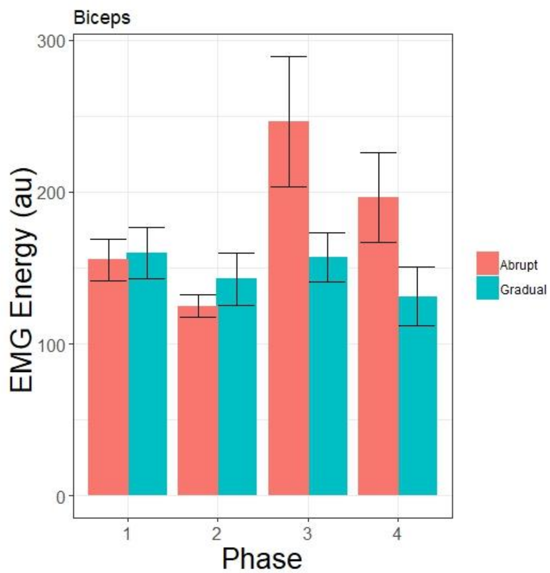
Electromyography

After performing reaches in the null field (pre-exposure period), participants performed reaches in a velocity-dependent force field (exposure period). To identify differences in EMG signal energy we performed separate repeated measures ANOVAs for each muscle with group (abrupt vs. gradual) as the between group factor and phase (baseline, early exposure, late exposure, early post-exposure) as the within group factor, and the EMG signal energy for each channel as the response variable.

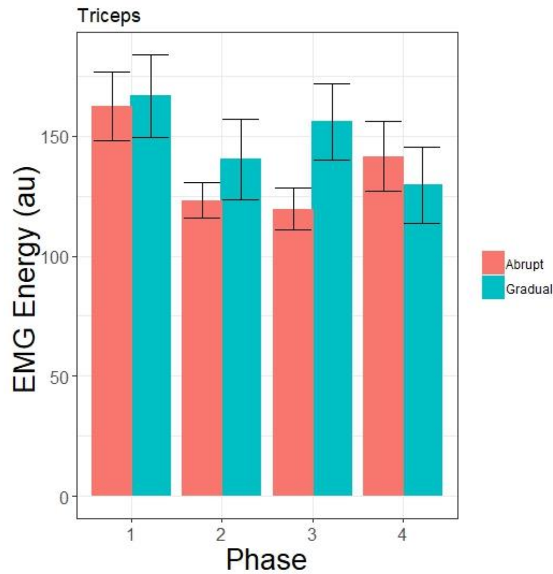
Differences in EMG signal energy over the course of the exposure period were also compared. During this comparison, between the early exposure period and the late exposure period, no significant differences were found between the two groups. However, when assessing aftereffects, by comparing the late exposure period with the early post exposure period, a significant group by phase interaction was found for three of the muscles (triceps: $F_{1,10} = 5.89$, $P = 0.04$; trapezius: $F_{1,10} = 3.83$, $P = 0.08$; pectoralis major: $F_{1,9} = 5.82$, $P = 0.04$). Post-hoc comparisons revealed that the differences between the phases differed significantly from late exposure to early post exposure for the trapezius ($t_{196.33} = 2.81$, $P = 0.005$) and pectoralis major ($t_{197.25} = 2.703$, $P = 0.007$).

Figure 5. EMG energy in the abrupt(red) and gradual(cyan) groups relative to each phase across all muscles. Phases are defined as 1 – Baseline, 2 – Early Exposure, 3 – Late Exposure, 4 – Early Post Exposure.

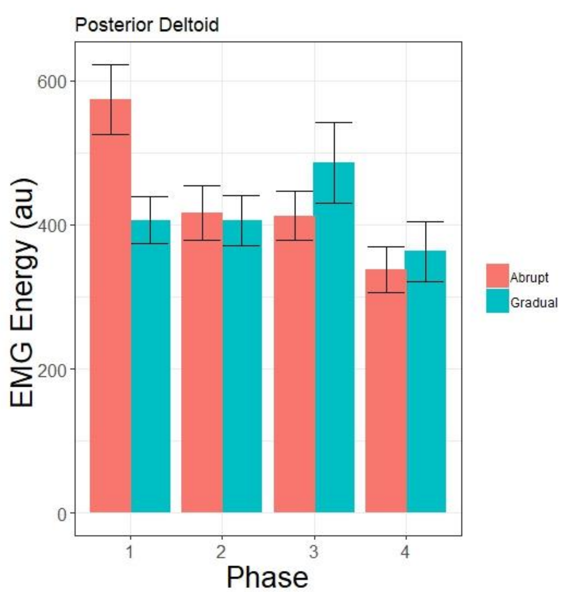
Biceps



Triceps



Posterior Deltoid



Anterior Deltoid

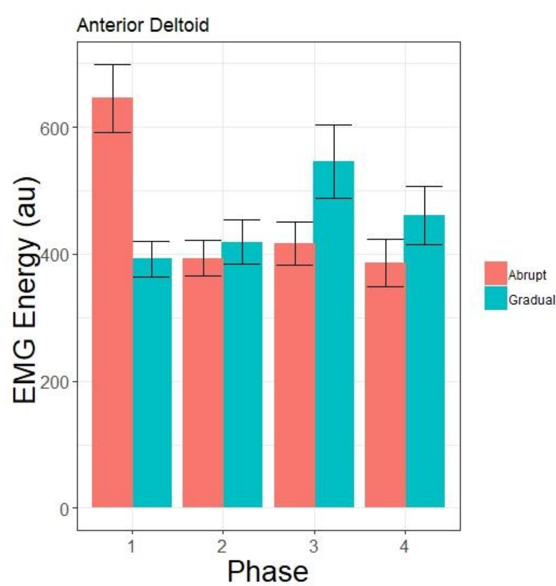
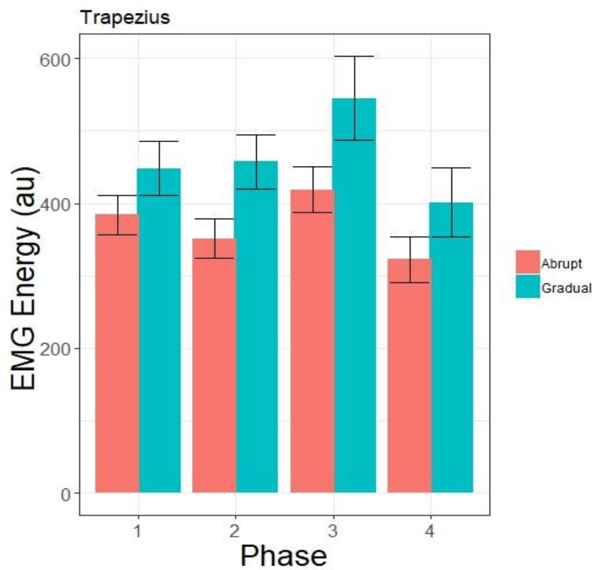
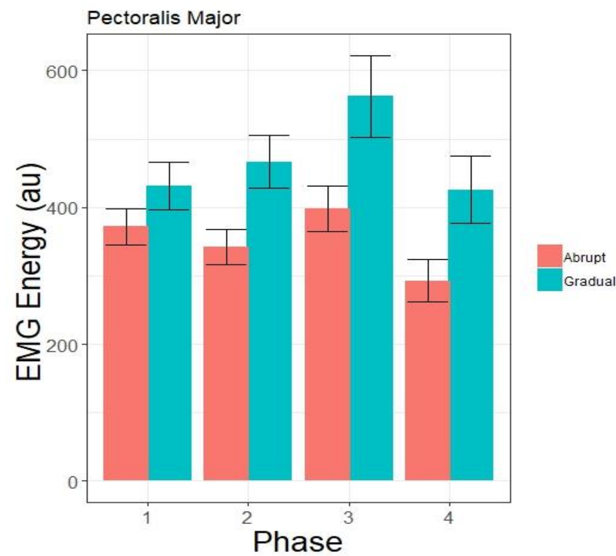


Figure 5. (cont'd)

Trapezius



Pectoralis Major



DISCUSSION

We examined the co-contractile response during adaptation to a dynamic perturbation administered in an abrupt and a gradual paradigm. Participants were randomly assigned to either the group that experienced the abrupt or the gradual dynamic perturbation. The abrupt group experienced a perturbation of 20 N proportionate to their movement velocity while the gradual group experienced incremental increases in the experienced perturbation by 5 N every 52 trials. Adaptation to the task was expressed as the peak lateral force exhibited during channel trials over the course of the experiment. Our results for MLD, RMSE, and PLF are in line with the earlier research (Kagerer et al. 1997, Klassen et al. 2005) that a group exposed to a gradual perturbation will be able to adapt better, as seen through differences between the groups during the transition from the exposure period to the post exposure period. This indicate that the internal model may have been updated to a higher degree during adaptation for the group exposed to the gradual perturbation (Kagerer et al. 1997) than the group exposed to an abrupt perturbation and can be seen clearly from the graphs for each measure that was recorded.

When examining the EMG energy data over the course of the experiment no significance differences in overall EMG energy were seen during the course of the exposure period (the region between early exposure and late exposure period). However, several differences amongst the groups were seen in select muscles (triceps, trapezius, and pectoralis major) between late exposure to early post exposure showing differential muscle activation between the two groups. Both the trapezius and the pectoralis major are involved in maintaining the stability of the shoulder girdle, as well as allowing for smooth movement of the joint. The triceps are the main extensor of the elbow joint and play a large role in the execution of a forward reaching movement.

Dynamic motor adaptation has been shown to be facilitated by exposure to a gradual (rather than an abrupt) perturbation and co-contraction has been shown in earlier studies to enhance the rate of internal model acquisition when participants were instructed to contract their muscles prior to movement onset (Heald et al. 2018). Although there were not global differences between the groups, increased EMG signal energy was more pronounced in the group exposed to the gradual dynamic perturbation than in the group exposed to the abrupt dynamic perturbation in several of the muscles, mainly those involved in shoulder joint stability and movement execution. These findings lend support to our hypothesis that if the co-contractile response was more pronounced in the gradual group than in the abrupt group, then there must be some relationship between the mechanisms of exposure to a gradual perturbation and co-contraction. This indicates that these two mechanisms may have some sort of relationship in how they contribute to dynamic motor adaptation.

These findings fall in line with the existing literature on how individuals adapt to novel movement dynamics. Our results confirm that exposure to a gradually increasing dynamic perturbation yields better adaptation to the task than exposure to an abrupt dynamic perturbation. They also show that along with a higher rate of internal model acquisition being attributed to higher levels of co-contraction, co-contractile response may have a connection to how individuals adapt to a task when different paradigms are present.

APPENDICES

APPENDIX A

Participant Documentation

Sample Data Sheet

Participant Name: ___________________________________ Sex: M F Age: _______

Date: ___________________________ Participant ID:____________________

Group: ABRUPT GRADUAL

_______ Handedness questionnaire: R L Laterality Quotient: _________

KINARM Filename:__________________________ Location:_______________________

________ Calibrated?

	Block Name	Procedure	# Trials
	Baseline	No perturbation, only dominant hand cursor visible	28
	Exposure	Perturbation applied	208
	Post- Exposure	No perturbation, only dominant hand cursor visible	28

ID: Experiment, Group, Number… CoContract_(Abrupt, Gradual)_(01, 02, 03…)

1 trial = 1 move up

KINARM Notes:

Principal Investigator: Dr. Florian Kagerer

Consent Form for Adults

Michigan State University, Movement Neuroscience Laboratory

Project Development of multisensory-motor adaptation

Statement of Age of Participant You are over 18 years of age and willing to participate in a research project conducted by Dr. Florian Kagerer at the Department of Kinesiology, Michigan State University, East Lansing.

Purpose The purpose of this research is to investigate arm and hand coordination under changing movement conditions. The experiment is designed in a way that makes it possible to determine the influence of different task conditions, such as movement direction, distance, and velocity, on arm and hand movements.

Procedures You will first be asked to fill out a brief handedness questionnaire. For the experiment, you will sit comfortably in a chair with your hands resting on a table. You will perform arm movements with one, or both hands, moving joysticks, or a pen on a digitizing tablet, or robotic handles. A computer will store information about the position of your hand(s) during the movement task. You may be asked to participate in one to three sessions. Depending on the experiment, we might attach electrodes to your arms to measure muscle activity or use an eyetracker camera to record your eye movements. Portions of the session may be video recorded for coding or demonstration purposes, using a small camcorder next to the setup. In that case, small markers may be placed on your arms and torso, allowing cameras to record your movements. Each session will last about 45-60 min.

Confidentiality All information collected in the study is strictly confidential (except as you specify on the signed permission form for video and image illustrations), and your name will not be identified at any time. Your data will be grouped with data others provide for reporting and presentation. Data will be stored in a locked file cabinet and on a password protected computer. Only the principal investigator, his collaborators, as well as the MSU Human Research Protection Program (HRPP) will have access to the project data. Your confidentiality will be protected to the maximum extent allowable by law. The consent form, your participant code, or videos made will be retained securely for at least three years after the close of the study.

Risk As a result of your participation in this study, you may experience a modest degree of fatigue from the concentration required during the performance of the test, but there are no other known risks and no long-term effects associated with participation in this study.

Benefits The experiment is not designed to help you specifically, but it may have substantial impact on understanding how the brain controls visually-guided movement.

Freedom to Withdraw and to Ask Questions Your participation is voluntary. You are free to ask questions or to withdraw permission for your participation at any time without penalty. You may refuse to participate in certain procedures or answer certain questions.

INJURY Statement: If you are injured as a result of your participation in this research project, Michigan State University will assist you in obtaining emergency care, if necessary, for your research related injuries. If you have insurance for

Approved by a Michigan State University Institutional Review Board effective 3/14/2019.
This version supersedes all previous versions. MSU Study ID LEGACY10-686.

medical care, your insurance carrier will be billed in the ordinary manner. As with any medical insurance, any costs that are not covered or are in excess of what are paid by your insurance, including deductibles, will be your responsibility. The University's policy is not to provide financial compensation for lost wages, disability, pain or discomfort, unless required by law to do so. This does not mean that you are giving up any legal rights you may have. You may contact Dr. Florian Kagerer at 517-432-9907 with any questions or to report an injury.

Cost and Compensation: As Kinesiology student, you can earn extra credit (equivalent to one quiz/session) by participating; if you do not wish to participate, other ways to earn extra credit will be provided. Psychology students can earn extra credit according to the departments 'Participation in Psychological Research: Information for Students' guidelines. Non-student participants will receive a gift card ($10) for participation.

Principal Investigator

If you have questions about the study, or want to report an injury, please contact:
Dr. Florian Kagerer, Dept. of Kinesiology, Michigan State University
308 W Circle Dr, Rm 126,
East Lansing, MI 48824
Ph: 517.432-9907, email: fkagerer@msu.edu

If you have questions or concerns about your role and rights as a research participant, would like to obtain information or offer input, or would like to register a complaint about this study, you may contact, anonymously if you wish, the Michigan State University's Human Research Protection Program at 517-355-2180, Fax 517-432-4503, or e-mail irb@msu.edu or regular mail at 4000 Collins Rd, Suite #136, Lansing, MI 48910.

Informed Consent Requirements

You are voluntarily making a decision whether or not to participate in the research study described above. Your signature indicates that you have read the information provided above, and have had all of your questions answered. You will be given a copy of this consent form to keep.

We will not share your contact information with anyone outside this project.

I voluntarily agree to be videotaped or photographed while performing the experiment.

☐ Yes ☐ No Initials____________

I voluntarily agree to allow the videotapes or photos to be used later in publications.

☐ Yes ☐ No Initials____________

Name of Participant: ________________________________ **DOB:**___________

Signature: ____

Contact Email & Phone #: ____

Today's Date: ____

MICHIGAN STATE UNIVERSITY

APPROVAL FORM

THESES/DISSERTATION and INSTITUTIONAL REVIEW BOARD APPROVAL

This is to certify that the document entitled:
CO-CONTRACTILE DIFFERENCES DURING ADAPTATION TO ABRUPT AND GRADUAL DYNAMIC PERTURBATIONS

presented by David W. Ptashnik

has been accepted towards fulfillment of the requirements for the degree in

Kinesiology - Master of Science

This form is not considered complete until: a box (yes or no) has been checked in ALL sections regardless of the use of human or animal subjects, the professor and student have filled out their information and signed the form, and the supplemental files section is completely filled out.

University and federal policies and procedures require that all research involving human or animal subjects receive **prior approval** from the appropriate review board. [See Faculty Handbook (http://www.hr.msu.edu/documents/facacadhandbooks/facultyhandbook/index.htm) and the Academic Programs book (http://www.reg.msu.edu/AcademicPrograms/].

HUMAN SUBJECTS

Does the thesis or dissertation you are submitting include research involving human subjects or materials of human origin? (Research involving human subjects includes surveys and telephone interviews used for research; materials of human origin include human blood and /or tissue.)

☑ Yes **IRB Log Number:** 10 - 686
☐ No

If yes, indicate Institutional Review Board (IRB) log number for the **approved** protocol and attach the Institutional Review Board **approval** letter for that protocol to this form. The student's name listed above must appear on the IRB approval letter.

ANIMAL SUBJECTS

Does the thesis or dissertation you are submitting include research involving vertebrate animals in any way?

☐ Yes **AUF Number:** ______
☑ No

If yes, and an animal use form was submitted to the Institutional Animal Care & Use Committee (IACUC), please list the **approval** number below and attach a copy of the IACUC **approval** letter to this form.

If yes, but your project did not need an animal use form, provide a copy of the letter from the IACUC which cites the relevant exclusionary policy.

This thesis/dissertation and the information presented above are approved by the faculty advisor/major professor. ☑ Yes ☐ No

Student's Name (print): David Ptashnik
Student's Signature: [signature]
Date (mm/dd/yyyy): 05/03/2019
Email: ptashnik@msu.edu

Major Prof. Name (print): Florian Kagerer
Major Prof. Signature: [signature]
Date (mm/dd/yyyy): 05/03/2019
Email: fkagerer@msu.edu

Supplemental files associated with the electronic version of this thesis/dissertation are approved by the faculty advisor/major professor. ☐ Yes ☐ No ☑ N/A FK 05/03/19
Major Prof initials/date

The major professor must initial and date in this section regardless of which box is checked.

