A novel device for precise training and perturbing of motor cortically driven forelimb behaviors in the rat

A Thesis
Presented to
The Academic Faculty

by

Anthony Nicholas Corsten

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
Weldon School of Biomedical Engineering and the Emory School of Medicine

Georgia Institute of Technology and Emory University
December 2018

A novel device for precise training and perturbing of motor cortically driven forelimb behaviors in the rat

Approved by:

Dr. Chethan Pandarinath, Advisor
School of Biomedical Engineering
Georgia Institute of Technology

Dr. Frank Hammond III
School of Biomedical Engineering
Georgia Institute of Technology

Dr. Nigel Pedersen
Department of Neurology
Emory University

Date Approved: 11/16/2018

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviation	Meaning
BCI	Brain-computer interface
SCI	Spinal Cord Injury
RFA	Rat Rostral Forelimb Area
CFA	Rat Caudal Forelimb Area
MotoTrak	A rodent training system developed by Vulintus, Inc.
ETH Pattus	A rodent training system developed by ETH Zurich
NIDAQ	National Instruments 6323 PCIe Data Acquisition Card
ESCON 24/2	Maxon Motors ESCON Module 24/2 servo motor controller
DCX16S	Maxon Motors brushed DC motor used in behavioral device
ENX16 RIO	Maxon ENX16 RIO 16-bit encoder
Nano17	ATI Nano17 6-axis Force / Torque transducer
MPR121	Freescale MPR121 proximity capacitive touch sensor controller
DC	Direct Current
DAR	Division of Animal Resources
IACUC	Institutional Animal Care and Use Committee
I/O	Input / Output
I^2C	Inter-integrated Circuit Protocol
RMS	Root-Mean Square
τ_{CONST}	The torque constant of the DCX16S (mNm / A)
V_{COMM}	The voltage command (V) to the ESCON 24/2 to control I_{COMM}
I_{COMM}	The commanded current (mA) by the ESCON 24/2 to the DCX16S

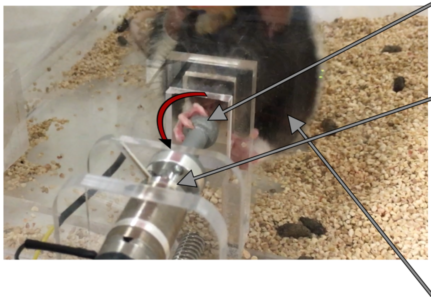
τ_{COMM}	The commanded torque (mNm) to the DCX16S, also called motor stiffness
I_{MEAS}	The measured current output (mA) by the ESCON 24/2 to the DCX16S
τ_{MEAS}	The measured motor torque (mNm) by the Nano17
τ_{PRED}	The predicted motor torque (mNm) using τ_{CONST} and I_{MEAS}
I_{KNOB}	The calculated moment of inertia (gcm^2) of the behavioral device
k_{KNOB}	The calculated radius of gyration (m) of the behavioral device
τ_{STATIC_FRIC}	The torsional static friction (mNm) of the DCX16S
τ_{FRIC}	The torsional dynamic friction (mNm) of the DCX16S
τ_{RAT}	The torque generated by the rat (mNm) used to turn the manipulandum
τ_{KNOB}	The net torque (mNm) on the manipulandum due to τ_{RAT}, τ_{FRIC} and τ_{COMM}
r_{KNOB}	The radius (m) of the manipulandum shaft
w_{W}	The weight (N) of a calibrated 10g weight
τ_{W}	The torque (mNm) generated by a 10g weight on the manipulandum shaft
α_{KNOB}	The rotational acceleration (rad/s^2) of the knob due to τ_{W}
α_{ROTOR}	The rotational deceleration (rad/s^2) of the rotor due to τ_{FRIC}
I_{ROTOR}	The manufacturer specified moment of inertia (gcm^2) of the DCX16S rotor
m_{ROTOR}	The estimated mass (g) of the DCX16S rotor
m_{KNOB}	The measured mass (g) of the manipulandum
$touch_{STATUS}$	A Boolean variable for whether the manipulandum is touched
$touch_{RAW}$	The raw output from the MPR121
$touch_{FILT}$	A two-sample lowpass filtered version of $touch_{RAW}$
$touch_{BASELINE}$	The calculated MPR121 signal baseline