The Graduate School · Michigan State University · 466 W. Circle Drive, 2nd Floor, Chittenden Hall, East Lansing, MI 48824
Phone: 517-355-0301 Fax: 517-353-3355 Email: msuetds.approval@grd.msu.edu

Revised 4/2017

Handedness Questionnaire

Edinburgh Handedness Inventory - Short Form

Please indicate your preferences in the use of hands in the following activities or objects:

	Always right	Usually right	Both equally	Usually left	Always left
Writing	☐	☐	☐	☐	☐
Throwing	☐	☐	☐	☐	☐
Toothbrush	☐	☐	☐	☐	☐
Spoon	☐	☐	☐	☐	☐

Scoring:

For each item: Always right = 100; Usually right = 50; Both equally = 0; Usually left = -50; Always left = -100

To calculate the Laterality Quotient add the scores for the four items in the scale and divide this by four:

Writing score	____
Throwing score	____
Toothbrush score	____
Spoon score	____
Total	____
Total ÷ 4 (Laterality Quotient)	____

Classification:	Laterality Quotient score:
Left handers	-100 to -61
Mixed handers	-60 to 60
Right handers	61 to 100

APPENDIX B

Raw Data

Raw Data Analysis Code: MATLAB

```
% This is the first attempt at generating code to import data files from
% David Ptashnik's CoContraction tests. In this study, 2 groups were assessed
% on their motor adaptation between 3 targets in the dominant hand under
% two conditions
clear all
close all
% Select the subject directory
str = computer;
if strcmp(str,'MACI64') == 1
  directory = uigetdir('/Volumes/mnl/Data/EEG_test/KINARM_DATA/');
  % Set Currect Directory to subject dir
  cd(directory)
  fname = directory(57:end);
  groupID = directory(44:54);
else
  directory = uigetdir('Z:\Data\Co-Contraction\Kinematic Data');
  % Set Currect Directory to subject dir
  cd(directory)
  fname = directory(46:end);
  groupID = directory(39:44);
end
unZip = zip_load(fname);
data = unZip.c3d;
filename = unZip.filename;
rawData = KINARM_add_hand_kinematics(data(:)); % this adds kinematics to c3d files
filtData = c3d_filter_dblpass(rawData, 'enhanced', 'fc', 10, 'fs', 1000); % 'fc' = cutoff freq, 'fs' = sample rate (don't change fs)
numTrials = size(rawData,1); % Number of Trials
%%
% Find trial numbers
trialNumber = zeros(numTrials,1);
for i = 1:numTrials
  trialNumber(i) = filtData(i).TRIAL.TRIAL_NUM;
end
% Find correct trial order
trialOrder = zeros(numTrials,1);
for i = 1:numTrials
  trialOrder(i) = find(trialNumber == i);
end
% reorders the data to reflect trial number, not TP number
for i = 1:numTrials
  sortData(i) = filtData(trialOrder(i));
end
sortData = sortData';

numDataPoints = zeros(numTrials,1);
for i = 1:numTrials
  numDataPoints(i) = size(sortData(i).Right_HandX,1); % Number of Data points in each trial
end
% Conversion between global and local reference frame (this is due to all
```

```
% x,y hand positions being referenced in the global frame, whereas the
% targets in the target table are referenced in a local frame specified in
% Deterit-E
Tx = sortData(1,1).TARGET_TABLE.X_GLOBAL(1) - sortData(1,1).TARGET_TABLE.X(1);
Ty = sortData(1,1).TARGET_TABLE.Y_GLOBAL(1) - sortData(1,1).TARGET_TABLE.Y(1);

% trial numbers in the sequence
BASELINE = 1:28;
EXP = 29:236;
POST_EXP = 237:264;
% rotation_type =  sortData(1,1).TP_TABLE.Rotation_Type(5);
% rotation_amount =  sortData(1,1).TP_TABLE.Rotation_Angle(5);
%
theta(BASELINE) = 0;
theta(POST_EXP) = 0;
theta(EXP) = 0;
%%
velR = cell(numTrials,1);
onset = zeros(numTrials,1);
offset = zeros(numTrials,1);
wrong_trial = zeros(numTrials,1);
RhX = cell(numTrials,1);
RhY = cell(numTrials,1);
mov_tanL = cell(numTrials,1);
On_err = zeros(numTrials,1);
Off_err = zeros(numTrials,1);
delta_t = 1/1000; % 1/fs
% Find Hand Speed (Magnitude of tangential velocity)
for i = 1:numTrials
  %Calculate hand speed
  % Tangential velocity from kinsym
  RhX{i,1} = sortData(i,1).Right_HandX; %(1:end/2);
  RhY{i,1} = sortData(i,1).Right_HandY; %(1:end/2);
  mov_tanL{i,1} = sqrt(RhX{i,1}.^2 + RhY{i,1}.^2);
  velR{i,1} = diff(mov_tanL{i,1})/delta_t;
  %Since in this study, 1 trial consists of BOTH a reach forward and a
  %reach backwards, need to cut each trial in half so that only outward
  %reach is identified
  velR_2{i,1} = velR{i,1}(1:round(length(velR{i,1})/3)*2);
  if wrong_trial(i) == 0
    onset(i,1) = movOnset2(velR_2{i,1}, 750, 10, 100); % !!! movOnset2 and movOffset3 is working better as of 5/28/15 !!!!
    offset(i,1) = movOffset3_PD(velR_2{i,1} ,50 ,10, 100);
  else
    onset(i,1) = NaN;
    offset(i,1) = NaN;
  end
end
%%
% Find the Cursor Position
% First, translate rotation point to global origin
% Then apply rotation, and translate back to target origin
cursorPosX = cell(numTrials,1);
cursorPosY = cell(numTrials,1);
handPosX = cell(numTrials,1);
handPosY = cell(numTrials,1);
for i = 1:numTrials
  handPosX{i,1} = sortData(i).Right_HandX - sortData(1).TARGET_TABLE.X_GLOBAL(2)/100; % Translate to global origin
  handPosY{i,1} = sortData(i).Right_HandY - sortData(1).TARGET_TABLE.Y_GLOBAL(2)/100;
  cursorPosX{i,1} = handPosX{i,1}.*cosd(theta(i)) - handPosY{i,1}.*sind(theta(i)); % Reverse the rotation
```

```
    cursorPosY{i,1} = handPosX{i,1}.*sind(theta(i)) + handPosY{i,1}.*cosd(theta(i));
    cursorPosX{i,1} = cursorPosX{i,1} + sortData(1).TARGET_TABLE.X_GLOBAL(2)/100; % Translate back to target origin
    cursorPosY{i,1} = cursorPosY{i,1} + sortData(1).TARGET_TABLE.Y_GLOBAL(2)/100;
end
%%
% Find the "Up - 90 deg" trials
upBool = zeros(numTrials,1);
for i = 1:numTrials
    upBool(i) = sortData(i).TRIAL.TP == 1 || sortData(i).TRIAL.TP == 4;
end
upBool = upBool';
upTrials = find(upBool == 1); % Trial numbers of "Up" targets
upTrials = upTrials';
%%
% Find the "Right - 45 deg" trials
rightBool = zeros(numTrials,1);
for i = 1:numTrials
    rightBool(i) = sortData(i).TRIAL.TP == 2 || sortData(i).TRIAL.TP == 5;
end
rightBool = rightBool';
rightTrials = find(rightBool == 1); % Trial numbers of "right - 45 degrees" targets
rightTrials = rightTrials';
%%
% Find the "Left - 135 deg" trials
leftBool = zeros(numTrials,1);
for i = 1:numTrials
    leftBool(i) = sortData(i).TRIAL.TP == 3 || sortData(i).TRIAL.TP == 6;
end
leftBool = leftBool';
leftTrials = find(leftBool == 1); % Trial numbers of "Left-135 deg" targets
leftTrials = leftTrials';
%%
% Find the "Channel" trials
channelBool = zeros(numTrials,1);
for i = 1:numTrials
    channelBool(i) = sortData(i).TRIAL.TP == 7;
end
channelBool = channelBool';
channelTrials = find(channelBool == 1); % Trial numbers of "Channel" targets
channelTrials = channelTrials';
%%
vel = cell(numTrials,1);
velPeak = zeros(numTrials,1);
indPeak = zeros(numTrials,1);
for i = 1:numTrials
    if wrong_trial(i) == 0
        %Calculate hand speed
        vel{i,1} = sqrt(sortData(i,1).Right_HandXVel.^2 + sortData(i,1).Right_HandYVel.^2);
        %Find Peak velocity
        [velPeak(i), indPeak(i)] = max(abs(vel{i,1}));
    end
end
%% IDE
%%%%%%%%%%%%%%%%%%%%%% Initial Directional Error %%%%%%%%%%%%%%%%%%%%%%
% Defined as the angle between the vector from hand position at movement
% onset to target position and a vector pointing to the hand
% position at peak velocity from movement onset hand position
upTargetPos = [sortData(1,1).TARGET_TABLE.X(2) sortData(1,1).TARGET_TABLE.Y(2)];
```

```
rightTargetPos = [sortData(1,1).TARGET_TABLE.X(3) sortData(1,1).TARGET_TABLE.Y(3)];
leftTargetPos = [sortData(1,1).TARGET_TABLE.X(4) sortData(1,1).TARGET_TABLE.Y(4)];
% channelTargetPos = upTargetPos;
xPeak = zeros(numTrials,1);
yPeak = zeros(numTrials,1);
xStart = zeros(numTrials,1);
yStart = zeros(numTrials,1);
imd = zeros(numTrials,2); % initial movement direction (x,y)
itd = zeros(numTrials,2); % initial target direction (x,y)
ide = zeros(numTrials,1);
for i = 1:numTrials
  if wrong_trial(i) == 0
    % Hand Position at movement onset
    xStart(i) = cursorPosX{i,1}(onset(i))*100-Tx; %in cm and workspace ref frame
    yStart(i) = cursorPosY{i,1}(onset(i))*100-Ty;
    % Hand Position at peak velocity
    xPeak(i) = cursorPosX{i,1}(indPeak(i))*100-Tx; %in cm and workspace ref frame
    yPeak(i) = cursorPosY{i,1}(indPeak(i))*100-Ty;
    % Vector from start position to peak velocity position
    imd(i,:) = [xPeak(i) - xStart(i) yPeak(i) - yStart(i)];

    if yPeak(i) > 0
      itd(i,:) = [upTargetPos(1) - xStart(i) upTargetPos(2) - yStart(i)];
    elseif yPeak(i) > 0
      itd(i,:) = [rightTargetPos(1) - xStart(i) rightTargetPos(2) - yStart(i)];
    elseif yPeak(i) > 0
      itd(i,:) = [leftTargetPos(1) - xStart(i) leftTargetPos(2) - yStart(i)];
%      elseif yPeak(i) > 0
%        itd(i,:) = [channelTargetPos(1) - xStart(i) channelTargetPos(2) - yStart(i)];
    end
    ide(i) = acosd(dot(itd(i,:),imd(i,:))./(norm(itd(i,:)).*norm(imd(i,:))));
    % Make ide the the 1st and 3rd quad negative
    if imd(i,1) > 0 && imd(i,2) > 0
      ide(i) = -ide(i);
    elseif imd(i,1) < 0 && imd(i,2) < 0
      ide(i) = -ide(i);
    end
  elseif wrong_trial(i) == 1
    xPeak(i) = NaN;
    yPeak(i) = NaN;
    xStart(i) = NaN;
    yStart(i) = NaN;
    imd(i,:) = NaN;
    ide(i) = NaN;
  end
end
%
% ide(upTrials) = ide(upTrials) + 90;
% ide(channelTrials) = ide(channelTrials) - 90;
%% Confirming Onset and Offset
% Plot movement trajectories and hand speeds to verify onset/offest
% figure(1);
% ang = 0:0.1:2.01*pi;
% r = sortData(i).TARGET_TABLE.Visual_Radius(2);
for i = 1:numTrials
  if wrong_trial(i) == 0
    flag = 2;
    while flag>1
      figure('Position', [100 100 1920/2 1080/2]); %[bottom left corner coords X and Y, W, H]
```

```
        subplot(1,2,1)        plot(sortData(i).TARGET_TABLE.X(2),sortData(i).TARGET_TABLE.Y(2),'Color',[255/255 117/255 
56/255])
        hold on
      plot(sortData(i).TARGET_TABLE.X(3),sortData(i).TARGET_TABLE.Y(3),'k')
        hold on         plot(sortData(i).TARGET_TABLE.X(4),sortData(i).TARGET_TABLE.Y(4),'k')
        hold on
%         plot(sortData(i).TARGET_TABLE.X(2),sortData(i).TARGET_TABLE.Y(2),'k')
%         hold on
        axis([sortData(i).TARGET_TABLE.X(2)-20 sortData(i).TARGET_TABLE.X(2)+20 sortData(i).TARGET_TABLE.Y(4)-10 
sortData(i).TARGET_TABLE.Y(3)+10]);
        axis square;
        title(['Trial: ',num2str(sortData(i).TRIAL.TRIAL_NUM)]); %'  ', 'Theta: ',num2str(ide(i))]);
        hold on
 % NOTE: need to subtract 20 cm from the y data. This is because the
 % reference frame for the output is in global coords, while the target
 % table is in relative coords (this will change, but I can't find where
 % in the data file it occurs. Can't parameterize it now)
        %Correction: this is Tx and Ty
        if sortData(i).TRIAL.TP == 1 || sortData(i).TRIAL.TP == 2 || sortData(i).TRIAL.TP == 3  || sortData(i).TRIAL.TP == 7 % Non-
pertrubed Trials
          plot(sortData(i).Right_HandX(750:end)*100-Tx,sortData(i).Right_HandY(750:end)*100-Ty); % in cm
          hold on
          % NOTE: The 750 comes from movOnset parameters
          plot(sortData(i).Right_HandX(onset(i,1))*100-Tx,sortData(i).Right_HandY(onset(i,1))*100-Ty,'go');
          hold on
          plot(sortData(i).Right_HandX(offset(i,1))*100-Tx,sortData(i).Right_HandY(offset(i,1))*100-Ty,'mo');
        elseif sortData(i).TRIAL.TP == 4 || sortData(i).TRIAL.TP == 5 || sortData(i).TRIAL.TP == 6  % Perturbed Trials
          plot(cursorPosX{i,1}(750:end)*100-Tx,cursorPosY{i,1}(750:end)*100-Ty);
          plot(sortData(i).Right_HandX(750:end)*100-Tx,sortData(i).Right_HandY(750:end)*100-Ty); % in cm
          hold on% in cm
          plot(sortData(i).Right_HandX(onset(i,1))*100-Tx,sortData(i).Right_HandY(onset(i,1))*100-Ty,'go');
          hold on
          plot(sortData(i).Right_HandX(offset(i,1))*100-Tx,sortData(i).Right_HandY(offset(i,1))*100-Ty,'mo')
          hold on
          % NOTE: The 750 comes from the movOnset parameters
          plot(cursorPosX{i,1}(onset(i,1))*100-Tx,cursorPosY{i,1}(onset(i,1))*100-Ty,'go');
          hold on
          if offset(i,1)> size(cursorPosX{i,1},1)
            plot(cursorPosX{i,1},cursorPosY{i,1},'mo');
          else
            plot(cursorPosX{i,1}(offset(i,1))*100-Tx,cursorPosY{i,1}(offset(i,1))*100-Ty,'mo');
          end
        end
        subplot(1,2,2)
        plot(velR{i,1});
        hold  on
        plot(onset(i,1),velR{i,1}(onset(i,1)), 'go');
        hold on
        plot(offset(i,1),velR{i,1}(offset(i,1)), 'mo');
        % Verify Onset/Offset
        button = questdlg('Confirm movement onset/offset?','Onset and offset markers:','Yes','No','Reject','No');
        if strcmp(button,'Yes')
          close(1);
          flag=1;
        elseif strcmp(button,'No') %user can verify if movement onset was computed correctly
          [loc_onset,loc_size]=ginput(2);
          onset(i,1)=round(loc_onset(1)); % replace onset with user defined input
          offset(i,1)=round(loc_onset(2)); % replace offset with use defined input
```

```
            close(1);
            flag=2;
          elseif strcmp(button,'Reject')
            onset(i,1) = NaN;
            offset(i,1) = NaN;
            wrong_trial(i,1)=1;        % Keeps track of rejected trials
            close(1);
            flag=1;
          end
        end
        clear button;
    end
end
% errors = inputdlg('Enter trials numbers which were errors (space-separated)');
% errors = str2num(errors{:});
% wrong_trial(errors) = 1;
%switch Directory
if strcmp(str,'MACI64') == 1
   cd(['/Volumes/mnl/Data/Adaptation/SICI_biman1/Post_Step_1']);
else
   cd(['Z:\Data\Co-Contraction\Kinematic Data\Post_Step_1']);
end
filename = cell2mat(filename);
filename = filename(1:end-4);
save([fname '_postStep1' '.mat'],'sortData');
save([fname '_postStep1' '.mat'],'onset', 'offset', 'wrong_trial','groupID', '-append');



% this loads the .mat file generated in step 1

clear all
close all
% Select the subject file
str = computer;
if strcmp(str,'MACI64') == 1
   cd('/Volumes/mnl/Data/Curl Field/Pilot/Curl field Kin Data/Post_Step_1');
   fname = uigetfile('*rh.mat');
   fname = fname(1:end);
else
   cd('Z:\Data\Co-Contraction\Kinematic Data\Post_Step_1');
   fname = uigetfile('*postStep1.mat');
   fname = fname(1:end-4);
   %fcTrials = xlsread('Z:Data\Adaptation\interference_dosing\Force_Channel_Trials_12_03_16.xlsx','Sheet1');
end
load([fname])
%load([fname '.mat']);
subID = fname(1:21);
%%
numTrials = size(sortData,1); % Number of Trials
fs = 1000; % Sample Rate (Hz)
delta_t = 1/fs; %Sample Period
% Conversion between global and local reference frame (this is due to all
% x,y hand positions being referenced in the global frame, whereas the
% targets in the target table are referenced in a local frame specified in
% Deterit-E
Tx = sortData(1,1).TARGET_TABLE.X_GLOBAL(1) - sortData(1,1).TARGET_TABLE.X(1);
Ty = sortData(1,1).TARGET_TABLE.Y_GLOBAL(1) - sortData(1,1).TARGET_TABLE.Y(1);
```

```
%visual baseline, kinesthetic baseline, exposure, and post-exposure
%trial numbers in the sequence
BASELINE = 1:28;
EXP = 29:236;
POST_EXP = 237:264;
%
% rotation_type =  sortData(1,1).TP_TABLE.Rotation_Type(5);
% rotation_amount =  sortData(1,1).TP_TABLE.Rotation_Angle(5);
theta(BASELINE) = 0;
theta(EXP) = 0;
theta(POST_EXP) = 0;
% if rotation_type == 1 && strcmp(sortData(1,1).EXPERIMENT.ACTIVE_ARM, 'RIGHT') == 1
%     theta(EXP) = rotation_amount;
% else
%     theta(EXP) = 0;
% end
%%
% Find the Cursor Position
% First, translate rotation point to global origin
% Then apply rotation, and translate back to target origin
cursorPosX = cell(numTrials,1);
cursorPosY = cell(numTrials,1);
handPosX = cell(numTrials,1);
handPosY = cell(numTrials,1);
 for i = 1:numTrials
   handPosX{i,1} = sortData(i).Right_HandX - sortData(1).TARGET_TABLE.X_GLOBAL(1)/100; % Translate to global origin
   handPosY{i,1} = sortData(i).Right_HandY - sortData(1).TARGET_TABLE.Y_GLOBAL(1)/100;
   cursorPosX{i,1} = handPosX{i,1}.*cosd(theta(i)) - handPosX{i,1}.*sind(theta(i)); % Reverse the rotation
   cursorPosY{i,1} = handPosY{i,1}.*sind(theta(i)) + handPosY{i,1}.*cosd(theta(i));
%
   cursorPosX{i,1} = cursorPosX{i,1} + sortData(1).TARGET_TABLE.X_GLOBAL(1)/100; % Translate back to target origin
   cursorPosY{i,1} = cursorPosY{i,1} + sortData(1).TARGET_TABLE.Y_GLOBAL(1)/100;
 end
%%
% Find the "Up- 90 degrees" trials
upBool = zeros(numTrials,1);
for i = 1:numTrials
   upBool(i) = sortData(i).TRIAL.TP == 1 || sortData(i).TRIAL.TP == 4;
end
upBool = upBool';
upTrials = find(upBool == 1); % Trial numbers of "Up- degrees" targets
upTrials = upTrials';
%%
% Find the "Right - 45 degrees" trials
rightBool = zeros(numTrials,1);
for i = 1:numTrials
   rightBool(i) = sortData(i).TRIAL.TP == 2 || sortData(i).TRIAL.TP == 5;
end
rightBool = rightBool';
rightTrials = find(rightBool == 1); % Trial numbers of "Right - 45 degrees" targets
rightTrials = rightTrials';
%%
% Find the "Left - 135 degrees" trials
leftBool = zeros(numTrials,1);
for i = 1:numTrials
   leftBool(i) = sortData(i).TRIAL.TP == 3 || sortData(i).TRIAL.TP == 6;
end
leftBool = leftBool';
```

```
leftTrials = find(leftBool == 1); % Trial numbers of "Left - 135 degrees" targets
leftTrials = leftTrials';
%%
% Find the "Channel" trials
channelBool = zeros(numTrials,1);
for i = 1:numTrials
   channelBool(i) = sortData(i).TRIAL.TP == 7;
end
channelTrials = find(channelBool == 1); % Trial numbers of "Channel" targets
channelTrials = channelTrials';
%%
numDataPoints = zeros(numTrials,1);
for i = 1:numTrials
   numDataPoints(i) = size(sortData(i).Right_HandX,1); % Number of Data points in each trial
end
vel = cell(numTrials,1);
velPeak = zeros(numTrials,1);
indPeak = zeros(numTrials,1);
for i = 1:numTrials
   %Calculate hand speed
   vel{i,1} = sqrt(sortData(i,1).Right_HandXVel.^2 + sortData(i,1).Right_HandYVel.^2);
   %Find Peak velocity
   if wrong_trial(i) == 1
      velPeak(i,1) = NaN;
      indPeak(i,1) = NaN;
   else
      [velPeak(i), indPeak(i)] = max(abs(vel{i,1}(1:offset(i))));
   end
end

%% Movement Time (MT)
MT = (offset - onset)/fs;
MT(wrong_trial==1) = NaN;
%% MT outlier analysis
data = outlier_t(MT(upTrials(1:7))); % Outlier for baseline
MT_c(upTrials(1:7)) = data;
data = outlier_t(MT(rightTrials(1:7)));
MT_c(rightTrials(1:7)) = data;
data = outlier_t(MT(leftTrials(1:7)));
MT_c(leftTrials(1:7)) = data;
data = outlier_t(MT(channelTrials(1:7)));
MT_c(channelTrials(1:7)) = data;
clear data;
% data = outlier_t(MT(upTrials(14:23))); % Outlier for first 20 exposure
% MT_c(upTrials(14:23)) = dat32:a;
% data = outlier_t(MT(downTrials(14:23)));
% MT_c(downTrials(14:23)) = data;
% clear data;
MT_c(upTrials(8:11)) = MT(upTrials(8:11));
MT_c(rightTrials(8:11)) = MT(rightTrials(8:11));
MT_c(leftTrials(8:11)) = MT(leftTrials(8:11));
MT_c(channelTrials(8:11)) = MT(channelTrials(8:11));
%%
data = outlier_t(MT(upTrials(12:59))); % Outlier for last 188 exposure
MT_c(upTrials(12:59)) = data;
data = outlier_t(MT(rightTrials(12:59)));
MT_c(rightTrials(12:59)) = data;
data = outlier_t(MT(leftTrials(12:59))); % Outlier for last 188 exposure
MT_c(leftTrials(12:59)) = data;
```

```
data = outlier_t(MT(channelTrials(12:59)));
MT_c(channelTrials(12:59)) = data;
clear data;
%%
% data = outlier_t(MT(upTrials(43:45))); % Outlier for first 10 post-exp
% MT_c(upTrials(43:45)) = data;
% data = outlier_t(MT(downTrials(43:45)));
% MT_c(downTrials(43:45)) = data;
% data = outlier_t(MT(leftTrials(43:45))); % Outlier for first 10 post-exp
% MT_c(leftTrials(43:45)) = data;
% data = outlier_t(MT(rightTrials(43:45)));
% MT_c(rightTrials(43:45)) = data;
% clear data;
MT_c(upTrials(60:61)) = MT(upTrials(60:61));
MT_c(rightTrials(60:61)) = MT(rightTrials(60:61));
MT_c(leftTrials(60:61)) = MT(leftTrials(60:61));
MT_c(channelTrials(60:61)) = MT(channelTrials(60:61));
%%
data = outlier_t(MT(upTrials(61:66))); % Outlier for last 18 post-exp
MT_c(upTrials(61:66)) = data;
data = outlier_t(MT(rightTrials(61:66)));
MT_c(rightTrials(61:66)) = data;
data = outlier_t(MT(leftTrials(61:66))); % Outlier for last 18 post-exp
MT_c(leftTrials(61:66)) = data;
data = outlier_t(MT(channelTrials(61:66)));
MT_c(channelTrials(61:66)) = data;
clear data;
%%
% transpose and calculate standardized variable
MT_c = MT_c';
bkup_mean = nanmean(MT_c(upTrials(1:7)));
bkup_std = nanstd(MT_c(upTrials(1:7)));
MT_up_st = (MT_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(MT_c(rightTrials(1:7)));
bkright_std = nanstd(MT_c(rightTrials(1:7)));
MT_right_st = (MT_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(MT_c(leftTrials(1:7)));
bkleft_std = nanstd(MT_c(leftTrials(1:7)));
MT_left_st = (MT_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(MT_c(channelTrials(1:7)));
bkchannel_std = nanstd(MT_c(channelTrials(1:7)));
MT_channel_st = (MT_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting Code for MT
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),MT(upTrials(1:7)),'bo');
hold on
plot(upTrials(1:7),MT_c(upTrials(1:7)),'bx');
hold on
plot(rightTrials(1:7),MT(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),MT_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),MT(leftTrials(1:7)),'go');
```

```
hold on
plot(leftTrials(1:7),MT_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),MT(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),MT_c(channelTrials(1:7)),'mx');
axis([0 28 0 3]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:1:7,'YTickLabel',0:1:7,'FontName','Arial','FontSize',10); ylabel('MT [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.2 0.2 0.5 0.6]); hold on; %8:58
plot(upTrials(8:58)-EXP(1)+1,MT(upTrials(8:58)),'bo');
hold on
plot(upTrials(8:58)-EXP(1)+1,MT_c(upTrials(8:58)),'bx');
hold on
plot(rightTrials(8:58)-EXP(1)+1,MT(rightTrials(8:58)),'ro');
hold on
plot(rightTrials(8:58)-EXP(1)+1,MT_c(rightTrials(8:58)),'rx');
hold on
plot(leftTrials(8:58)-EXP(1)+1,MT(leftTrials(8:58)),'go');
hold on
plot(leftTrials(8:58)-EXP(1)+1,MT_c(leftTrials(8:58)),'gx');
hold on
plot(channelTrials(8:58)-EXP(1)+1,MT(channelTrials(8:58)),'mo');
hold on
plot(channelTrials(8:58)-EXP(1)+1,MT_c(channelTrials(8:58)),'mx');
axis([0 208 0 3]); set(gca,'LineWidth',2,'XTick',[1 52 105 157 208],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;%59:66
plot(upTrials(59:66)-POST_EXP(1)+1,MT(upTrials(59:66)),'bo');
hold on
plot(upTrials(59:66)-POST_EXP(1)+1,MT_c(upTrials(59:66)),'bx');
hold on
plot(rightTrials(59:66)-POST_EXP(1)+1,MT(rightTrials(59:66)),'ro');
hold on
plot(rightTrials(59:66)-POST_EXP(1)+1,MT_c(rightTrials(59:66)),'rx');
hold on
plot(leftTrials(59:66)-POST_EXP(1)+1,MT(leftTrials(59:66)),'go');
hold on
plot(leftTrials(59:66)-POST_EXP(1)+1,MT_c(leftTrials(59:66)),'gx');
hold on
plot(channelTrials(59:66)-POST_EXP(1)+1,MT(channelTrials(59:66)),'mo');
hold on
plot(channelTrials(59:66)-POST_EXP(1)+1,MT_c(channelTrials(59:66)),'mx');
axis([0 28 0 3]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['Post-Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title([ 'Post-Exposure ']);
%% IDE
%%%%%%%%%%%%%%%%%%%%%% Initial Directional Error %%%%%%%%%%%%%%%%%%%%%%
% Defined as the angle between the vector from hand position at movement
% onset to target position and a vector pointing to the hand
% position at peak velocity from movement onset hand position
startPos = [sortData(1,1).TARGET_TABLE.X(1) sortData(1,1).TARGET_TABLE.Y(2)];
upTargetPos = [sortData(1,1).TARGET_TABLE.X(2) sortData(1,1).TARGET_TABLE.Y(2)];
rightTargetPos = [sortData(1,1).TARGET_TABLE.X(3) sortData(1,1).TARGET_TABLE.Y(3)];
leftTargetPos = [sortData(1,1).TARGET_TABLE.X(4) sortData(1,1).TARGET_TABLE.Y(4)];
channelTargetPos = [sortData(1,1).TARGET_TABLE.X(2) sortData(1,1).TARGET_TABLE.Y(2)];
xPeak = zeros(numTrials,1);
```

```
yPeak = zeros(numTrials,1);
xStart = zeros(numTrials,1);
yStart = zeros(numTrials,1);
imd = zeros(numTrials,2); % initial movement direction (x,y)
itd = zeros(numTrials,2); % initial target direction (x,y)
ide = zeros(numTrials,1);
for i = 1:numTrials
  if wrong_trial(i) == 0
    % Hand Position at movement onset
    xStart(i) = cursorPosX{i,1}(onset(i))*100-Tx; %in cm and workspace ref frame
    yStart(i) = cursorPosY{i,1}(onset(i))*100-Ty;
    % Hand Position at peak velocity
    xPeak(i) = cursorPosX{i,1}(indPeak(i))*100-Tx; %in cm and workspace ref frame
    yPeak(i) = cursorPosY{i,1}(indPeak(i))*100-Ty;
    % Vector from start position to peak velocity position
    imd(i,:) = [xPeak(i) - xStart(i) yPeak(i) - yStart(i)];
    if sortData(i).TRIAL.TP == 1 || sortData(i).TRIAL.TP == 4
      itd(i,:) = [upTargetPos(1) - xStart(i) upTargetPos(2) - yStart(i)];
    elseif sortData(i).TRIAL.TP == 2 || sortData(i).TRIAL.TP == 5
      itd(i,:) = [rightTargetPos(1) - xStart(i) rightTargetPos(2) - yStart(i)];
    elseif sortData(i).TRIAL.TP == 3 || sortData(i).TRIAL.TP == 6
      itd(i,:) = [leftTargetPos(1) - xStart(i) leftTargetPos(2) - yStart(i)];
    elseif sortData(i).TRIAL.TP == 7
      itd(i,:) = [channelTargetPos(1) - xStart(i) channelTargetPos(2) - yStart(i)];
    end
  ide(i) = acosd(dot(itd(i,:),imd(i,:))./(norm(itd(i,:)).*norm(imd(i,:))));
  % Make ide the the 1st and 3rd quad negative for up and down trials
%    if imd(i,1) > 0 && imd(i,2) > 0 && (upBool(i) == 1) % ChannelBool(i) == 1)
%      ide(i) = -ide(i);
%    elseif imd(i,1) < 0 && imd(i,2) < 0 && (upBool(i) == 1)  %ChannelBool(i) == 1)
%      ide(i) = -ide(i);
%    end
%
%    % Make ide in the 2nd and 4th quads negative for right and left trials
%    if imd(i,1) > 0 && imd(i,2) < 0 && (leftBool(i) == 1 || rightBool(i) == 1)
%      ide(i) = -ide(i);
%    elseif imd(i,1) < 0 && imd(i,2) > 0 && (leftBool(i) == 1 || rightBool(i) == 1)
%      ide(i) = -ide(i);
%    end
  else
    xPeak(i) = NaN;
    yPeak(i) = NaN;
    xStart(i) = NaN;
    yStart(i) = NaN;
    imd(i,:) = NaN;
    ide(i) = NaN;
  end
end
% ide(EXP) = ide(EXP)-40;
%% ide outlier analysis
data = outlier_t(ide(upTrials(1:7)));% baseline is 1:7
ide_c(upTrials(1:7)) = data;
data = outlier_t(ide(rightTrials(1:7)));
ide_c(rightTrials(1:7)) = data;
data = outlier_t(ide(leftTrials(1:7)));
ide_c(leftTrials(1:7)) = data;
data = outlier_t(ide(channelTrials(1:7)));
ide_c(channelTrials(1:7)) = data;
```

```
clear data;
% data = outlier_t(ide(upTrials(14:23)));
% ide_c(upTrials(14:23)) = data;
% data = outlier_t(ide(downTrials(14:23)));
% ide_c(downTrials(14:23)) = data;
% clear data;
ide_c(upTrials(8:11)) = ide(upTrials(8:11));
ide_c(rightTrials(8:11)) = ide(rightTrials(8:11));% initial exposure is 8:11
ide_c(leftTrials(8:11)) = ide(leftTrials(8:11));
ide_c(channelTrials(8:11)) = ide(channelTrials(8:11));
data = outlier_t(ide(upTrials(12:59)));% exposure is 12:59
ide_c(upTrials(12:59)) = data;
data = outlier_t(ide(rightTrials(12:59)));
ide_c(rightTrials(12:59)) = data;
data = outlier_t(ide(leftTrials(12:59)));
ide_c(leftTrials(12:59)) = data;
data = outlier_t(ide(channelTrials(12:59)));
ide_c(channelTrials(12:59)) = data;
clear data;
data = outlier_t(ide(upTrials(60:61)));% initial post exposure is 60:61
ide_c(upTrials(60:61)) = data;
data = outlier_t(ide(rightTrials(60:61)));
ide_c(rightTrials(60:61)) = data;
data = outlier_t(ide(leftTrials(60:61)));
ide_c(leftTrials(60:61)) = data;
data = outlier_t(ide(channelTrials(60:61)));
ide_c(channelTrials(60:61)) = data;
clear data;
data = outlier_t(ide(upTrials(62:66)));% rest of post exposure is 62:66
ide_c(upTrials(62:66)) = data;
data = outlier_t(ide(rightTrials(62:66)));
ide_c(rightTrials(62:66)) = data;
data = outlier_t(ide(leftTrials(62:66)));
ide_c(leftTrials(62:66)) = data;
data = outlier_t(ide(channelTrials(62:66)));
ide_c(channelTrials(62:66)) = data;
clear data;
%%
% transpose and calculate standardized variable
ide_c = ide_c';
bkup_mean = nanmean(ide_c(upTrials(1:7)));
bkup_std = nanstd(ide_c(upTrials(1:7)));
ide_up_st = (ide_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(ide_c(rightTrials(1:7)));
bkright_std = nanstd(ide_c(rightTrials(1:7)));
ide_right_st = (ide_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(ide_c(leftTrials(1:7)));
bkleft_std = nanstd(ide_c(leftTrials(1:7)));
ide_left_st = (ide_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(ide_c(channelTrials(1:7)));
bkchannel_std = nanstd(ide_c(channelTrials(1:7)));
ide_channel_st = (ide_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting Code for ide
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
```

```
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),ide(upTrials(1:7)),'bo');
hold on
plot(upTrials(1:7),ide_c(upTrials(1:7)),'bx');
hold on
plot(rightTrials(1:7),ide(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),ide_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),ide(leftTrials(1:7)),'go');
hold on
plot(leftTrials(1:7),ide_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),ide(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),ide_c(channelTrials(1:7)),'mx');
axis([0 28 0 50]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:5:80,'YTickLabel',0:5:80,'FontName','Arial','FontSize',10);
ylabel('ide [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.2 0.2 0.5 0.6]); hold on;
plot(upTrials(8:59)-EXP(1)+1,ide(upTrials(8:59)),'bo');
hold on
plot(upTrials(8:59)-EXP(1)+1,ide_c(upTrials(8:59)),'bx');
hold on
plot(rightTrials(8:59)-EXP(1)+1,ide(rightTrials(8:59)),'ro');
hold on
plot(rightTrials(8:59)-EXP(1)+1,ide_c(rightTrials(8:59)),'rx');
hold on
plot(leftTrials(8:59)-EXP(1)+1,ide(leftTrials(8:59)),'go');
hold on
plot(leftTrials(8:59)-EXP(1)+1,ide_c(leftTrials(8:59)),'gx');
hold on
plot(channelTrials(8:59)-EXP(1)+1,ide(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,ide_c(channelTrials(8:59)),'mx');
axis([0 208 0 50]); set(gca,'LineWidth',2,'XTick',[29 81 133 185 236],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
plot(upTrials(60:66)-POST_EXP(1)+1,ide(upTrials(60:66)),'bo');
hold on
plot(upTrials(60:66)-POST_EXP(1)+1,ide_c(upTrials(60:66)),'bx');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,ide(rightTrials(60:66)),'ro');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,ide_c(rightTrials(60:66)),'rx');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,ide(leftTrials(60:66)),'go');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,ide_c(leftTrials(60:66)),'gx');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,ide(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,ide_c(channelTrials(60:66)),'mx');
axis([0 28 0 50]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['Post-Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title(['Post Exposure']);
%% MLD
```

```
%%%%%%%%%%%%%%%%%%%%%Max. Lateral Deviation%%%%%%%%%%%%%%%%%%%%
xPeak = zeros(numTrials,1);
yPeak = zeros(numTrials,1);
xStart = zeros(numTrials,1);
yStart = zeros(numTrials,1);
xEnd = zeros(numTrials,1);
yEnd = zeros(numTrials,1);
dev_theta = zeros(numTrials,1);
MLD = zeros(numTrials,1);
PosX = cursorPosX;
PosY = cursorPosY;
for i = 1:numTrials
    if wrong_trial(i) == 0
        % Hand Position at movement onset
        xo(i) = cursorPosX{i,1}(onset(i)); %in m in global
        yo(i) = cursorPosY{i,1}(onset(i));
%       %Hand Position at movement offset
%       xEnd(i) = cursorPosX{i,1}(offset(i))*100-Tx;
%       yEnd(i) = cursorPosY{i,1}(offset(i))*100-Ty;
%       % Hand Position at peak velocity
%       xPeak(i) = cursorPosX{i,1}(indPeak(i))*100-Tx; %in cm and workspace ref frame
%       yPeak(i) = cursorPosY{i,1}(indPeak(i))*100-Ty;

        if sortData(i).TRIAL.TP == 1 || sortData(i).TRIAL.TP == 4
            MLD(i)= abs(xo(i) - min(PosX{i,1}))*100;
        elseif sortData(i).TRIAL.TP == 2 || sortData(i).TRIAL.TP == 5
            dev_theta = 45;
            handPosX{i,1} = sortData(i).Right_HandX(onset(i):offset(i)) - sortData(i).Right_HandX(onset(i)); % Translate to global
origin
            handPosY{i,1} = sortData(i).Right_HandY(onset(i):offset(i)) - sortData(i).Right_HandY(onset(i));
            PosX{i,1} = handPosX{i,1}.*cosd(dev_theta) - handPosY{i,1}.*sind(dev_theta); % rotate
            PosY{i,1} = handPosX{i,1}.*sind(dev_theta) + handPosY{i,1}.*cosd(dev_theta);
            MLD_x(i) = min(PosX{i,1});
            MLD(i)  = abs(xStart(i)- MLD_x(i))*100;
        elseif sortData(i).TRIAL.TP == 3 || sortData(i).TRIAL.TP == 6
           dev_theta = -45;
            handPosX{i,1} = sortData(i).Right_HandX(onset(i):offset(i)) - sortData(i).Right_HandX(onset(i)); % Translate to global
origin
            handPosY{i,1} = sortData(i).Right_HandY(onset(i):offset(i)) - sortData(i).Right_HandY(onset(i));
            PosX{i,1} = handPosX{i,1}.*cosd(dev_theta) - handPosY{i,1}.*sind(dev_theta); % rotate
            PosY{i,1} = handPosX{i,1}.*sind(dev_theta) + handPosY{i,1}.*cosd(dev_theta);
            MLD_x(i) = min(PosX{i,1});
            MLD(i)  = abs(xStart(i)- MLD_x(i))*100;
        elseif sortData(i).TRIAL.TP == 7
            MLD(i) = NaN;
        end
    end
end
%% MLD outlier analysis
data = outlier_t(MLD(upTrials(1:7))); % Outlier for baseline
MLD_c(upTrials(1:7)) = data;
data = outlier_t(MLD(rightTrials(1:7)));
MLD_c(rightTrials(1:7)) = data;
data = outlier_t(MLD(leftTrials(1:7))); % Outlier for  baseline
MLD_c(leftTrials(1:7)) = data;
data = outlier_t(MLD(channelTrials(1:7)));
MLD_c(channelTrials(1:7)) = data;
clear data;
% data = outlier_t(rmse(upTrials(14:23))); % Outlier for first 20 exposure
```

```
% rmse_c(upTrials(14:23)) = data;
% data = outlier_t(rmse(downTrials(14:23)));
% rmse_c(downTrials(14:23)) = data;
% clear data;
MLD_c(upTrials(8:11)) = MLD(upTrials(8:11));
MLD_c(rightTrials(8:11)) = MLD(rightTrials(8:11));
MLD_c(leftTrials(8:11)) = MLD(leftTrials(8:11));
MLD_c(channelTrials(8:11)) = MLD(channelTrials(8:11));
data = outlier_t(MLD(upTrials(12:59))); % Outlier for last 188 exposure
MLD_c(upTrials(12:59)) = data;
data = outlier_t(MLD(rightTrials(12:59)));
MLD_c(rightTrials(12:59)) = data;
data = outlier_t(MLD(leftTrials(12:59))); % Outlier for last 188 exposure
MLD_c(leftTrials(12:59)) = data;
data = outlier_t(MLD(channelTrials(12:59)));
MLD_c(channelTrials(12:59)) = data;
clear data;
% data = outlier_t(rmse(upTrials(43:45))); % Outlier for first 10 post-exp
% rmse_c(upTrials(43:45)) = data;
% data = outlier_t(rmse(downTrials(43:45)));
% rmse_c(downTrials(43:45)) = data;
% clear data;
MLD_c(upTrials(60:61)) = MLD(upTrials(60:61));
MLD_c(rightTrials(60:61)) = MLD(rightTrials(60:61));
MLD_c(leftTrials(60:61)) = MLD(leftTrials(60:61));
MLD_c(channelTrials(60:61)) = MLD(channelTrials(60:61));
data = outlier_t(MLD(upTrials(62:66))); % Outlier for last 18 post-exp
MLD_c(upTrials(62:66)) = data;
data = outlier_t(MLD(rightTrials(62:66)));
MLD_c(rightTrials(62:66)) = data;
data = outlier_t(MLD(leftTrials(62:66))); % Outlier for last 18 post-exp
MLD_c(leftTrials(62:66)) = data;
data = outlier_t(MLD(channelTrials(62:66)));
MLD_c(channelTrials(62:66)) = data;
clear data;
% transpose and calculate standardized variable
MLD_c = MLD_c';
bkup_mean = nanmean(MLD_c(upTrials(1:7)));
bkup_std = nanstd(MLD_c(upTrials(1:7)));
MLD_up_st = (MLD_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(MLD_c(rightTrials(1:7)));
bkright_std = nanstd(MLD_c(rightTrials(1:7)));
MLD_right_st = (MLD_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(MLD_c(leftTrials(1:7)));
bkleft_std = nanstd(MLD_c(leftTrials(1:7)));
MLD_left_st = (MLD_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(MLD_c(channelTrials(1:7)));
bkchannel_std = nanstd(MLD_c(channelTrials(1:7)));
MLD_channel_st = (MLD_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting code for MLD
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),MLD(upTrials(1:7)),'bo');
hold on
```

```
plot(upTrials(1:7),MLD_c(upTrials(1:7)),'bx');
hold on
plot(rightTrials(1:7),MLD(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),MLD_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),MLD(leftTrials(1:7)),'go');
hold on
plot(leftTrials(1:7),MLD_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),MLD(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),MLD_c(channelTrials(1:7)),'mx');
axis([0 28 0 3]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:0.5:10,'YTickLabel',0:0.5:10,'FontName','Arial','FontSize',10);
ylabel('Max Lateral Deviation [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.2 0.2 0.5 0.6]); hold on;
plot(upTrials(8:59)-EXP(1)+1,MLD(upTrials(8:59)),'bo');
hold on
plot(upTrials(8:59)-EXP(1)+1,MLD_c(upTrials(8:59)),'bx');
hold on
plot(rightTrials(8:59)-EXP(1)+1,MLD(rightTrials(8:59)),'ro');
hold on
plot(rightTrials(8:59)-EXP(1)+1,MLD_c(rightTrials(8:59)),'rx');
hold on
plot(leftTrials(8:59)-EXP(1)+1,MLD(leftTrials(8:59)),'go');
hold on
plot(leftTrials(8:59)-EXP(1)+1,MLD_c(leftTrials(8:59)),'gx');
hold on
plot(channelTrials(8:59)-EXP(1)+1,MLD(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,MLD_c(channelTrials(8:59)),'mx');
axis([0 208 0 3]); set(gca,'LineWidth',2,'XTick',[29 81 133 185 236],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
plot(upTrials(60:66)-POST_EXP(1)+1,MLD(upTrials(60:66)),'bo');
hold on
plot(upTrials(60:66)-POST_EXP(1)+1,MLD_c(upTrials(60:66)),'bx');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,MLD(rightTrials(60:66)),'ro');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,MLD_c(rightTrials(60:66)),'rx');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,MLD(leftTrials(60:66)),'go');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,MLD_c(leftTrials(60:66)),'gx');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,MLD(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,MLD_c(channelTrials(60:66)),'mx');
axis([0 28 0 3]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['Post-Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title(['Post Exposure']);
%% RMSE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RMSE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Taken straight from kinsym2 step 2 files
rmse=zeros(numTrials,1); % allocate space for rmse
```

```
mov_int = zeros(numTrials,1);
for i=1:numTrials
   if wrong_trial(i)==0
      xx=cursorPosX{i,1}(onset(i):offset(i))*1000; % convert to mm
      yy=cursorPosY{i,1}(onset(i):offset(i))*1000;
      % spatial resampling of movement path
      N= 2000; N1= length(xx); % Computes equally-spaced vector assuming 1000 samples
      xc= 1/(N-1)*(0:N-1)*(xx(N1)-xx(1))+xx(1);
      yc= 1/(N-1)*(0:N-1)*(yy(N1)-yy(1))+yy(1);
      % integrates the movement length
      mov_int(i)=sum(sqrt(diff(xx).^2+ diff(yy).^2))
      di=(0:N-1)*mov_int(i)/(N-1);
      d=[0; (cumsum(sqrt((diff(xx).^2)+ (diff(yy).^2))))];
      % interpolates the movement path to make it equally spaced
      x2i= interp1q(d,xx,di');
      y2i= interp1q(d,yy,di');
      x2i(N)=xc(N);
      y2i(N)=yc(N);
      optimal =[xc', yc'];
      resampled_path =[x2i, y2i];
      rmse(i) = sqrt(sum(sum((resampled_path - optimal).^2))/N);
   else rmse(i)=NaN;
   end
end
%% rmse outlier analysis
data = outlier_t(rmse(upTrials(1:7))); % Outlier for baseline
rmse_c(upTrials(1:7)) = data;
data = outlier_t(rmse(rightTrials(1:7)));
rmse_c(rightTrials(1:7)) = data;
data = outlier_t(rmse(leftTrials(1:7))); % Outlier for  baseline
rmse_c(leftTrials(1:7)) = data;
data = outlier_t(rmse(channelTrials(1:7)));
rmse_c(channelTrials(1:7)) = data;
clear data;
% data = outlier_t(rmse(upTrials(14:23))); % Outlier for first 20 exposure
% rmse_c(upTrials(14:23)) = data;
% data = outlier_t(rmse(downTrials(14:23)));
% rmse_c(downTrials(14:23)) = data;
% clear data;
rmse_c(upTrials(8:11)) = rmse(upTrials(8:11));
rmse_c(rightTrials(8:11)) = rmse(rightTrials(8:11));
rmse_c(leftTrials(8:11)) = rmse(leftTrials(8:11));
rmse_c(channelTrials(8:11)) = rmse(channelTrials(8:11));
data = outlier_t(rmse(upTrials(12:59))); % Outlier for last 188 exposure
rmse_c(upTrials(12:59)) = data;
data = outlier_t(rmse(rightTrials(12:59)));
rmse_c(rightTrials(12:59)) = data;
data = outlier_t(rmse(leftTrials(12:59))); % Outlier for last 188 exposure
rmse_c(leftTrials(12:59)) = data;
data = outlier_t(rmse(channelTrials(12:59)));
rmse_c(channelTrials(12:59)) = data;
clear data;

% data = outlier_t(rmse(upTrials(43:45))); % Outlier for first 10 post-exp
% rmse_c(upTrials(43:45)) = data;
% data = outlier_t(rmse(downTrials(43:45)));
% rmse_c(downTrials(43:45)) = data;
% clear data;
rmse_c(upTrials(60:61)) = rmse(upTrials(60:61));
```

```
rmse_c(rightTrials(60:61)) = rmse(rightTrials(60:61));
rmse_c(leftTrials(60:61)) = rmse(leftTrials(60:61));
rmse_c(channelTrials(60:61)) = rmse(channelTrials(60:61));
data = outlier_t(rmse(upTrials(62:66))); % Outlier for last 18 post-exp
rmse_c(upTrials(62:66)) = data;
data = outlier_t(rmse(rightTrials(62:66)));
rmse_c(rightTrials(62:66)) = data;
data = outlier_t(rmse(leftTrials(62:66))); % Outlier for last 18 post-exp
rmse_c(leftTrials(62:66)) = data;
data = outlier_t(rmse(channelTrials(62:66)));
rmse_c(channelTrials(62:66)) = data;
clear data;
% transpose and calculate standardized variable
rmse_c = rmse_c';
bkup_mean = nanmean(rmse_c(upTrials(1:7)));
bkup_std = nanstd(rmse_c(upTrials(1:7)));
rmse_up_st = (rmse_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(rmse_c(rightTrials(1:7)));
bkright_std = nanstd(rmse_c(rightTrials(1:7)));
rmse_right_st = (rmse_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(rmse_c(leftTrials(1:7)));
bkleft_std = nanstd(rmse_c(leftTrials(1:7)));
rmse_left_st = (rmse_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(rmse_c(channelTrials(1:7)));
bkchannel_std = nanstd(rmse_c(channelTrials(1:7)));
rmse_channel_st = (rmse_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting Code for rmse
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),rmse(upTrials(1:7)),'bo');
hold on
plot(upTrials(1:7),rmse_c(upTrials(1:7)),'bx');
hold on
plot(rightTrials(1:7),rmse(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),rmse_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),rmse(leftTrials(1:7)),'go');
hold on
plot(leftTrials(1:7),rmse_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),rmse(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),rmse_c(channelTrials(1:7)),'mx');
axis([0 28 0 25]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:5:25,'YTickLabel', 0:5:25,'FontName','Arial','FontSize',10);
ylabel('rmse [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.2 0.2 0.5 0.6]); hold on;
plot(upTrials(8:59)-EXP(1)+1,rmse(upTrials(8:59)),'bo');
hold on
plot(upTrials(8:59)-EXP(1)+1,rmse_c(upTrials(8:59)),'bx');
hold on
plot(rightTrials(8:59)-EXP(1)+1,rmse(rightTrials(8:59)),'ro');
hold on
```

```
plot(rightTrials(8:59)-EXP(1)+1,rmse_c(rightTrials(8:59)),'rx');
hold on
plot(leftTrials(8:59)-EXP(1)+1,rmse(leftTrials(8:59)),'go');
hold on
plot(leftTrials(8:59)-EXP(1)+1,rmse_c(leftTrials(8:59)),'gx');
hold on
plot(channelTrials(8:59)-EXP(1)+1,rmse(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,rmse_c(channelTrials(8:59)),'mx');
axis([29 208 0 25]); set(gca,'LineWidth',2,'XTick',[29 82 135 186 236],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
plot(upTrials(60:66)-POST_EXP(1)+1,rmse(upTrials(60:66)),'bo');
hold on
plot(upTrials(60:66)-POST_EXP(1)+1,rmse_c(upTrials(60:66)),'bx');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,rmse(rightTrials(60:66)),'ro');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,rmse_c(rightTrials(60:66)),'rx');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,rmse(leftTrials(60:66)),'go');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,rmse_c(leftTrials(60:66)),'gx');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,rmse(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,rmse_c(channelTrials(60:66)),'mx');
axis([0 28 0 25]); set(gca,'LineWidth',2,'XTick',[237 250 264],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['Post-
Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title(['Post Exposure'])
%%%%%%%%%%%%%%%%%%%%% Movement Length %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mov_int = zeros(numTrials,1);
for i = 1:numTrials
  if wrong_trial(i) == 0
    mov_int(i) = sum(sqrt(diff(cursorPosX{i,1}(onset(i):offset(i))).^2 + diff(cursorPosY{i,1}(onset(i):offset(i))).^2)) * 100;
%movement length in cm
  else
    mov_int(i) = NaN;
  end
end
%% mov_int outlier analysis
data = outlier_t(mov_int(upTrials(1:7))); % Outlier for baseline
mov_int_c(upTrials(1:7)) = data;
data = outlier_t(mov_int(rightTrials(1:7)));
mov_int_c(rightTrials(1:7)) = data;
data = outlier_t(mov_int(leftTrials(1:7))); % Outlier for  baseline
mov_int_c(leftTrials(1:7)) = data;
data = outlier_t(mov_int(channelTrials(1:7)));
mov_int_c(channelTrials(1:7)) = data;
clear data;
% data = outlier_t(mov_int(upTrials(14:23))); % Outlier for first 20 exposure
% mov_int_c(upTrials(14:23)) = data;
% data = outlier_t(mov_int(downTrials(14:23)));
% mov_int_c(downTrials(14:23)) = data;
% clear data;
mov_int_c(upTrials(8:11)) = mov_int(upTrials(8:11));
mov_int_c(rightTrials(8:11)) = mov_int(rightTrials(8:11));
```

```
mov_int_c(leftTrials(8:11)) = mov_int(leftTrials(8:11));
mov_int_c(channelTrials(8:11)) = mov_int(channelTrials(8:11));
data = outlier_t(mov_int(upTrials(12:59))); % Outlier for last 188 exposure
mov_int_c(upTrials(12:59)) = data;
data = outlier_t(mov_int(rightTrials(12:59)));
mov_int_c(rightTrials(12:59)) = data;
data = outlier_t(mov_int(leftTrials(12:59))); % Outlier for last 188 exposure
mov_int_c(leftTrials(12:59)) = data;
data = outlier_t(mov_int(channelTrials(12:59)));
mov_int_c(channelTrials(12:59)) = data;
clear data;
% data = outlier_t(mov_int(upTrials(43:45))); % Outlier for first 10 post-exp
% mov_int_c(upTrials(43:45)) = data;
% data = outlier_t(mov_int(downTrials(43:45)));
% mov_int_c(downTrials(43:45)) = data;
% data = outlier_t(mov_int(leftTrials(43:45))); % Outlier for first 8 post-exp
% mov_int_c(leftTrials(43:45)) = data;
% data = outlier_t(mov_int(rightTrials(43:45)));
% mov_int_c(rightTrials(43:45)) = data;
% clear data;
mov_int_c(upTrials(60:61)) = mov_int(upTrials(60:61));
mov_int_c(rightTrials(60:61)) = mov_int(rightTrials(60:61));
mov_int_c(leftTrials(60:61)) = mov_int(leftTrials(60:61));
mov_int_c(channelTrials(60:61)) = mov_int(channelTrials(60:61));

data = outlier_t(mov_int(upTrials(62:66))); % Outlier for last 20 post-exp
mov_int_c(upTrials(62:66)) = data;
data = outlier_t(mov_int(rightTrials(62:66)));
mov_int_c(rightTrials(62:66)) = data;
data = outlier_t(mov_int(leftTrials(62:66))); % Outlier for last 20 post-exp
mov_int_c(leftTrials(62:66)) = data;
data = outlier_t(mov_int(channelTrials(62:66)));
mov_int_c(channelTrials(62:66)) = data;
clear data;
% transpose and calculate standardized variable
mov_int_c = mov_int_c';
bkup_mean = nanmean(mov_int_c(upTrials(1:7)));
bkup_std = nanstd(mov_int_c(upTrials(1:7)));
mov_int_up_st = (mov_int_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(mov_int_c(rightTrials(1:7)));
bkright_std = nanstd(mov_int_c(rightTrials(1:7)));
mov_int_right_st = (mov_int_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(mov_int_c(leftTrials(1:7)));
bkleft_std = nanstd(mov_int_c(leftTrials(1:7)));
mov_int_left_st = (mov_int_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(mov_int_c(channelTrials(1:7)));
bkchannel_std = nanstd(mov_int_c(channelTrials(1:7)));
mov_int_channel_st = (mov_int_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting Code for mov_int
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),mov_int(upTrials(1:7)),'bo');
hold on
plot(upTrials(1:7),mov_int_c(upTrials(1:7)),'bx');
hold on
```

```
plot(rightTrials(1:7),mov_int(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),mov_int_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),mov_int(leftTrials(1:7)),'go');
hold on
plot(leftTrials(1:7),mov_int_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),mov_int(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),mov_int_c(channelTrials(1:7)),'mx');
axis([0 28 8 15]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:4:20,'YTickLabel', 0:4:20,'FontName','Arial','FontSize',10);
ylabel('Movement Length [s]'); title('Baseline','fontsize',11);

hold on
subplot('Position',[0.2 0.2 0.5 0.6]); hold on;
plot(upTrials(8:59)-EXP(1)+1,mov_int(upTrials(8:59)),'bo');
hold on
plot(upTrials(8:59)-EXP(1)+1,mov_int_c(upTrials(8:59)),'bx');
hold on
plot(rightTrials(8:59)-EXP(1)+1,mov_int(rightTrials(8:59)),'ro');
hold on
plot(rightTrials(8:59)-EXP(1)+1,mov_int_c(rightTrials(8:59)),'rx');
hold on
plot(leftTrials(8:59)-EXP(1)+1,mov_int(leftTrials(8:59)),'go');
hold on
plot(leftTrials(8:59)-EXP(1)+1,mov_int_c(leftTrials(8:59)),'gx');
hold on
plot(channelTrials(8:59)-EXP(1)+1,mov_int(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,mov_int_c(channelTrials(8:59)),'mx');
axis([0 208 8 15]); set(gca,'LineWidth',2,'XTick',[1 52 105 157 208],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
plot(upTrials(60:66)-POST_EXP(1)+1,mov_int(upTrials(60:66)),'bo');
hold on
plot(upTrials(60:66)-POST_EXP(1)+1,mov_int_c(upTrials(60:66)),'bx');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,mov_int(rightTrials(60:66)),'ro');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,mov_int_c(rightTrials(60:66)),'rx');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,mov_int(leftTrials(60:66)),'go');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,mov_int_c(leftTrials(60:66)),'gx');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,mov_int(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,mov_int_c(channelTrials(60:66)),'mx');
axis([0 28 8 15]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['Post-
Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title(['Post Exposure']);
%% Normalized Jerk Score
%%%%%%%%%%%%%%%%%%%%%% Normalized Jerk %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
acc_tan = cell(numTrials,1);
jerk = cell(numTrials,1);
jerk_square = cell(numTrials,1);
delta_1 = cell(numTrials,1);
```

```matlab
jerk_int = zeros(numTrials,1);
norm_jerk = zeros(numTrials,1);
for i = 1:numTrials
   if wrong_trial(i) == 0
      acc_tan{i,1} = 100*sqrt((sortData(i).Right_HandXAcc).^2 + (sortData(i).Right_HandYAcc).^2); % in cm/s/s
      jerk{i,1} = diff(acc_tan{i,1})/delta_t;
      jerk_square{i,1} = jerk{i,1}.^2;
      delta_1{i,1} = (0:1:(length(jerk_square{i,1}) - 1)) ./fs;
      jerk_int(i) = trapz(delta_1{i,1},jerk_square{i,1});
      norm_jerk(i) = sqrt(0.5 *jerk_int(i) * ((MT(i))^5)/ (mov_int(i)^2));
   else
      norm_jerk(i) = NaN;
   end
end
%% norm_jerk outlier analysis
data = outlier_t(norm_jerk(upTrials(1:7))); % Outlier for visual baseline
norm_jerk_c(upTrials(1:7)) = data;
data = outlier_t(norm_jerk(rightTrials(1:7)));
norm_jerk_c(rightTrials(1:7)) = data;
data = outlier_t(norm_jerk(leftTrials(1:7))); % Outlier for visual baseline
norm_jerk_c(leftTrials(1:7)) = data;
data = outlier_t(norm_jerk(channelTrials(1:7)));