SUMMARY

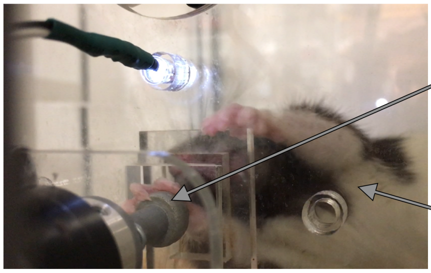
Compared to humans and non-human primates, the rat is a promising model for studying the motor cortex during structured behavioral tasks due to its low cost and rapid trainability. However, options for behavioral tools for investigating motor cortically driven forelimb behaviors are limited. Here, we developed a one-dimensional rotation manipulandum for rat forelimb supination training that has low-latency, high-resolution detection of holding and turning. Additionally, we characterized the system to accurately produce a range of torques that could be used to dynamically perturb rodent forelimb rotation behavior with high precision. Following characterization, we validated the behavioral device using two behavioral paradigms, a static holding task and a knob turning task with virtual stiffness. Naïve rats (N=2) trained on the static holding task improved their median session hold durations by an average of over 350ms from untrained holding durations of approximately 50ms. In the knob turning task (N=1), median session turning angles in a trained animal were found to decrease (88.31° to 44.02°) as knob stiffness increased between 0.18mNm and 1.44mNm. This end-to-end characterization showed our device to be effective at training and perturbing multiple potentially motor cortically driven behaviors. Ultimately, we hope to use this tool to uncover evidence of a dynamical system in rat motor cortex, like those already discovered in humans and monkeys.

VISUAL REFERENCE FOR THESIS

Forelimb supination task



Static holding task



1. INTRODUCTION

1.1 Historical approaches to motor cortex activity and its relation to movement

How does the brain encode external variables related to movement? While there have been recent advancements in the ability to record from large populations of neurons in motor cortex and other brain regions, there is still significant room for improvements in decoding neural activity. The ability to accurately model the activity of motor cortex has far-reaching implications, from neuropathological investigations into the mechanisms of stroke to significant improvement in the field of brain-computer interfaces (BCIs) [1][2][3][4].

Historically, the firing rate of motor cortical neurons was thought to directly correlate with external variables such as kinematics (e.g., hand position in reaching) or kinetics (e.g., joint torques) [5][6]. A landmark study by Georgopoulos et al. in 1982 [5] found that in a two-dimensional reaching task in monkeys, specific motor neurons tended to fire more strongly with reaches in particular directions. This has been called the representational model of motor cortex because neural activity is expected to be generated and fine-tuned to 'represent', the values of external variables. Here, external variables could be something like muscle forces or limb kinematics. This has been the de facto model for the activity of motor cortex, with hundreds of studies published on the premise of a representational system.

1.2 The dynamical systems approach to neuronal population activity

As recording techniques improved some inconsistencies have arisen with a purely representational hypothesis. Firstly, it appears that the directional or force tuning of specific neurons would sometimes change dramatically. For example, a neuron that was tuned to hand movement to the left may switch during an action to being tuned to movement to the right. Secondly, it appeared that some motor neurons had activity that was uncorrelated to movement. Thirdly, from a systems view, motor cortex contains over 100 million neurons – there are significantly fewer movement-related variables that could be controlled, so how is it possible that each neuron could represent an external variable [7] [9]?

To address these concerns, a study by Churchland et. al 2012 [7] proposed an alternative to the representational model. Instead of individual neurons being tuned for movement, the population-level activity of many neurons was instead posited to drive cortical output. Further, it was proposed that the neural population firing rate of motor cortex was driven not directly by a tuning function between neurons and external variables, but rather by a consistent set of internal governing dynamics. More specifically, in this dynamical systems model, given initial conditions for the "neural state", the change in neural activity at the current time can be used to predict future neural activity. Additionally, this dynamical system was found to be low-dimensional, partially based on the fact that the task-space is much lower dimension than the neural space. If the activity of a neuron is considered a dimension, then the activity of *N* neurons can be represented accurately with much fewer than *N* dimensions, as the neural activity is highly cross-correlated.

Several techniques have been developed in the past decade to elucidate dynamical systems in neural activity, and dynamical systems have been consistently identified in motor cortical neural recordings in monkeys and humans [8][10][11][12][13][14], lending credence to their existence. Recently, our laboratory has developed a new machine learning technique called Latent Factors

Analysis via Dynamical Systems (LFADS) to denoise neural data by modeling the underlying dynamical system [15]. LFADS is a sequential autoencoder, whose output is a denoised version of its input. Neural spiking data is an inherently noisy representation of the true activity of the system. LFADS takes in these discrete spikes as an input and predicts the true, low-dimensional underlying firing rate as its output. It has been found that denoising data with LFADS prior to kinematics decoding in a maze reaching task resulted in significantly better R^2 predictions than smoothing or other denoising methods [15].