norm_jerk_c(channelTrials(1:7)) = data;
clear data;
% data = outlier_t(norm_jerk(upTrials(14:23))); % Outlier for first 20 exposure
% norm_jerk_c(upTrials(14:23)) = data;
% data = outlier_t(norm_jerk(downTrials(14:23)));
% norm_jerk_c(downTrials(14:23)) = data;
% clear data;
norm_jerk_c(upTrials(8:11)) = norm_jerk(upTrials(8:11));
norm_jerk_c(rightTrials(8:11)) = norm_jerk(rightTrials(8:11));
norm_jerk_c(leftTrials(8:11)) = norm_jerk(leftTrials(8:11));
norm_jerk_c(channelTrials(8:11)) = norm_jerk(channelTrials(8:11));
data = outlier_t(norm_jerk(upTrials(12:59))); % Outlier for last 188 exposure
norm_jerk_c(upTrials(12:59)) = data;
data = outlier_t(norm_jerk(rightTrials(12:59)));
norm_jerk_c(rightTrials(12:59)) = data;
data = outlier_t(norm_jerk(leftTrials(12:59))); % Outlier for last 188 exposure
norm_jerk_c(leftTrials(12:59)) = data;
data = outlier_t(norm_jerk(channelTrials(12:59)));
norm_jerk_c(channelTrials(12:59)) = data;
clear data;
% data = outlier_t(norm_jerk(upTrials(43:45))); % Outlier for first 8 post-exp
% norm_jerk_c(upTrials(43:45)) = data;
% data = outlier_t(norm_jerk(downTrials(43:45)));
% norm_jerk_c(downTrials(43:45)) = data;
% data = outlier_t(norm_jerk(leftTrials(43:45))); % Outlier for first 8 post-exp
% norm_jerk_c(leftTrials(43:45)) = data;
% data = outlier_t(norm_jerk(rightTrials(43:45)));
% norm_jerk_c(rightTrials(43:45)) = data;
% clear data;
norm_jerk_c(upTrials(60:61)) = norm_jerk(upTrials(60:61));
norm_jerk_c(rightTrials(60:61)) = norm_jerk(rightTrials(60:61));
norm_jerk_c(leftTrials(60:61)) = norm_jerk(leftTrials(60:61));
norm_jerk_c(channelTrials(60:61)) = norm_jerk(channelTrials(60:61));
data = outlier_t(norm_jerk(upTrials(62:66))); % Outlier for last 20 post-exp
norm_jerk_c(upTrials(62:66)) = data;
data = outlier_t(norm_jerk(rightTrials(62:66)));
```

```
norm_jerk_c(rightTrials(62:66)) = data;
data = outlier_t(norm_jerk(leftTrials(62:66))); % Outlier for last 20 post-exp
norm_jerk_c(leftTrials(62:66)) = data;
data = outlier_t(norm_jerk(channelTrials(62:66)));
norm_jerk_c(channelTrials(62:66)) = data;
clear data;
% transpose and calculate standardized variable
norm_jerk_c = norm_jerk_c';
bkup_mean = nanmean(norm_jerk_c(upTrials(1:7)));
bkup_std = nanstd(norm_jerk_c(upTrials(1:7)));
norm_jerk_up_st = (norm_jerk_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(norm_jerk_c(rightTrials(1:7)));
bkright_std = nanstd(norm_jerk_c(rightTrials(1:7)));
norm_jerk_right_st = (norm_jerk_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(norm_jerk_c(leftTrials(1:7)));
bkleft_std = nanstd(norm_jerk_c(leftTrials(1:7)));
norm_jerk_left_st = (norm_jerk_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(norm_jerk_c(channelTrials(1:7)));
bkchannel_std = nanstd(norm_jerk_c(channelTrials(1:7)));
norm_jerk_channel_st = (norm_jerk_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting Code for norm_jerk
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),norm_jerk(upTrials(1:7)),'bo');
hold on
plot(upTrials(1:7),norm_jerk_c(upTrials(1:7)),'bx');
hold on
plot(rightTrials(1:7),norm_jerk(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),norm_jerk_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),norm_jerk(leftTrials(1:7)),'go');
hold on
plot(leftTrials(1:7),norm_jerk_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),norm_jerk(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),norm_jerk_c(channelTrials(1:7)),'mx');
axis([0 28 0 500]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:50:500,'YTickLabel', 0:50:500,'FontName','Arial','FontSize',10);
ylabel('norm jerk [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.2 0.2 0.5 0.6]); hold on;
plot(upTrials(8:59)-EXP(1)+1,norm_jerk(upTrials(8:59)),'bo');
hold on
plot(upTrials(8:59)-EXP(1)+1,norm_jerk_c(upTrials(8:59)),'bx');
hold on
plot(rightTrials(8:59)-EXP(1)+1,norm_jerk(rightTrials(8:59)),'ro');
hold on
plot(rightTrials(8:59)-EXP(1)+1,norm_jerk_c(rightTrials(8:59)),'rx');
hold on
plot(leftTrials(8:59)-EXP(1)+1,norm_jerk(leftTrials(8:59)),'go');
hold on
plot(leftTrials(8:59)-EXP(1)+1,norm_jerk_c(leftTrials(8:59)),'gx');
hold on
```

```
plot(channelTrials(8:59)-EXP(1)+1,norm_jerk(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,norm_jerk_c(channelTrials(8:59)),'mx');
axis([0 208 0 500]); set(gca,'LineWidth',2,'XTick', [1 53 105 157 208],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
plot(upTrials(60:66)-POST_EXP(1)+1,norm_jerk(upTrials(60:66)),'bo');
hold on
plot(upTrials(60:66)-POST_EXP(1)+1,norm_jerk_c(upTrials(60:66)),'bx');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,norm_jerk(rightTrials(60:66)),'ro');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,norm_jerk_c(rightTrials(60:66)),'rx');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,norm_jerk(leftTrials(60:66)),'go');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,norm_jerk_c(leftTrials(60:66)),'gx');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,norm_jerk(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,norm_jerk_c(channelTrials(60:66)),'mx');
axis([0 28 0 500]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['post-
exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title(['Post Exposure'])
%% Peak Velocity
velPeak = velPeak * 100; % Convert to cm/s
%% Peak Velocity outlier calc
data = outlier_t(velPeak(upTrials(1:7))); % Outlier for visual baseline
velPeak_c(upTrials(1:7)) = data;
data = outlier_t(velPeak(rightTrials(1:7)));
velPeak_c(rightTrials(1:7)) = data;
data = outlier_t(velPeak(leftTrials(1:7))); % Outlier for visual baseline
velPeak_c(leftTrials(1:7)) = data;
data = outlier_t(velPeak(channelTrials(1:7)));
velPeak_c(channelTrials(1:7)) = data;
clear data;
% data = outlier_t(velPeak(upTrials(14:23))); % Outlier for first 20 exposure
% velPeak_c(upTrials(14:23)) = data;
% data = outlier_t(velPeak(downTrials(14:23)));
% velPeak_c(downTrials(14:23)) = data;
% clear data;
velPeak_c(upTrials(8:11)) = velPeak(upTrials(8:11));
velPeak_c(rightTrials(8:11)) = velPeak(rightTrials(8:11));
velPeak_c(leftTrials(8:11)) = velPeak(leftTrials(8:11));
velPeak_c(channelTrials(8:11)) = velPeak(channelTrials(8:11));
data = outlier_t(velPeak(upTrials(12:59))); % Outlier for last 100 exposure
velPeak_c(upTrials(12:59)) = data;
data = outlier_t(velPeak(rightTrials(12:59)));
velPeak_c(rightTrials(12:59)) = data;
data = outlier_t(velPeak(leftTrials(12:59))); % Outlier for last 100 exposure
velPeak_c(leftTrials(12:59)) = data;
data = outlier_t(velPeak(channelTrials(12:59)));
velPeak_c(channelTrials(12:59)) = data;
clear data;
% data = outlier_t(velPeak(upTrials(43:45))); % Outlier for first 10 post-exp
% velPeak_c(upTrials(43:45)) = data;
% data = outlier_t(velPeak(downTrials(43:45)));
```

```
% velPeak_c(downTrials(43:45)) = data;
% clear data;
velPeak_c(upTrials(60:61)) = velPeak(upTrials(60:61));
velPeak_c(rightTrials(60:61)) = velPeak(rightTrials(60:61));
velPeak_c(leftTrials(60:61)) = velPeak(leftTrials(60:61));
velPeak_c(channelTrials(60:61)) = velPeak(channelTrials(60:61));
data = outlier_t(velPeak(upTrials(62:66))); % Outlier for last 10 post-exp
velPeak_c(upTrials(62:66)) = data;
data = outlier_t(velPeak(rightTrials(62:66)));
velPeak_c(rightTrials(62:66)) = data;
data = outlier_t(velPeak(leftTrials(62:66))); % Outlier for last 10 post-exp
velPeak_c(leftTrials(62:66)) = data;
data = outlier_t(velPeak(channelTrials(62:66)));
velPeak_c(channelTrials(62:66)) = data;
clear data;
% transpose and calculate standardized variable
velPeak_c = velPeak_c';
bkup_mean = nanmean(velPeak_c(upTrials(1:7)));
bkup_std = nanstd(velPeak_c(upTrials(1:7)));
velPeak_up_st = (velPeak_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(velPeak_c(rightTrials(1:7)));
bkright_std = nanstd(velPeak_c(rightTrials(1:7)));
velPeak_right_st = (velPeak_c(rightTrials) - bkright_mean)/bkright_std;

bkleft_mean = nanmean(velPeak_c(leftTrials(1:7)));
bkleft_std = nanstd(velPeak_c(leftTrials(1:7)));
velPeak_left_st = (velPeak_c(leftTrials) - bkleft_mean)/bkleft_std;

bkchannel_mean = nanmean(velPeak_c(channelTrials(1:7)));
bkchannel_std = nanstd(velPeak_c(channelTrials(1:7)));
velPeak_channel_st = (velPeak_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting Code for velPeak
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),velPeak(upTrials(1:7)),'bo');
hold on
plot(upTrials(1:7),velPeak_c(upTrials(1:7)),'bx');
hold on
plot(rightTrials(1:7),velPeak(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),velPeak_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),velPeak(leftTrials(1:7)),'go');
hold on
plot(leftTrials(1:7),velPeak_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),velPeak(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),velPeak_c(channelTrials(1:7)),'mx');
axis([0 28 0 100]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:10:100,'YTickLabel', 0:10:100,'FontName','Arial','FontSize',10);
ylabel('velPeak [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.2 0.2 0.5 0.6]); hold on;
plot(upTrials(8:59)-EXP(1)+1,velPeak(upTrials(8:59)),'bo');
hold on
```

```
plot(upTrials(8:59)-EXP(1)+1,velPeak_c(upTrials(8:59)),'bx');
hold on
plot(rightTrials(8:59)-EXP(1)+1,velPeak(rightTrials(8:59)),'ro');
hold on
plot(rightTrials(8:59)-EXP(1)+1,velPeak_c(rightTrials(8:59)),'rx');
hold on
plot(leftTrials(8:59)-EXP(1)+1,velPeak(leftTrials(8:59)),'go');
hold on
plot(leftTrials(8:59)-EXP(1)+1,velPeak_c(leftTrials(8:59)),'gx');
hold on
plot(channelTrials(8:59)-EXP(1)+1,velPeak(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,velPeak_c(channelTrials(8:59)),'mx');
axis([0 208 0 100]); set(gca,'LineWidth',2,'XTick', [1 52 105 157 208],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
plot(upTrials(60:66)-POST_EXP(1)+1,velPeak(upTrials(60:66)),'bo');
hold on
plot(upTrials(60:66)-POST_EXP(1)+1,velPeak_c(upTrials(60:66)),'bx');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,velPeak(rightTrials(60:66)),'ro');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,velPeak_c(rightTrials(60:66)),'rx');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,velPeak(leftTrials(60:66)),'go');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,velPeak_c(leftTrials(60:66)),'gx');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,velPeak(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,velPeak_c(channelTrials(60:66)),'mx');
axis([0 28 0 100]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['Post-
Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
%% EPE, EP_X, and EP_Y calcs
%%%%%%%%%%%%%%%%%%%%%% End-Point Error (EPE)%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% and EP_X and EP_Y
EPE = zeros(numTrials,1);
EP_X = zeros(numTrials,1);
EP_Y = zeros(numTrials,1);
for i = 1:numTrials
   if wrong_trial(i) == 0
      if upBool(i) == 1 || channelBool(i) == 1
         EP_X(i) = (cursorPosX{i,1}(offset(i))*100 - Tx) - sortData(i).TARGET_TABLE.X(2);
      elseif leftBool(i) == 1
         EP_X(i) = (cursorPosX{i,1}(offset(i))*100 - Tx) - sortData(i).TARGET_TABLE.X(4);
      elseif rightBool(i) == 1
         EP_X(i) = (cursorPosX{i,1}(offset(i))*100 - Tx) - sortData(i).TARGET_TABLE.X(3);
      end
%       if leftBool(i) == 1 || rightBool(i) == 1
%          EP_Y(i) = (cursorPosY{i,1}(offset(i))*100 - Ty) - sortData(i).TARGET_TABLE.Y(6);
%       elseif upBool(i) == 1
%          EP_Y(i) = (cursorPosY{i,1}(offset(i))*100 - Ty) - sortData(i).TARGET_TABLE.Y(3);
%       elseif rightBool(i) == 1
%          EP_Y(i) = (cursorPosY{i,1}(offset(i))*100 - Ty) - sortData(i).TARGET_TABLE.Y(4);
%       end
   else
      EPE(i) = NaN;
```

```
        EP_X(i) = NaN;
        EP_Y(i) = NaN;
    end
end
EPE = sqrt(EP_X.^2 + EP_Y.^2);
%% EPE outlier analysis
data = outlier_t(EPE(upTrials(1:7))); % Outlier for visual baseline
EPE_c(upTrials(1:7)) = data;
data = outlier_t(EPE(rightTrials(1:7)));
EPE_c(rightTrials(1:7)) = data;
data = outlier_t(EPE(leftTrials(1:7))); % Outlier for visual baseline
EPE_c(leftTrials(1:7)) = data;
data = outlier_t(EPE(channelTrials(1:7)));
EPE_c(channelTrials(1:7)) = data;
clear data;
% data = outlier_t(EPE(upTrials(14:23))); % Outlier for first 20 exposure
% EPE_c(upTrials(14:23)) = data;
% data = outlier_t(EPE(downTrials(14:23)));
% EPE_c(downTrials(14:23)) = data;
% clear data;
EPE_c(upTrials(8:11)) = EPE(upTrials(8:11));
EPE_c(rightTrials(8:11)) = EPE(rightTrials(8:11));
EPE_c(leftTrials(8:11)) = EPE(leftTrials(8:11));
EPE_c(channelTrials(8:11)) = EPE(channelTrials(8:11));
data = outlier_t(EPE(upTrials(12:59))); % Outlier for last 100 exposure
EPE_c(upTrials(12:59)) = data;
data = outlier_t(EPE(rightTrials(12:59)));
EPE_c(rightTrials(12:59)) = data;
data = outlier_t(EPE(leftTrials(12:59))); % Outlier for last 100 exposure
EPE_c(leftTrials(12:59)) = data;
data = outlier_t(EPE(channelTrials(12:59)));
EPE_c(channelTrials(12:59)) = data;
clear data;
% data = outlier_t(EPE(upTrials(157:161))); % Outlier for first 10 post-exp
% EPE_c(upTrials(157:161)) = data;
% data = outlier_t(EPE(downTrials(157:161)));
% EPE_c(downTrials(157:161)) = data;
% clear data;
EPE_c(upTrials(60:61)) = EPE(upTrials(60:61));
EPE_c(rightTrials(60:61)) = EPE(rightTrials(60:61));
EPE_c(leftTrials(60:61)) = EPE(leftTrials(60:61));
EPE_c(channelTrials(60:61)) = EPE(channelTrials(60:61));
data = outlier_t(EPE(upTrials(62:66))); % Outlier for last 10 post-exp
EPE_c(upTrials(62:66)) = data;
data = outlier_t(EPE(rightTrials(62:66)));
EPE_c(rightTrials(62:66)) = data;
data = outlier_t(EPE(leftTrials(62:66))); % Outlier for last 10 post-exp
EPE_c(leftTrials(62:66)) = data;
data = outlier_t(EPE(channelTrials(62:66)));
EPE_c(channelTrials(62:66)) = data;
clear data;
% transpose and calculate standardized variable
EPE_c = EPE_c';
bkup_mean = nanmean(EPE_c(upTrials(1:7)));
bkup_std = nanstd(EPE_c(upTrials(1:7)));
EPE_up_st = (EPE_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(EPE_c(rightTrials(1:7)));
bkright_std = nanstd(EPE_c(rightTrials(1:7)));
```

```
EPE_right_st = (EPE_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(EPE_c(leftTrials(1:7)));
bkleft_std = nanstd(EPE_c(leftTrials(1:7)));
EPE_left_st = (EPE_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(EPE_c(channelTrials(1:7)));
bkchannel_std = nanstd(EPE_c(channelTrials(1:7)));
EPE_channel_st = (EPE_c(channelTrials) - bkright_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting Code for EPE
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),EPE(upTrials(1:7)),'bo');
hold on
plot(upTrials(1:7),EPE_c(upTrials(1:7)),'bx');
hold on
plot(rightTrials(1:7),EPE(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),EPE_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),EPE(leftTrials(1:7)),'go');
hold on
plot(leftTrials(1:7),EPE_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),EPE(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),EPE_c(channelTrials(1:7)),'mx');
axis([0 28 0 5]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:4:20,'YTickLabel', 0:4:20,'FontName','Arial','FontSize',10);
ylabel('EPE [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.32 0.2 0.4 0.6]); hold on;
plot(upTrials(8:59)-EXP(1)+1,EPE(upTrials(8:59)),'bo');
hold on
plot(upTrials(8:59)-EXP(1)+1,EPE_c(upTrials(8:59)),'bx');
hold on
plot(rightTrials(8:59)-EXP(1)+1,EPE(rightTrials(8:59)),'ro');
hold on
plot(rightTrials(8:59)-EXP(1)+1,EPE_c(rightTrials(8:59)),'rx');
hold on
plot(leftTrials(8:59)-EXP(1)+1,EPE(leftTrials(8:59)),'go');
hold on
plot(leftTrials(8:59)-EXP(1)+1,EPE_c(leftTrials(8:59)),'gx');
hold on
plot(channelTrials(8:59)-EXP(1)+1,EPE(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,EPE_c(channelTrials(8:59)),'mx');
axis([0 208 0 5]); set(gca,'LineWidth',2,'XTick',[1 52 105 157 208],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
plot(upTrials(60:66)-POST_EXP(1)+1,EPE(upTrials(60:66)),'bo');
hold on
plot(upTrials(60:66)-POST_EXP(1)+1,EPE_c(upTrials(60:66)),'bx');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,EPE(rightTrials(60:66)),'ro');
hold on
```

```
plot(rightTrials(60:66)-POST_EXP(1)+1,EPE_c(rightTrials(60:66)),'rx');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,EPE(leftTrials(60:66)),'go');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,EPE_c(leftTrials(60:66)),'gx');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,EPE(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,EPE_c(channelTrials(60:66)),'mx');
axis([0 28 0 5]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['Post-Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
%% Initial Endpoint Error
% This is defined as the distance to from the target at the first period of
% acceleration following a period of deceleration, or a change in the sign
% of velocity.  Thus, the initial movement has "ended" when there is either
% a change in movement direction or when an additional propulsive action is
% made (Seidler, 2006, Brain Research Bulletin)
RhX = cell(numTrials,1);
RhY = cell(numTrials,1);
mov_tanL = cell(numTrials,1);
velR = cell(numTrials,1);
for i = 1: numTrials
   RhX{i,1} = sortData(i,1).Right_HandX; %(1:end/2);
   RhY{i,1} = sortData(i,1).Right_HandY; %(1:end/2);
   mov_tanL{i,1} = sqrt(RhX{i,1}.^2 + RhY{i,1}.^2);
   velR{i,1} = diff(mov_tanL{i,1})/delta_t;
end
ind_localPeak = zeros(numTrials,1);
ind_localPeak_bool = cell(numTrials,1);
ind_IEE = zeros(numTrials,1);
vel_thresh = cell(numTrials,1);
velPeak2 = zeros(numTrials,1);
vel_thresh_bool = cell(numTrials,1);
ind_velThresh = zeros(numTrials,1);
for i = 1:numTrials
   if wrong_trial(i) == 0
      %finds first local min for up/left trials and local max for
      %down/right trials after PeakVel - AKA, period of acceleration
      %following a period of deceleration
      velPeak2(i) = max(vel{i,1}(1:offset(i)));
      if velPeak2(i) > 0
         ind_localPeak_bool{i} = islocalmin(vel{i}(indPeak(i):end,1));
      elseif velPeak2(i) < 0
         ind_localPeak_bool{i} = islocalmax(vel{i}(indPeak(i):end,1));
      else
         ind_localPeak_bool{i} = 'NaN';
      end
      ind_localPeak_temp = find(ind_localPeak_bool{i},1,'first');
      ind_localPeak(i) = indPeak(i) + ind_localPeak_temp(1);         %this is sort of working

      %        %finds where velocity crosses zero after velPeak
      %        for j = indPeak:length(velR{i})-1
      %           if velR{i}(j,1) > 0 && velR{i}(j+1,1) > 0 %positive to positive, no change
      %              vel_thresh_bool{i}(j) = 0;
      %           elseif velR{i}(j,1) < 0 && velR{i}(j+1,1) < 0 %negative to negative, no change
      %              vel_thresh_bool{i}(j) = 0;
      %           elseif velR{i}(j,1) < 0 && velR{i}(j+1,1) > 0 %negative to positive, change!
      %              vel_thresh_bool{i}(j) = 1;
      %           elseif velR{i}(j,1) > 0 && velR{i}(j+1,1) < 0 %positive to negaive, change!
```

```
    %             vel_thresh_bool{i}(j) = 1;
    %           else
    %             vel_thresh_bool{i}(j) = 'NaN';
    %           end
    %         end
    for j = indPeak:length(vel{i})-1
      if vel{i}(j,1) < 0.001
        vel_thresh_bool{i}(j) = 1;
      elseif vel{i}(j,1) > 0.001
        vel_thresh_bool{i}(j) = 0;
      else
        vel_thresh_bool{i}(j) = 'NaN';
      end
    end

    if isempty(find(vel_thresh_bool{i},1,'first'))
      ind_velThresh(i) = 1000000;
    else
      ind_velThresh(i) = find(vel_thresh_bool{i},1,'first');
    end
    %assign ind_IEE to whichever comes first after indPeak
    if ind_localPeak(i) > ind_velThresh(i)
      ind_IEE(i) = ind_velThresh(i);
    elseif ind_localPeak(i) < ind_velThresh(i)
      ind_IEE(i) = ind_localPeak(i);
    else
      ind_IEE(i) = 'NaN';
    end
  end
end
%manually check IEE algorithm success
for i = 1:numTrials
  if wrong_trial(i) == 0
    flag = 2;
    while flag>1
      figure('Position', [100 100 1920/2 1080/2]); %[bottom left corner coords X and Y, W, H]
      plot(vel{i})
      hold on
      plot(indPeak(i), vel{i}(indPeak(i),1), 'bo')
      hold on
      if ind_IEE(i) == ind_velThresh(i)
        plot(ind_IEE(i), vel{i}(ind_IEE(i),1), 'ro') %red if due to crossing 0
      else
        plot(ind_IEE(i), vel{i}(ind_IEE(i),1), 'go') %green if due to local min
      end
      % Verify Onset/Offset
      button = questdlg('Confirm accel at zero(ish)','Acc threshold markers:','Yes','No','Reject','No');
      if strcmp(button,'Yes')
        close(1);
        flag=1;
      elseif strcmp(button,'No') %user can verify if movement onset was computed correctly
        [loc_acc_thresh,loc_size]=ginput(1);
        ind_IEE(i,1)=round(loc_acc_thresh(1)); % replace accel threshold with user defined input
        close(1);
        flag=2;
      elseif strcmp(button,'Reject')
        ind_IEE(i,1) = NaN;
        wrong_trial(i,1)=1;       % Keeps track of rejected trials
        close(1);
```

```
            flag=1;
          end
        end
        clear button;
    end
end
%%
%calculate IEE - distance to hand from target at ind_IEE
IEE_X = zeros(numTrials,1);
IEE_Y = zeros(numTrials,1);
IEE = zeros(numTrials,1);
for i = 1:numTrials
    if wrong_trial(i) == 0
        if upBool(i) == 1 || channelBool(i) == 1
            IEE_X(i) = (cursorPosX{i,1}(ind_IEE(i))*100 - Tx) - sortData(i).TARGET_TABLE.X(2);
        elseif leftBool(i) == 1
            IEE_X(i) = (cursorPosX{i,1}(ind_IEE(i))*100 - Tx) - sortData(i).TARGET_TABLE.X(4);
        elseif rightBool(i) == 1
            IEE_X(i) = (cursorPosX{i,1}(ind_IEE(i))*100 - Tx) - sortData(i).TARGET_TABLE.X(3);
        end
%         if leftBool(i) == 1 || rightBool(i) == 1
%             IEE_Y(i) = (cursorPosY{i,1}(ind_IEE(i))*100 - Ty) - sortData(i).TARGET_TABLE.Y(6);
%         elseif upBool(i) == 1
%             IEE_Y(i) = (cursorPosY{i,1}(ind_IEE(i))*100 - Ty) - sortData(i).TARGET_TABLE.Y(3);
%         elseif rightBool(i) == 1
%             IEE_Y(i) = (cursorPosY{i,1}(ind_IEE(i))*100 - Ty) - sortData(i).TARGET_TABLE.Y(4);
%         end
    else
        IEE(i) = NaN;
        IEE_X(i) = NaN;
        IEE_Y(i) = NaN;
    end
end
IEE = sqrt(IEE_X.^2 + IEE_Y.^2);
%% IEE outlier analysis
data = outlier_t(IEE(upTrials(1:7))); % Outlier for visual baseline
IEE_c(upTrials(1:7)) = data;
data = outlier_t(IEE(rightTrials(1:7)));
IEE_c(rightTrials(1:7)) = data;
data = outlier_t(IEE(leftTrials(1:7))); % Outlier for visual baseline
IEE_c(leftTrials(1:7)) = data;
data = outlier_t(IEE(channelTrials(1:7)));
IEE_c(channelTrials(1:7)) = data;
clear data;
% data = outlier_t(IEE(upTrials(14:23))); % Outlier for first 20 exposure
% IEE_c(upTrials(14:23)) = data;
% data = outlier_t(IEE(downTrials(14:23)));
% IEE_c(downTrials(14:23)) = data;
% clear data;
IEE_c(upTrials(8:11)) = IEE(upTrials(8:11));
IEE_c(rightTrials(8:11)) = IEE(rightTrials(8:11));
IEE_c(leftTrials(8:11)) = IEE(leftTrials(8:11));
IEE_c(channelTrials(8:11)) = IEE(channelTrials(8:11));

data = outlier_t(IEE(upTrials(12:59))); % Outlier for last 100 exposure
IEE_c(upTrials(12:59)) = data;
data = outlier_t(IEE(rightTrials(12:59)));
IEE_c(rightTrials(12:59)) = data;
data = outlier_t(IEE(leftTrials(12:59))); % Outlier for last 100 exposure
```

```
IEE_c(leftTrials(12:59)) = data;
data = outlier_t(IEE(channelTrials(12:59)));
IEE_c(channelTrials(12:59)) = data;
clear data;
% data = outlier_t(IEE(upTrials(157:161))); % Outlier for first 10 post-exp
% IEE_c(upTrials(157:161)) = data;
% data = outlier_t(IEE(downTrials(157:161)));
% IEE_c(downTrials(157:161)) = data;
% clear data;
IEE_c(upTrials(60:61)) = IEE(upTrials(60:61));
IEE_c(rightTrials(60:61)) = IEE(rightTrials(60:61));
IEE_c(leftTrials(60:61)) = IEE(leftTrials(60:61));
IEE_c(channelTrials(60:61)) = IEE(channelTrials(60:61));
data = outlier_t(IEE(upTrials(62:66))); % Outlier for last 10 post-exp
IEE_c(upTrials(62:66)) = data;
data = outlier_t(IEE(rightTrials(62:66)));
IEE_c(rightTrials(62:66)) = data;
data = outlier_t(IEE(leftTrials(62:66))); % Outlier for last 10 post-exp
IEE_c(leftTrials(62:66)) = data;
data = outlier_t(IEE(channelTrials(62:66)));
IEE_c(channelTrials(62:66)) = data;
clear data;
% transpose and calculate standardized variable
IEE_c = IEE_c';
bkup_mean = nanmean(IEE_c(upTrials(1:7)));
bkup_std = nanstd(IEE_c(upTrials(1:7)));
IEE_up_st = (IEE_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(IEE_c(rightTrials(1:7)));
bkright_std = nanstd(IEE_c(rightTrials(1:7)));
IEE_right_st = (IEE_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(IEE_c(leftTrials(1:7)));
bkleft_std = nanstd(IEE_c(leftTrials(1:7)));
IEE_left_st = (IEE_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(IEE_c(channelTrials(1:7)));
bkchannel_std = nanstd(IEE_c(channelTrials(1:7)));
IEE_channel_st = (IEE_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std;
%% Plotting Code for IEE
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
plot(upTrials(1:7),IEE(upTrials(1:7)),'bo');
hold on
plot(upTrials(1:7),IEE_c(upTrials(1:7)),'bx');
hold on
plot(rightTrials(1:7),IEE(rightTrials(1:7)),'ro');
hold on
plot(rightTrials(1:7),IEE_c(rightTrials(1:7)),'rx');
hold on
plot(leftTrials(1:7),IEE(leftTrials(1:7)),'go');
hold on
plot(leftTrials(1:7),IEE_c(leftTrials(1:7)),'gx');
hold on
plot(channelTrials(1:7),IEE(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),IEE_c(channelTrials(1:7)),'mx');
```

```
axis([0 28 0 10]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:4:20,'YTickLabel', 0:4:20,'FontName','Arial','FontSize',10);
ylabel('IEE [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.32 0.2 0.4 0.6]); hold on;
plot(upTrials(8:59)-EXP(1)+1,IEE(upTrials(8:59)),'bo');
hold on
plot(upTrials(8:59)-EXP(1)+1,IEE_c(upTrials(8:59)),'bx');
hold on
plot(rightTrials(8:59)-EXP(1)+1,IEE(rightTrials(8:59)),'ro');
hold on
plot(rightTrials(8:59)-EXP(1)+1,IEE_c(rightTrials(8:59)),'rx');
hold on
plot(leftTrials(8:59)-EXP(1)+1,IEE(leftTrials(8:59)),'go');
hold on
plot(leftTrials(8:59)-EXP(1)+1,IEE_c(leftTrials(8:59)),'gx');
hold on
plot(channelTrials(8:59)-EXP(1)+1,IEE(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,IEE_c(channelTrials(8:59)),'mx');
axis([0 208 0 10]); set(gca,'LineWidth',2,'XTick',[1 52 105 157 208],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
plot(upTrials(60:66)-POST_EXP(1)+1,IEE(upTrials(60:66)),'bo');
hold on
plot(upTrials(60:66)-POST_EXP(1)+1,IEE_c(upTrials(60:66)),'bx');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,IEE(rightTrials(60:66)),'ro');
hold on
plot(rightTrials(60:66)-POST_EXP(1)+1,IEE_c(rightTrials(60:66)),'rx');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,IEE(leftTrials(60:66)),'go');
hold on
plot(leftTrials(60:66)-POST_EXP(1)+1,IEE_c(leftTrials(60:66)),'gx');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,IEE(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,IEE_c(channelTrials(60:66)),'mx');
axis([0 28 0 10]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['post-
exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title(['Post Exposure']);
%% Movement Path Plots
% ang = 0:0.1:2.01*pi;
% r = sortData(1).TARGET_TABLE.Visual_Radius(2);
figure
subplot(2,3,1)
for i = 1:28
  if wrong_trial(i) == 0
    plot(cursorPosX{i,1}(onset(i):offset(i))*100 -Tx , cursorPosY{i,1}(onset(i):offset(i))*100 - Ty )
    hold on
  end
end
axis([-6.5 23.5 -5 15]); set(gca,'LineWidth',2,'XTick',[-6.5 8.5 23.5],'YTick',[-15 -10 0 10 15],'YTickLabel',[-15 -10 0 10
15],'FontName','Arial','FontSize',10); title('Baseline','fontsize',11);
axis square
hold on
plot(sortData(1).TARGET_TABLE.X(1),sortData(1).TARGET_TABLE.Y(1),'Color',[255/255 117/255 56/255]) %home position
hold on
```

```
plot(sortData(1).TARGET_TABLE.X(2),sortData(1).TARGET_TABLE.Y(2),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(3),sortData(1).TARGET_TABLE.Y(3),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(4),sortData(1).TARGET_TABLE.Y(4),'r')
subplot(2,3,2)
for i = 29:45
  if wrong_trial(i) == 0
    plot(cursorPosX{i,1}(onset(i):offset(i))*100 - Tx , cursorPosY{i,1}(onset(i):offset(i))*100 - Ty )
    hold on
  end
end
axis([-6.5 23.5 -5 15]); set(gca,'LineWidth',2,'XTick',[-6.5 8.5 23.5],'YTick',[-15 -10 0 10 15],'YTickLabel',[-15 -10 0 10
15],'FontName','Arial','FontSize',10); title('Early Exposure','fontsize',11);
axis square
hold on
plot(sortData(1).TARGET_TABLE.X(1),sortData(1).TARGET_TABLE.Y(1),'Color',[255/255 117/255 56/255]) %home position
hold on
plot(sortData(1).TARGET_TABLE.X(2),sortData(1).TARGET_TABLE.Y(2),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(3),sortData(1).TARGET_TABLE.Y(3),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(4),sortData(1).TARGET_TABLE.Y(4),'r')
hold on
subplot(2,3,3)
for i = 46:200
  if wrong_trial(i) == 0
    plot(cursorPosX{i,1}(onset(i):offset(i))*100 - Tx , cursorPosY{i,1}(onset(i):offset(i))*100 -Ty )
    hold on
  end
end
axis([-6.5 23.5 -5 15]); set(gca,'LineWidth',2,'XTick',[-6.5 8.5 23.5],'YTick',[-15 -10 0 10 15],'YTickLabel',[-15 -10 0 10
15],'FontName','Arial','FontSize',10); title('Exposure','fontsize',11);
axis square
hold on
plot(sortData(1).TARGET_TABLE.X(1),sortData(1).TARGET_TABLE.Y(1),'Color',[255/255 117/255 56/255]) %home position
hold on
plot(sortData(1).TARGET_TABLE.X(2),sortData(1).TARGET_TABLE.Y(2),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(3),sortData(1).TARGET_TABLE.Y(3),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(4),sortData(1).TARGET_TABLE.Y(4),'r')
hold on
subplot(2,3,4)
for i = 201:236
  if wrong_trial(i) == 0
    plot(cursorPosX{i,1}(onset(i):offset(i))*100 - Tx , cursorPosY{i,1}(onset(i):offset(i))*100 - Ty )
    hold on
  end
end
axis([-6.5 23.5 -5 15]); set(gca,'LineWidth',2,'XTick',[-6.5 8.5 23.5],'YTick',[-15 -10 0 10 15],'YTickLabel',[-15 -10 0 10
15],'FontName','Arial','FontSize',10); title('Late Exposure','fontsize',11);
axis square
hold on
plot(sortData(1).TARGET_TABLE.X(1),sortData(1).TARGET_TABLE.Y(1),'Color',[255/255 117/255 56/255]) %home position
hold on
plot(sortData(1).TARGET_TABLE.X(2),sortData(1).TARGET_TABLE.Y(2),'r')
hold on
```

```
plot(sortData(1).TARGET_TABLE.X(3),sortData(1).TARGET_TABLE.Y(3),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(4),sortData(1).TARGET_TABLE.Y(4),'r')
hold on
subplot(2,3,5)
for i = 237:250
   if wrong_trial(i) == 0
      plot(cursorPosX{i,1}(onset(i):offset(i))*100 -Tx , cursorPosY{i,1}(onset(i):offset(i))*100 -Ty )
      hold on
   end
end
axis([-6.5 23.5 -5 15]); set(gca,'LineWidth',2,'XTick',[-6.5 8.5 23.5],'YTick',[-15 -10 0 10 15],'YTickLabel',[-15 -10 0 10
15],'FontName','Arial','FontSize',10); title('Early Post-Exposure','fontsize',11);
axis square
hold on
plot(sortData(1).TARGET_TABLE.X(1),sortData(1).TARGET_TABLE.Y(1),'Color',[255/255 117/255 56/255]) %home position
hold on
plot(sortData(1).TARGET_TABLE.X(2),sortData(1).TARGET_TABLE.Y(2),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(3),sortData(1).TARGET_TABLE.Y(3),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(4),sortData(1).TARGET_TABLE.Y(4),'r')
hold on

subplot(2,3,6)
for i = 250:264
   if wrong_trial(i) == 0
      plot(cursorPosX{i,1}(onset(i):offset(i))*100 - Tx, cursorPosY{i,1}(onset(i):offset(i))*100 - Ty)
      hold on
   end
end
axis([-6.5 23.5 -5 15]); set(gca,'LineWidth',2,'XTick',[-6.5 8.5 23.5],'YTick',[-15 -10 0 10 15],'YTickLabel',[-15 -10 0 10
15],'FontName','Arial','FontSize',10); title('Late Post-Exposure','fontsize',11);
axis square
hold on
plot(sortData(1).TARGET_TABLE.X(1),sortData(1).TARGET_TABLE.Y(1),'Color',[255/255 117/255 56/255]) %home position
hold on
plot(sortData(1).TARGET_TABLE.X(2),sortData(1).TARGET_TABLE.Y(2),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(3),sortData(1).TARGET_TABLE.Y(3),'r')
hold on
plot(sortData(1).TARGET_TABLE.X(4),sortData(1).TARGET_TABLE.Y(4),'r')
hold on
%% Perpendicular Force Using Right_FS_ForceX
delay = 250; % Delay of peak force calculation to avoid measuring artifacts at movement onset.
latForce = cell(numTrials,1);
for i = 1:numTrials
   latForce{i,1} = sortData(i).Right_FS_ForceX;
end
% Peak Lateral force using Left_FS_ForceX
peakLatForce = zeros(numTrials,1);
for i = 1:numTrials
%     if wrong_trial(i) == 0 channelTrials(i) == 1
%     if upTrials(i)  == 0 && channelTrials(i) == 1
%     elseif leftTrials(i) == 0 && channelTrials(i) == 1
%     elseif rightTrials(i) == 0 && channelTrials(i) == 1
      peakLatForce(i) = max(abs(latForce{i,1}(onset(i)+delay:offset(i))));
%     else
%        peakLatForce(i) = NaN;
```

```
    %end
end
%% Outlier correction for peakLatForce
data = outlier_t(peakLatForce(upTrials(1:7))); % Outlier for baseline
peakLatForce_c(upTrials(1:7)) = data;
data = outlier_t(peakLatForce(rightTrials(1:7)));
peakLatForce_c(rightTrials(1:7)) = data;
data = outlier_t(peakLatForce(leftTrials(1:7)));
peakLatForce_c(leftTrials(1:7)) = data;
data = outlier_t(peakLatForce(channelTrials(1:7)));
peakLatForce_c(channelTrials(1:7)) = data;
clear data;
data = outlier_t(peakLatForce(upTrials(8:11))); % Outlier for initial exposure
peakLatForce_c(upTrials(8:11)) = data;
data = outlier_t(peakLatForce(rightTrials(8:11)));
peakLatForce_c(rightTrials(8:11)) = data;
data = outlier_t(peakLatForce(leftTrials(8:11)));
peakLatForce_c(leftTrials(8:11)) = data;
data = outlier_t(peakLatForce(channelTrials(8:11)));
peakLatForce_c(channelTrials(8:11)) = data;
clear data;
data = outlier_t(peakLatForce(upTrials(12:59))); % Outlier for exposure
peakLatForce_c(upTrials(12:59)) = data;
data = outlier_t(peakLatForce(rightTrials(12:59)));
peakLatForce_c(rightTrials(12:59)) = data;
data = outlier_t(peakLatForce(leftTrials(12:59)));
peakLatForce_c(leftTrials(12:59)) = data;
data = outlier_t(peakLatForce(channelTrials(12:59)));
peakLatForce_c(channelTrials(12:59)) = data;
clear data;
data = outlier_t(peakLatForce(upTrials(60:61))); % Outlier for first 10 post-exp
peakLatForce_c(upTrials(60:61)) = data;
data = outlier_t(peakLatForce(rightTrials(60:61)));
peakLatForce_c(rightTrials(60:61)) = data;
data = outlier_t(peakLatForce(leftTrials(60:61)));
peakLatForce_c(leftTrials(60:61)) = data;
data = outlier_t(peakLatForce(channelTrials(60:61)));
peakLatForce_c(channelTrials(60:61)) = data;
clear data;
data = outlier_t(peakLatForce(upTrials(62:66))); % Outlier for last post-exp
peakLatForce_c(upTrials(62:66)) = data;
data = outlier_t(peakLatForce(rightTrials(62:66)));
peakLatForce_c(rightTrials(62:66)) = data;
data = outlier_t(peakLatForce(leftTrials(62:66)));
peakLatForce_c(leftTrials(62:66)) = data;
data = outlier_t(peakLatForce(channelTrials(62:66)));
peakLatForce_c(channelTrials(62:66)) = data;
clear data;
% transpose and calculate standardized variable
peakLatForce_c = peakLatForce_c';
bkup_mean = nanmean(peakLatForce_c(upTrials(1:7)));
bkup_std = nanstd(peakLatForce_c(upTrials(1:7)));
peakLatForce_up_st = (peakLatForce_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(peakLatForce_c(rightTrials(1:7)));
bkright_std = nanstd(peakLatForce_c(rightTrials(1:7)));
peakLatForce_right_st = (peakLatForce_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(peakLatForce_c(leftTrials(1:7)));
bkleft_std = nanstd(peakLatForce_c(leftTrials(1:7)));
```

```
peakLatForce_left_st = (peakLatForce_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(peakLatForce_c(channelTrials(1:7)));
bkchannel_std = nanstd(peakLatForce_c(channelTrials(1:7)));
peakLatForce_channel_st = (peakLatForce_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std
%% Plotting Code for peakLatForce
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
% plot(upTrials(1:7),IEE(upTrials(1:7)),'bo');
% hold on
% plot(upTrials(1:7),IEE_c(upTrials(1:7)),'bx');
% hold on
% plot(rightTrials(1:7),IEE(rightTrials(1:7)),'ro');
% hold on
% plot(rightTrials(1:7),IEE_c(rightTrials(1:7)),'rx');
% hold on
% plot(leftTrials(1:7),IEE(leftTrials(1:7)),'go');
% hold on
% plot(leftTrials(1:7),IEE_c(leftTrials(1:7)),'gx');
% hold on
plot(channelTrials(1:7),peakLatForce(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),peakLatForce_c(channelTrials(1:7)),'mx');
axis([0 28 0 5]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:1:10,'YTickLabel', 0:1:10,'FontName','Arial','FontSize',10);
ylabel('Peak Lateral Force [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.32 0.2 0.4 0.6]); hold on;
% plot(upTrials(8:59)-EXP(1)+1,IEE(upTrials(8:59)),'bo');
% hold on
% plot(upTrials(8:59)-EXP(1)+1,IEE_c(upTrials(8:59)),'bx');
% hold on
% plot(rightTrials(8:59)-EXP(1)+1,IEE(rightTrials(8:59)),'ro');
% hold on
% plot(rightTrials(8:59)-EXP(1)+1,IEE_c(rightTrials(8:59)),'rx');
% hold on
% plot(leftTrials(8:59)-EXP(1)+1,IEE(leftTrials(8:59)),'go');
% hold on
% plot(leftTrials(8:59)-EXP(1)+1,IEE_c(leftTrials(8:59)),'gx');
% hold on
plot(channelTrials(8:59)-EXP(1)+1,peakLatForce(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,peakLatForce_c(channelTrials(8:59)),'mx');
axis([0 208 0 5]); set(gca,'LineWidth',2,'XTick',[1 52 105 157 208],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
% plot(upTrials(60:66)-POST_EXP(1)+1,IEE(upTrials(60:66)),'bo');
% hold on
% plot(upTrials(60:66)-POST_EXP(1)+1,IEE_c(upTrials(60:66)),'bx');
% hold on
% plot(rightTrials(60:66)-POST_EXP(1)+1,IEE(rightTrials(60:66)),'ro');
% hold on
% plot(rightTrials(60:66)-POST_EXP(1)+1,IEE_c(rightTrials(60:66)),'rx');
% hold on
% plot(leftTrials(60:66)-POST_EXP(1)+1,IEE(leftTrials(60:66)),'go');
```

```
% hold on
% plot(leftTrials(60:66)-POST_EXP(1)+1,IEE_c(leftTrials(60:66)),'gx');
%hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,peakLatForce(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,peakLatForce_c(channelTrials(60:66)),'mx');
axis([0 28 0 5]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['post-
exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title(['Post Exposure']);
%% Peak Lateral force using Right_Hand_ForceCMD_X
peakLatForceCMD = zeros(numTrials,1);
for i = 1:264
%    if wrong_trial(i) == 0 && channel_trial(i) == 1
%        if upTrials(i)  == 0 && channelTrials(i) == 1
%    elseif leftTrials(i) == 0 && channelTrials(i) == 1
%    elseif rightTrials(i) == 0 && channelTrials(i) == 1
     peakLatForceCMD(i) = max(abs(sortData(i).Right_Hand_ForceCMD_X(onset(i)+delay:offset(i))));
%    else
%        peakLatForceCMD(i) = NaN;
%    end
end
%% Outlier correction for peakLatForceCMD
data = outlier_t(peakLatForceCMD(upTrials(1:7))); % Outlier for baseline
peakLatForceCMD_c(upTrials(1:7)) = data;
data = outlier_t(peakLatForceCMD(rightTrials(1:7)));
peakLatForceCMD_c(rightTrials(1:7)) = data;
data = outlier_t(peakLatForceCMD(leftTrials(1:7)));
peakLatForceCMD_c(leftTrials(1:7)) = data;
data = outlier_t(peakLatForceCMD(channelTrials(1:7)));
peakLatForceCMD_c(channelTrials(1:7)) = data;
clear data;
data = outlier_t(peakLatForceCMD(upTrials(8:11))); % Outlier for early exposure
peakLatForceCMD_c(upTrials(8:11)) = data;
data = outlier_t(peakLatForceCMD(rightTrials(8:11)));
peakLatForceCMD_c(rightTrials(8:11)) = data;
data = outlier_t(peakLatForceCMD(leftTrials(8:11)));
peakLatForceCMD_c(leftTrials(8:11)) = data;
data = outlier_t(peakLatForceCMD(channelTrials(8:11)));
peakLatForceCMD_c(channelTrials(8:11)) = data;
clear data;
data = outlier_t(peakLatForceCMD(upTrials(12:59))); % Outlier for exposure
peakLatForceCMD_c(upTrials(12:59)) = data;
data = outlier_t(peakLatForceCMD(rightTrials(12:59)));
peakLatForceCMD_c(rightTrials(12:59)) = data;
data = outlier_t(peakLatForceCMD(leftTrials(12:59)));
peakLatForceCMD_c(leftTrials(12:59)) = data;
data = outlier_t(peakLatForceCMD(channelTrials(12:59)));
peakLatForceCMD_c(channelTrials(12:59)) = data;
clear data;
data = outlier_t(peakLatForceCMD(upTrials(60:61))); % Outlier for initial post exposure
peakLatForceCMD_c(upTrials(60:61)) = data;
data = outlier_t(peakLatForceCMD(rightTrials(60:61)));
peakLatForceCMD_c(rightTrials(60:61)) = data;
data = outlier_t(peakLatForceCMD(leftTrials(60:61)));
peakLatForceCMD_c(leftTrials(60:61)) = data;
data = outlier_t(peakLatForceCMD(channelTrials(60:61)));
peakLatForceCMD_c(channelTrials(60:61)) = data;
clear data;
```

```
data = outlier_t(peakLatForceCMD(upTrials(62:66))); % Outlier for baseline
peakLatForceCMD_c(upTrials(62:66)) = data;
data = outlier_t(peakLatForceCMD(rightTrials(62:66)));
peakLatForceCMD_c(rightTrials(62:66)) = data;
data = outlier_t(peakLatForceCMD(leftTrials(62:66)));
peakLatForceCMD_c(leftTrials(62:66)) = data;
data = outlier_t(peakLatForceCMD(channelTrials(62:66)));
peakLatForceCMD_c(channelTrials(62:66)) = data;
clear data;
%% transpose and calculate standardized variable
peakLatForceCMD_c = peakLatForceCMD_c';
bkup_mean = nanmean(peakLatForceCMD_c(upTrials(1:7)));
bkup_std = nanstd(peakLatForceCMD_c(upTrials(1:7)));
peakLatForceCMD_up_st = (peakLatForceCMD_c(upTrials) - bkup_mean)/bkup_std;
bkright_mean = nanmean(peakLatForceCMD_c(rightTrials(1:7)));
bkright_std = nanstd(peakLatForceCMD_c(rightTrials(1:7)));
peakLatForceCMD_right_st = (peakLatForceCMD_c(rightTrials) - bkright_mean)/bkright_std;
bkleft_mean = nanmean(peakLatForceCMD_c(leftTrials(1:7)));
bkleft_std = nanstd(peakLatForceCMD_c(leftTrials(1:7)));
peakLatForceCMD_left_st = (peakLatForceCMD_c(leftTrials) - bkleft_mean)/bkleft_std;
bkchannel_mean = nanmean(peakLatForceCMD_c(channelTrials(1:7)));
bkchannel_std = nanstd(peakLatForceCMD_c(channelTrials(1:7)));
peakLatForceCMD_channel_st = (peakLatForceCMD_c(channelTrials) - bkchannel_mean)/bkchannel_std;
clear bkup_mean; clear bkup_std; clear bkright_mean; clear bkright_std;
clear bkleft_mean; clear bkleft_std; clear bkchannel_mean; clear bkchannel_std
%% Plotting Code for peakLatForceCMD
figure
set(gcf,'Color','w','Position',[560 528 600 420])
hold on;
subplot('Position',[0.06 0.2 0.1 0.6]); hold on;
% plot(upTrials(1:7),IEE(upTrials(1:7)),'bo');
% hold on
% plot(upTrials(1:7),IEE_c(upTrials(1:7)),'bx');
% hold on
% plot(rightTrials(1:7),IEE(rightTrials(1:7)),'ro');
% hold on
% plot(rightTrials(1:7),IEE_c(rightTrials(1:7)),'rx');
% hold on
% plot(leftTrials(1:7),IEE(leftTrials(1:7)),'go');
% hold on
% plot(leftTrials(1:7),IEE_c(leftTrials(1:7)),'gx');
% hold on
plot(channelTrials(1:7),peakLatForceCMD(channelTrials(1:7)),'mo');
hold on
plot(channelTrials(1:7),peakLatForceCMD_c(channelTrials(1:7)),'mx');
axis([0 28 0 5]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',0:1:10,'YTickLabel', 0:1:10,'FontName','Arial','FontSize',10);
ylabel('Peak Lateral Force CMD [s]'); title('Baseline','fontsize',11);
hold on
subplot('Position',[0.32 0.2 0.4 0.6]); hold on;
% plot(upTrials(8:59)-EXP(1)+1,IEE(upTrials(8:59)),'bo');
% hold on
% plot(upTrials(8:59)-EXP(1)+1,IEE_c(upTrials(8:59)),'bx');
% hold on
% plot(rightTrials(8:59)-EXP(1)+1,IEE(rightTrials(8:59)),'ro');
% hold on
% plot(rightTrials(8:59)-EXP(1)+1,IEE_c(rightTrials(8:59)),'rx');
% hold on
% plot(leftTrials(8:59)-EXP(1)+1,IEE(leftTrials(8:59)),'go');
```

```
% hold on
% plot(leftTrials(8:59)-EXP(1)+1,IEE_c(leftTrials(8:59)),'gx');
% hold on
plot(channelTrials(8:59)-EXP(1)+1,peakLatForceCMD(channelTrials(8:59)),'mo');
hold on
plot(channelTrials(8:59)-EXP(1)+1,peakLatForceCMD_c(channelTrials(8:59)),'mx');
axis([0 208 0 5]); set(gca,'LineWidth',2,'XTick',[1 52 105 157 208],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10);
title(['Exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
hold on
subplot('Position',[0.75 0.2 0.24 0.6]); hold on;
% plot(upTrials(60:66)-POST_EXP(1)+1,IEE(upTrials(60:66)),'bo');
% hold on
% plot(upTrials(60:66)-POST_EXP(1)+1,IEE_c(upTrials(60:66)),'bx');
% hold on
% plot(rightTrials(60:66)-POST_EXP(1)+1,IEE(rightTrials(60:66)),'ro');
% hold on
% plot(rightTrials(60:66)-POST_EXP(1)+1,IEE_c(rightTrials(60:66)),'rx');
% hold on
% plot(leftTrials(60:66)-POST_EXP(1)+1,IEE(leftTrials(60:66)),'go');
% hold on
% plot(leftTrials(60:66)-POST_EXP(1)+1,IEE_c(leftTrials(60:66)),'gx');
%hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,peakLatForceCMD(channelTrials(60:66)),'mo');
hold on
plot(channelTrials(60:66)-POST_EXP(1)+1,peakLatForceCMD_c(channelTrials(60:66)),'mx');
axis([0 28 0 5]); set(gca,'LineWidth',2,'XTick',[1 14 28],'YTick',[],'YTickLabel',[],'FontName','Arial','FontSize',10); title(['post-
exposure'], 'fontsize',11); xlabel('Trials','fontsize',11);
title(['Post Exposure']);
%% Data Export
%switch Directory
if strcmp(str,'MACI64') == 1
   cd(['/Volumes/mnl/Data/Curl Field/Pilot/Curl field Kin Data/Post_Step_2']);
else
   cd(['Z:\Data\Co-Contraction\Kinematic Data\Post_Step_2']);
end
save([subID '_postStep2' '.mat'],'sortData','channelTrials','upTrials', 'leftTrials',
'rightTrials','fname','onset','offset','wrong_trial',...
   'ide', 'ide_c', 'ide_channel_st', 'ide_up_st', 'ide_left_st', 'ide_right_st', ...
   'EPE', 'EPE_c', 'EPE_channel_st', 'EPE_up_st', 'EPE_left_st', 'EPE_right_st', ...
   'IEE', 'IEE_c', 'IEE_channel_st', 'IEE_up_st', 'IEE_left_st', 'IEE_right_st', ...
   'mov_int', 'mov_int_c', 'mov_int_channel_st', 'mov_int_up_st', 'mov_int_left_st', 'mov_int_right_st',...
   'MT', 'MT_c', 'MT_channel_st', 'MT_up_st', 'MT_left_st', 'MT_right_st',...
   'norm_jerk', 'norm_jerk_c', 'norm_jerk_channel_st', 'norm_jerk_up_st', 'norm_jerk_left_st', 'norm_jerk_right_st',...
   'MLD', 'MLD_c', 'MLD_up_st', 'MLD_right_st', 'MLD_left_st', 'MLD_channel_st',...
   'rmse', 'rmse_c', 'rmse_channel_st', 'rmse_up_st', 'rmse_left_st', 'rmse_right_st',...
   'velPeak', 'velPeak_c', 'velPeak_channel_st', 'velPeak_up_st', 'velPeak_left_st', 'velPeak_right_st',...
'peakLatForce', 'peakLatForce_c','peakLatForce_up_st', 'peakLatForce_right_st', 'peakLatForce_left_st',
'peakLatForce_channel_st',...
'peakLatForceCMD', 'peakLatForceCMD_c','peakLatForceCMD_up_st', 'peakLatForceCMD_right_st', 'peakLatForceCMD_left_st',
'peakLatForceCMD_channel_st',  'channelBool')

% This program save data to do MANOVA analysis
clear all
close all
% Select the subject file
str = computer;
if strcmp(str,'MACI64') == 1
   cd('/Volumes/mnl/Data/Curl Field/Pilot/Curl field Kin Data/Post_Step_2');
```

```
else
   cd('Z:\Data\Co-Contraction\Kinematic Data\Post_Step_2'); % Home PC
end
dir_list = dir('*_postStep2.mat');    %Store subject *mat data file names in variable (struct array).
dir_list = {dir_list.name}; % filenames
dir_list = sort(dir_list);  % sorts files
A2 = length(dir_list);      % how many files to process?
hand='_rh';
numTrials = 264;
tstamp_start = zeros(numTrials,1); % allocate space
tstamp_end = zeros(numTrials,1);
target_theta = zeros(numTrials,1);
MT_st = zeros(numTrials,1);
rmse_st = zeros(numTrials,1);
ide_st = zeros(numTrials,1);
norm_jerk_st = zeros(numTrials,1);
mov_int_st = zeros(numTrials,1);
EPE_st = zeros(numTrials,1);
IEE_st = zeros(numTrials,1);
peakLatForce_st = zeros(numTrials,1);
peakLatForceCMD_st = zeros(numTrials,1);
MLD_st = zeros(numTrials,1);
channelBool = zeros(numTrials,1);
for B = 1:1:A2
   load(char(dir_list(B)));
   subject1 = str2num(fname(19:21));
   if fname(12:17) == "Abrupt"
      group1 = 1;
   elseif fname(12:17) == "Gradual"
     group1 = 2;
%    elseif fname(7:8) == "OA"
%       group1 = 3;
   end
   for trials = 1:1:numTrials
      subject2(trials,:)=subject1;
      group2(trials,:)=group1;
      if wrong_trial(trials) == 0
         tstamp_start(trials,:) = sortData(trials).Right_FS_TimeStamp(onset(trials)); %pulls timestamp of movement onset from 
time matrix
         tstamp_end(trials,:) = sortData(trials).Right_FS_TimeStamp(offset(trials));  %pulls timestamp of movement offset from 
time matrix
         end_X_pos(trials) = sortData(trials).Right_HandX(offset(trials));
         end_Y_pos(trials) = sortData(trials).Right_HandY(offset(trials));
      else
         tstamp_start(trials) = NaN; % if trial thrown out, set all to NaN
         tstamp_end(trials) = NaN;
      end
   end
   trial = [1:1:numTrials]';
   target_theta(channelTrials) = pi/2;
   target_theta(upTrials) = pi/2;
   target_theta(leftTrials) = 3*pi/4;
   target_theta(rightTrials) = pi/4;
   MT_st(channelTrials) = MT_channel_st;
   MT_st(upTrials) = MT_up_st;
   MT_st(leftTrials) = MT_left_st;
   MT_st(rightTrials) = MT_right_st;
   rmse_st(channelTrials) = rmse_channel_st;
```

```
rmse_st(upTrials) = rmse_up_st;
rmse_st(leftTrials) = rmse_left_st;
rmse_st(rightTrials) = rmse_right_st;
ide_st(channelTrials) = ide_channel_st;
ide_st(upTrials) = ide_up_st;
ide_st(leftTrials) = ide_left_st;
ide_st(rightTrials) = ide_right_st;
norm_jerk_st(channelTrials) = norm_jerk_channel_st;
norm_jerk_st(upTrials) = norm_jerk_up_st;
norm_jerk_st(leftTrials) = norm_jerk_left_st;
norm_jerk_st(rightTrials) = norm_jerk_right_st;
mov_int_st(channelTrials) = mov_int_channel_st;
mov_int_st(upTrials) = mov_int_up_st;
mov_int_st(leftTrials) = mov_int_left_st;
mov_int_st(rightTrials) = mov_int_right_st;
EPE_st(channelTrials) = EPE_channel_st;
EPE_st(upTrials) = EPE_up_st;
EPE_st(leftTrials) = EPE_left_st;
EPE_st(rightTrials) = EPE_right_st;
IEE_st(channelTrials) = IEE_channel_st;
IEE_st(upTrials) = IEE_up_st;
IEE_st(leftTrials) = IEE_left_st;
IEE_st(rightTrials) = IEE_right_st;
peakLatForce_st(upTrials) = peakLatForce_up_st;
peakLatForce_st(rightTrials) = peakLatForce_right_st;
peakLatForce_st(leftTrials) = peakLatForce_left_st;
peakLatForce_st(channelTrials) = peakLatForce_channel_st;
peakLatForceCMD_st(upTrials) = peakLatForceCMD_up_st;
peakLatForceCMD_st(rightTrials)= peakLatForceCMD_right_st;
peakLatForceCMD_st(leftTrials) = peakLatForceCMD_left_st;
peakLatForceCMD_st(channelTrials)= peakLatForceCMD_channel_st;
MLD_st(upTrials) = MLD_up_st;
MLD_st(rightTrials) = MLD_right_st;
MLD_st(leftTrials) = MLD_left_st;
MLD_st(channelTrials) = MLD_channel_st;
ALL_subjects=[group2 subject2 trial target_theta...
    MT MT_c MT_st...
    rmse rmse_c rmse_st...
    ide ide_c ide_st...
    norm_jerk norm_jerk_c norm_jerk_st...
    mov_int mov_int_c mov_int_st...
    EPE EPE_c EPE_st...
    IEE IEE_c IEE_st...
    peakLatForce peakLatForce_c peakLatForce_st...
    peakLatForceCMD peakLatForceCMD_c peakLatForceCMD_st...
    MLD MLD_c MLD_st...
    end_X_pos.' end_Y_pos.'...
    tstamp_start tstamp_end...
    wrong_trial channelBool]; %You store the current matrix
if strcmp(str,'MACI64') == 1
    cd('/Volumes/mnl/Data/Adaptation/interference_dosing/Post_Step_3');
    dlmwrite('rh_raw', ALL_subjects, '-append', 'delimiter', ',', 'precision','%.6f');
    cd('/Volumes/mnl/Data/Adaptation/interference_dosing/Post_Step_2');
else
    cd(['Z:\Data\Co-Contraction\Kinematic Data\Post_Step_3']);
    dlmwrite('rh_gradual_raw', ALL_subjects, '-append', 'delimiter', ',', 'precision','%.6f');
    cd(['Z:\Data\Co-Contraction\Kinematic Data\Post_Step_2']);
end
```

```
% the notation '%.6f' writes each variable out to six decimal places, should get rid of engineering notation

end
```

Statistical Analysis Code: R Studio

```
---
title: "Co-Contracile Responses to Dynamic Perturbation"
output: html_document
---
```{r setup, include=FALSE}
#knitr::opts_chunk$set(echo = TRUE)
## Install new packages
```{r}
#install.packages('plotrix')
#install.packages("grid")
#install.packages("reshape")
#install.packages("ez")
#install.packages("Cairo")
```
Load packages
```{r Load packages}
library(plyr)
library(ggplot2)
library(plotrix)
#library(grid)
#library(gridExtra)
#library(lattice)
#library(reshape) # This one interferes with dplyr...don't use it.
library(ez)
library(dplyr)
library(afex)
library(lme4)
library(lmerTest)
```
Column rearranger
```{r}
##arrange df vars by position
##'vars' must be a named vector, e.g. c("var.name"=1)
arrange.vars <- function(data, vars){
  ##stop if not a data.frame (but should work for matrices as well)
  stopifnot(is.data.frame(data))

  ##sort out inputs
  data.nms <- names(data)
  var.nr <- length(data.nms)
  var.nms <- names(vars)
  var.pos <- vars
  ##sanity checks
  stopifnot( !any(duplicated(var.nms)),
        !any(duplicated(var.pos)) )
  stopifnot( is.character(var.nms),
        is.numeric(var.pos) )
  stopifnot( all(var.nms %in% data.nms) )
  stopifnot( all(var.pos > 0),
        all(var.pos <= var.nr) )
```

```
  ##prepare output
  out.vec <- character(var.nr)
  out.vec[var.pos] <- var.nms
  out.vec[-var.pos] <- data.nms[ !(data.nms %in% var.nms) ]
  stopifnot( length(out.vec)==var.nr )

  ##re-arrange vars by position
  data <- data[ , out.vec]
  return(data)
}
```
```
## Load in Data
```{r Load in Data, include=FALSE}
setwd("Z:\\Data\\Co-Contraction\\Kinematic Data\\Post_Step_3") # PC
gradualData = read.delim('rh_gradual_raw',header = FALSE, sep = ",", na.strings = 'NaN')
gradualData$V2 = rep(9:16, each = 264)
abruptData = read.delim('rh_abrupt_raw',header = FALSE, sep = ",", na.strings = 'NaN')
leftieData = read.delim('lh_abrupt_raw',header = FALSE, sep = ",", na.strings = 'NaN')
testData = rbind(gradualData, abruptData, leftieData)

numGroup = 2 # Number of groups (works only for my current dataset)

colnames(testData) = c('group', 'subjectID', 'trial', 'target_theta',
            'MT', 'MT_c', 'MT_st',
            'rmse', 'rmse_c', 'rmse_st',
            'ide', 'ide_c', 'ide_st',
            'norm_jerk', 'norm_jerk_c', 'norm_jerk_st',
            'mov_int', 'mov_int_c', 'mov_int_st',
            'EPE', 'EPE_c', 'EPE_st',
            'IEE', 'IEE_c', 'IEE_st',
            'PLF', 'PLF_c', 'PLF_st',
            'PLFcmd', 'PLFcmd_c', 'PLFcmd_st',
            'MLD', 'MLD_c', 'MLD_st',
            'end_X_pos', 'end_Y_pos',
            'tstamp_start', 'tstamp_end',
            'wrong_trial', 'channel_trial')
factors = c('group', 'subjectID')
testData[,factors] = lapply(testData[,factors], factor)
# Remove PLF and PLFcmd data for NON channel trials
testData$PLF[testData$channel_trial == 0] = NA
testData$PLF_c[testData$channel_trial == 0] = NA
testData$PLF_st[testData$channel_trial == 0] = NA
testData$PLFcmd[testData$channel_trial == 0] = NA
testData$PLFcmd_c[testData$channel_trial == 0] = NA
testData$PLFcmd_st[testData$channel_trial == 0] = NA

# Remove rmse, ide, norm_jerk, EPE, IEE from channel trials
testData$rmse[testData$channel_trial == 1] = NA
testData$rmse_c[testData$channel_trial == 1] = NA
testData$rmse_st[testData$channel_trial == 1] = NA
testData$ide[testData$channel_trial == 1] = NA
testData$ide_c[testData$channel_trial == 1] = NA
testData$ide_st[testData$channel_trial == 1] = NA
testData$norm_jerk[testData$channel_trial == 1] = NA
testData$norm_jerk_c[testData$channel_trial == 1] = NA
testData$norm_jerk_st[testData$channel_trial == 1] = NA
testData$EPE[testData$channel_trial == 1] = NA
testData$EPE_c[testData$channel_trial == 1] = NA
```
```

```
testData$EPE_st[testData$channel_trial == 1] = NA
testData$IEE[testData$channel_trial == 1] = NA
testData$IEE_c[testData$channel_trial == 1] = NA
testData$IEE_st[testData$channel_trial == 1] = NA
# This will get rid of oulier subjects BEFORE generating the dataset to follow.
outliers = c(0)
for (i in 1:length(outliers)){
testData = subset(testData, testData$subjectID != outliers[i])
}
numSub = nrow(testData)/264 # number of subjects in the data.frame (scrubbed of outliers)
numTrial = nrow(testData)/numSub # number of trials per subject
# Rename entries for target_theta to indicate whether trial was "up" (90 deg), or "down" (270 deg)
testData$target_theta = as.character(testData$target_theta)
testData$target_theta = revalue(testData$target_theta, c("4.712389" = "down", "1.570796" = "up"))
testData$target_theta = as.factor(testData$target_theta)
# Normalize to baseline. I will make a new variable (*.bc) that subtracts the mean of all KB trials from all values.
# WARNING: If you get this error 'replacement has length zero' it is because plyr was loaded and is interfering with dplyr. You can fix it by putting "dplyr::" in front of the function. EXAMPLE - dplyr::group_by() %>% dplyr::summarise(). A better fix is to load dplyr AFTER plyr in the package installation chunk.
# Apply to ide_c
temp = testData %>% select(c(group, subjectID, trial, ide_c)) %>% filter(trial %in% 1:28) %>% group_by(subjectID) %>%
summarise(avg.ide.kb = mean(ide_c, na.rm = TRUE))
temp2 = data.frame(matrix(ncol = 2, nrow = numSub*264))
for(i in 1:numSub){
  temp2$X1[(264*(i-1)+1):(264*i)] = rep(temp$subjectID[i],264)
  temp2$X2[(264*(i-1)+1):(264*i)] = rep(temp$avg.ide.kb[i],264)
}
testData$ide.bc = testData$ide_c - temp2$X2
rm(temp, temp2) # Clean up workspace

# Apply to rmse_c
temp = testData %>% select(c(group, subjectID, trial, rmse_c)) %>% filter(trial %in% 1:28) %>% group_by(subjectID) %>%
summarise(avg.rmse.kb = mean(rmse_c, na.rm = TRUE))
temp2 = data.frame(matrix(ncol = 2, nrow = numSub*264))
for(i in 1:numSub){
  temp2$X1[(264*(i-1)+1):(264*i)] = rep(temp$subjectID[i],264)
  temp2$X2[(264*(i-1)+1):(264*i)] = rep(temp$avg.rmse.kb[i],264)
}
testData$rmse.bc = testData$rmse_c - temp2$X2
rm(temp, temp2) # Clean up workspace
# Apply to mov_int_c
temp = testData %>% select(c(group, subjectID, trial, mov_int_c)) %>% filter(trial %in% 1:28) %>% group_by(subjectID) %>%
summarise(avg.mov_int.kb = mean(mov_int_c, na.rm = TRUE))
temp2 = data.frame(matrix(ncol = 2, nrow = numSub*264))
for(i in 1:numSub){
  temp2$X1[(264*(i-1)+1):(264*i)] = rep(temp$subjectID[i],264)
  temp2$X2[(264*(i-1)+1):(264*i)] = rep(temp$avg.mov_int.kb[i],264)}
testData$mov_int.bc = testData$mov_int_c - temp2$X2
rm(temp, temp2) # Clean up workspace
# Apply to norm_jerk_c
temp = testData %>% select(c(group, subjectID, trial, norm_jerk_c)) %>% filter(trial %in% 1:28) %>% group_by(subjectID) %>%
summarise(avg.norm_jerk.kb = mean(norm_jerk_c, na.rm = TRUE))
temp2 = data.frame(matrix(ncol = 2, nrow = numSub*264))
for(i in 1:numSub){
  temp2$X1[(264*(i-1)+1):(264*i)] = rep(temp$subjectID[i],264)
  temp2$X2[(264*(i-1)+1):(264*i)] = rep(temp$avg.norm_jerk.kb[i],264)
}
testData$norm_jerk.bc = testData$norm_jerk_c - temp2$X2
```

~~~~
rm(temp, temp2) # Clean up workspace
# Apply to IEE_c
temp = testData %>% select(c(group, subjectID, trial, IEE_c)) %>% filter(trial %in% 1:28) %>% group_by(subjectID) %>%
summarise(avg.IEE.kb = mean(IEE_c, na.rm = TRUE))
temp2 = data.frame(matrix(ncol = 2, nrow = numSub*264))
for(i in 1:numSub){
  temp2$X1[(264*(i-1)+1):(264*i)] = rep(temp$subjectID[i],264)
  temp2$X2[(264*(i-1)+1):(264*i)] = rep(temp$avg.IEE.kb[i],264)
}
testData$IEE.bc = testData$IEE_c - temp2$X2
rm(temp, temp2) # Clean up workspace
# Apply to PLF_c
temp = testData %>% select(c(group, subjectID, trial, PLF_c)) %>% filter(trial %in% 1:28) %>% group_by(subjectID) %>%
summarise(avg.PLF.kb = mean(PLF_c, na.rm = TRUE))
temp2 = data.frame(matrix(ncol = 2, nrow = numSub*264))
for(i in 1:numSub){
  temp2$X1[(264*(i-1)+1):(264*i)] = rep(temp$subjectID[i],264)
  temp2$X2[(264*(i-1)+1):(264*i)] = rep(temp$avg.PLF.kb[i],264)
}
testData$PLF.bc = testData$PLF_c - temp2$X2
rm(temp, temp2) # Clean up workspace
# Apply to PLFcmd_c
temp = testData %>% select(c(group, subjectID, trial, PLFcmd_c)) %>% filter(trial %in% 1:28) %>% group_by(subjectID) %>%
summarise(avg.PLFcmd.kb = mean(PLFcmd_c, na.rm = TRUE))
temp2 = data.frame(matrix(ncol = 2, nrow = numSub*264))
for(i in 1:numSub){
  temp2$X1[(264*(i-1)+1):(264*i)] = rep(temp$subjectID[i],264)
  temp2$X2[(264*(i-1)+1):(264*i)] = rep(temp$avg.PLFcmd.kb[i],264)
}
testData$PLFcmd.bc = testData$PLFcmd_c - temp2$X2
rm(temp, temp2) # Clean up workspace

# Apply to MLD_c
temp = testData %>% select(c(group, subjectID, trial, MLD_c)) %>% filter(trial %in% 1:28) %>% group_by(subjectID) %>%
summarise(avg.MLD.kb = mean(MLD_c, na.rm = TRUE))
temp2 = data.frame(matrix(ncol = 2, nrow = numSub*264))
for(i in 1:numSub){
  temp2$X1[(264*(i-1)+1):(264*i)] = rep(temp$subjectID[i],264)
  temp2$X2[(264*(i-1)+1):(264*i)] = rep(temp$avg.MLD.kb[i],264)
}
testData$MLD.bc = testData$MLD_c - temp2$X2
rm(temp, temp2) # Clean up workspace
# Create the average speed variable. This is computed as the total distance (mov_int) divided by total time (MT)
testData = testData %>% mutate(avSpeed = mov_int_c/MT_c)
# Define trial numbers for the different phases
vb = 1:28
ex = 29:236
pe = 237:264
```
## Data Wrangling
```{r Caclulate means}
# Create 'block' factor
nr = nrow(testData)
blockSize = 4
testData$block = as.factor(rep(1:(numTrial/blockSize), each = blockSize, times = numSub))
testData = arrange.vars(testData, c("block" = 4))
# Take mean by block
~~~~

````
testData_mbb = testData %>% group_by(subjectID, group, block) %>% summarise_each(funs(mean(., na.rm = TRUE))) %>%
select(-c(trial, target_theta, wrong_trial))
# Take absolute value to get |CE| for each 10 trial block
#testData_mbb[4:ncol(testData_mbb)] = lapply(testData_mbb[4:ncol(testData_mbb)], function(x) abs(x)) # NOTE: this assumes
columns 1-3 are the only factors
# Find mean by group
testData_mbg = testData_mbb %>% group_by(group, block) %>% summarise_all(funs(mean(., na.rm = TRUE), std.error(., na.rm
= TRUE))) %>% select(-c(subjectID_mean))

# Create a phase variable (phase1 = vb, phase2 = kb, phase3 = ex, phase4 = pe)
phase1 = rep(1,28/blockSize)
phase2 = rep(2,208/blockSize)
phase3 = rep(3,28/blockSize)
phase = rep(c(phase1,phase2,phase3),numGroup)
testData_mbg$phase = as.factor(phase)
testData_mbg = arrange.vars(testData_mbg, c("phase"= 3))
testData_mbg$block = as.numeric(testData_mbg$block) # make block numeric
```

## Plot individual P Data
```{r Plot individual data and color by uptrial vs. downtrial. Consider this for determining mirror/iso}
#Data = subset(emgData, emgData$subID == 4)
#exposure_phase = 1:32
#ggplot(data = Data[exposure_phase,], aes(x = trial, y = ch1, color = group))+
#  geom_point()+
#  geom_smooth()
```

## Plot grouped, baseline corrected data
```{r Plot ide.bc}
pd = position_dodge(width = 0.4)
ggplot(data = testData_mbg, aes(x = block, y = ide.bc_mean, group = group))+
  geom_point(aes(color = group), position = pd)+
  geom_errorbar(data = testData_mbg, aes(x = block, ymin = ide.bc_mean-ide.bc_std.error, ymax =
ide.bc_mean+ide.bc_std.error, color = group),
            position = pd, width = 0)+
  geom_line(aes(color = group), position = pd)
```

```{r Plot rmse.bc}
pd = position_dodge(width = 0.4)
ggplot(data = testData_mbg, aes(x = block, y = rmse.bc_mean, group = group))+
  geom_point(aes(color = group), position = pd)+
  geom_errorbar(data = testData_mbg, aes(x = block, ymin = rmse.bc_mean-rmse.bc_std.error, ymax =
rmse.bc_mean+rmse.bc_std.error, color = group),
            position = pd, width = 0)+
  geom_line(aes(color = group), position = pd)
```

```{r Plot mov_int.bc}
pd = position_dodge(width = 0.4)
ggplot(data = testData_mbg, aes(x = block, y = mov_int.bc_mean, group = group))+
  geom_point(aes(color = group), position = pd)+
  geom_errorbar(data = testData_mbg, aes(x = block, ymin = mov_int.bc_mean-mov_int.bc_std.error, ymax =
mov_int.bc_mean+mov_int.bc_std.error, color = group),
            position = pd, width = 0)+
  geom_line(aes(color = group), position = pd)
```
````
````

````
## Plot grouped, baseline corrected data EXPOSURE ONLY
```{r Plot ide.bc}
pd = position_dodge(width = 0.4)
block_ex = subset(testData_mbg, testData_mbg$phase==2)$block
group_ex = subset(testData_mbg, testData_mbg$phase==2)$group
ggplot(data = subset(testData_mbg, testData_mbg$phase==2), aes(x = block, y = ide.bc_mean, group = group_ex))+
  geom_point(aes(color = group_ex), position = pd)+
  geom_errorbar(data = subset(testData_mbg, testData_mbg$phase==2), aes(x = block, ymin = ide.bc_mean-ide.bc_std.error,
ymax = ide.bc_mean+ide.bc_std.error, color = group),
            position = pd, width = 0)+
  geom_line(aes(color = group_ex), position = pd)
```

```{r Plot rmse.bc}
pd = position_dodge(width = 0.4)
block_ex = subset(testData_mbg, testData_mbg$phase==2)$block
group_ex = subset(testData_mbg, testData_mbg$phase==2)$group
ggplot(data = subset(testData_mbg, testData_mbg$phase==2), aes(x = block, y = rmse.bc_mean, group = group_ex))+
  geom_point(aes(color = group_ex), position = pd)+
  geom_errorbar(data = subset(testData_mbg, testData_mbg$phase==2), aes(x = block, ymin = rmse.bc_mean-
rmse.bc_std.error, ymax = rmse.bc_mean+rmse.bc_std.error, color = group),
            position = pd, width = 0)+
  geom_line(aes(color = group_ex), position = pd)
```

## Stats
```{r ANOVA across exposure (first -> last)}
aov_ez(subset(testData_mbb, testData_mbb$block %in% c(8,59)), id = 'subjectID', dv = 'ide.bc', between = 'group', within =
'block')
aov_ez(subset(testData_mbb, testData_mbb$block %in% c(8,59)), id = 'subjectID', dv = 'rmse.bc', between = 'group', within =
'block')
aov_ez(subset(testData_mbb, testData_mbb$block %in% c(8,59)), id = 'subjectID', dv = 'MLD.bc', between = 'group' , within =
'block')
aov_ez(subset(testData_mbb, testData_mbb$block %in% c(8,59)), id = 'subjectID', dv = 'PLF.bc', between = 'group' , within =
'block')
aov_ez(subset(testData_mbb, testData_mbb$block %in% c(8,59)), id = 'subjectID', dv = 'norm_jerk.bc', between = 'group' ,
within = 'block')

# If you want partial eta squared, use this
#aov_ez(subset(testData_mbb, testData_mbb$block %in% c(5,14)), id = 'subjectID', dv = 'mov_int.bc', between = 'group', within
= "block", anova_table = list(es = 'pes'))
```

```{r t-test at first block exposure}
t.test(subset(testData_mbb, testData_mbb$block %in% c(8) & testData_mbb$group %in% c(1))$ide.bc, subset(testData_mbb,
testData_mbb$block %in% c(8) & testData_mbb$group %in% c(2))$ide.bc)
t.test(subset(testData_mbb, testData_mbb$block %in% c(8) & testData_mbb$group %in% c(1))$rmse.bc, subset(testData_mbb,
testData_mbb$block %in% c(8) & testData_mbb$group %in% c(2))$rmse.bc)
t.test(subset(testData_mbb, testData_mbb$block %in% c(8) & testData_mbb$group %in% c(1))$MLD.bc, subset(testData_mbb,
testData_mbb$block %in% c(8) & testData_mbb$group %in% c(2))$MLD.bc)
t.test(subset(testData_mbb, testData_mbb$block %in% c(56) & testData_mbb$group %in% c(1))$PLF.bc, subset(testData_mbb,
testData_mbb$block %in% c(56) & testData_mbb$group %in% c(2))$PLF.bc)
t.test(subset(testData_mbb, testData_mbb$block %in% c(8) & testData_mbb$group %in% c(1))$norm_jerk.bc,
subset(testData_mbb, testData_mbb$block %in% c(8) & testData_mbb$group %in% c(2))$norm_jerk.bc)
````
````

```
# aov_ez(subset(testData_mbb, testData_mbb$block %in% c(60)), id = 'subjectID', dv = 'ide.bc', between = 'group')
#aov_ez(subset(testData_mbb, testData_mbb$block %in% c(60)), id = 'subjectID', dv = 'rmse.bc', between = 'group')
#aov_ez(subset(testData_mbb, testData_mbb$block %in% c(60)), id = 'subjectID', dv = 'PLF.bc', between = 'group')
#aov_ez(subset(testData_mbb, testData_mbb$block %in% c(60)), id = 'subjectID', dv = 'MLD.bc', between = 'group')
#aov_ez(subset(testData_mbb, testData_mbb$block %in% c(60)), id = 'subjectID', dv = 'norm_jerk.bc', between = 'group')
```

REFERENCES

REFERENCES

Bennett, D. J., Hollerbach, J. M., Xu, Y., & Hunter, I. W. (1992). Time-varying stiffness of human elbow joint during cyclic voluntary movement. *Experimental Brain Research*, *88*(2), 433-442.

Bennett, D. J. (1993). Electromyographic responses to constant position errors imposed during voluntary elbow joint movement in human. *Experimental brain research*, *95*(3), 499-508.

Bizzi, E., & Mussa-Ivaldi, F. A. (1998). Neural basis of motor control and its cognitive implications. *Trends in Cognitive Sciences*, *2*(3), 97-102.

Burdet, E., Osu, R., Franklin, D. W., Milner, T. E., & Kawato, M. (2001). The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature*, *414*(6862), 446.

Burdet, E., Tee, K. P., Mareels, I., Milner, T. E., Chew, C. M., Franklin, D. W., Osu, R., & Kawato, M. (2006). Stability and motor adaptation in human arm movements. *Biological cybernetics*, *94*(1), 20-32.

Desmurget, M., & Grafton, S. (2000). Forward modeling allows feedback control for fast reaching movements. *Trends in cognitive sciences*, *4*(11), 423-431.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, *47*(6), 381.

Franklin, D. W., Osu, R., Burdet, E., Kawato, M., & Milner, T. E. (2003). Adaptation to stable and unstable dynamics achieved by combined impedance control and inverse dynamics model. *Journal of neurophysiology*, *90*(5), 3270-3282.

Franklin, D. W., Burdet, E., Tee, K. P., Osu, R., Chew, C. M., Milner, T. E., & Kawato, M. (2008). CNS learns stable, accurate, and efficient movements using a simple algorithm. *Journal of neuroscience*, *28*(44), 11165-11173.

Gribble, P. L., & Ostry, D. J. (2000). Compensation for loads during arm movements using equilibrium-point control. *Experimental Brain Research*, *135*(4), 474-482.

Gribble, P. L., Mullin, L. I., Cothros, N., & Mattar, A. (2003). Role of cocontraction in arm movement accuracy. *Journal of neurophysiology*, *89*(5), 2396-2405.

Harris, C. M., & Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, *394*(6695), 780.

Heald, J. B., Franklin, D. W., & Wolpert, D. M. (2018). Increasing muscle co-contraction speeds up internal model acquisition during dynamic motor learning. *Scientific reports*, *8*(1), 16355.

Hogan, N. (1984). Adaptive control of mechanical impedance by coactivation of antagonist muscles. *IEEE Transactions on automatic control*, *29*(8), 681-690.

Imamizu, H., Miyauchi, S., Tamada, T., Sasaki, Y., Takino, R., PuÈtz, B., ... & Kawato, M. (2000). Human cerebellar activity reflecting an acquired internal model of a new tool. *Nature*, *403*(6766), 192.

Kagerer, F. A., Contreras-Vidal, J. L., & Stelmach, G. E. (1997). Adaptation to gradual as compared with sudden visuo-motor distortions. *Experimental brain research*, *115*(3), 557-561.

Kagerer, F. A., Contreras-Vidal, J. L., Bo, J., & Clark, J. E. (2006). Abrupt, but not gradual visuomotor distortion facilitates adaptation in children with developmental coordination disorder. *Human movement science*, *25*(4-5), 622-633.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, *9*(6), 718-727.

Klassen, J., Tong, C., & Flanagan, J. R. (2005). Learning and recall of incremental kinematic and dynamic sensorimotor transformations. *Experimental Brain Research*, *164*(2), 250-259.

Lametti, D. R., Houle, G., & Ostry, D. J. (2007). Control of movement variability and the regulation of limb impedance. *Journal of neurophysiology*.

Meulenbroek, R. G., Van Galen, G. P., Hulstijn, M., Hulstijn, W., & Bloemsaat, G. (2005). Muscular co-contraction covaries with task load to control the flow of motion in fine motor tasks. *Biological psychology*, *68*(3), 331-352.

Miall, R. C., & Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural networks*, *9*(8), 1265-1279.

Milner, T. E., & Cloutier, C. (1993). Compensation for mechanically unstable loading in voluntary wrist movement. *Experimental Brain Research*, *94*(3), 522-532.

Milner, T. E. (2002). Adaptation to destabilizing dynamics by means of muscle cocontraction. *Experimental brain research*, *143*(4), 406-416.

Milner, T. E., & Franklin, D. W. (2005). Impedance control and internal model use during the initial stage of adaptation to novel dynamics in humans. *The Journal of physiology*, *567*(2), 651-664.

Mitrovic, D., Klanke, S., Osu, R., Kawato, M., & Vijayakumar, S. (2010). A computational model of limb impedance control based on principles of internal model uncertainty. *PloS one*, *5*(10), e13601.

Oldfield, R. C. (1971). The assessment and analysis of handedness: the Edinburgh inventory. *Neuropsychologia*, *9*(1), 97-113.

Ostry, D. J., & Feldman, A. G. (2003). A critical evaluation of the force control hypothesis in motor control. *Experimental brain research*, *153*(3), 275-288.

Osu, R., & Gomi, H. (1999). Multijoint muscle regulation mechanisms examined by measured human arm stiffness and EMG signals. *Journal of neurophysiology*, *81*(4), 1458-1468.

Osu, R., Franklin, D. W., Kato, H., Gomi, H., Domen, K., Yoshioka, T., & Kawato, M. (2002). Short-and long-term changes in joint co-contraction associated with motor learning as revealed from surface EMG. *Journal of neurophysiology*, *88*(2), 991-1004.

Osu, R., Kamimura, N., Iwasaki, H., Nakano, E., Harris, C. M., Wada, Y., & Kawato, M. (2004). Optimal impedance control for task achievement in the presence of signal-dependent noise. *Journal of Neurophysiology*.

Rosa, M., (2015). Co-contraction Role on Human Motor Control. A Neural Basis. *Journal of Novel Physiotherapies*, *5*(1).

Selen, L. P., Beek, P. J., & Van Dieën, J. H. (2005). Can co-activation reduce kinematic variability? A simulation study. *Biological cybernetics*, *93*(5), 373-381.

Thoroughman, K. A., & Shadmehr, R. (1999). Electromyographic correlates of learning an internal model of reaching movements. *Journal of Neuroscience*, *19*(19), 8573-8588.

Van Galen, G. P., & Schomaker, L. R. (1992). Fitts' law as a low-pass filter effect of muscle stiffness. *Human movement science*, *11*(1-2), 11-21.

Visser, B., De Looze, M. P., De Graaff, M. P., & Van Dieën, J. H. (2004). Effects of precision demands and mental pressure on muscle activation and hand forces in computer mouse tasks. *Ergonomics*, *47*(2), 202-217.

Wolpert, D. M., Ghahramani, Z., & Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, *269*(5232), 1880-1882.