Although these studies have shown that dynamical systems are a powerful way to model motor cortical activity, this work has only been conducted in monkeys and humans – the former are expensive and difficult to train while the latter require extensive overhead for each subject in a study. Thus, it would benefit the rehabilitative community to identify motor cortical dynamics in an animal model that is not only cost-effective, but also has the capability to be trained in a scalable manner.

1.3 The rodent as a potential model for motor cortical dynamics

While finding dynamical structure in monkeys and humans is promising, it is difficult to obtain a large sample size from these models due to study costs, lack of scalability and the large time commitment required for training. Machine learning algorithms like LFADS perform significantly better with higher sample sizes, so finding an animal model that can facilitate high data-throughput is imperative. The rat is a strong potential model for studying motor cortical dynamics, due to its rapid trainability and low-cost. Additionally, rodent caudal forelimb area (CFA) and rostral forelimb area (RFA) are somatotopically organized in a similar way to analogous structures in non-human primate and human motor cortical regions [16][17]. Despite these organizational similarities, the existence of a dynamical system has not been shown in the rat.

The exact roles of RFA and CFA in forelimb behavior are not as well-studied as the analogous areas in primates, but they have been shown to be active during some skilled forelimb movements [16][18][19]. Further, in previous studies, rat motor cortex was shown to be involved in a one-dimensional forelimb supination task [20][21] that we aim to modify and use here. Although this forelimb rotation task is promising as a potential paradigm to investigate dynamical systems, it is missing two critical elements used in paradigms for studying similar systems in monkeys and humans. Firstly, previous dynamical systems studies, such as in Churchland 2012, used multiple targets in a center-out reaching task to show that the initial conditions of the dynamical system in monkey motor cortex were consistently condition-dependent [1][7]. In the rodent forelimb supination system shown in Meyers et al. 2016 [20], called the MotoTrak, there was only a single condition, as the subject would turn a knob with a fixed-weight attached by a pulley. A simple example of additional conditions for this task could be the ability to dynamically adjust the knob stiffness or to cue turns in either direction. With the current MotoTrak design, however, these task conditions would not be possible due to physical construction limitations. Secondly, the MotoTrak did not provide a way to temporally separate reaching and turning behaviors. In the center-out reaching task experiments in monkeys, the subject's posture was rigidly enforced, and food rewards were provided without requiring the subject to leave the manipulandum, isolating planar forelimb reaching movements from other forelimb movements. It would be difficult, if not impossible, to parse out the neural activity related to knob turning without eliminating reaching movement representations in rat cortical activity. Therefore, any paradigm intended to determine the existence of a dynamical system in rat motor cortex should have both multiple task conditions and be able to temporally separate distinct motor phases of a task (e.g., reaching and turning). *To our knowledge, there is no such behavioral training device that can fulfill both requirements within the scope of one-dimensional rotation tasks for rodents.*

The goals are this project are: 1) a behavioral training device can be developed that can detect holding and turning behaviors with low-latency and high-resolution while being able to deliver a stiffness to oppose manipulandum rotation, and 2) this device can be used to train rodents to perform both forelimb supinations at varied stiffness levels and static holds of a manipulandum.

To address these questions, we first present and characterize a novel one-dimensional rotation device that can detect holding and turning behaviors with a touch sensor and high-resolution encoder, respectively. Further, our system is shown to dynamically adjust the stiffness of the manipulandum during turning using an electric motor. Finally, we validate our device on two separate behavioral paradigms to individually evaluate the effectiveness of our dynamic stiffness adjustment and static hold training. The success of validating these behavioral paradigms will directly lead into electrophysiological studies to explore the existence of a dynamical system in rodent motor cortex.

1.4 Sensor selection for assessing static holding behavior

There exist several sensors that are commercially available for detecting forelimb interactions of a manipulandum using direct measurement. Readily available options include capacitive touch sensors, load cells / pressure transducers and motor shaft encoders. Load cells are commonly used in more complex multi degree-of-freedom robotics to calculate joint angles and end-effector forces and torques but are expensive and can be complex to implement on a rotating end effector or at the scale of forces and torques applied by rodents. Alternatively, encoder measurements can indicate that a manipulandum is being interacted with but are counterintuitive for developing a paradigm for static holding behavior. An effective middle-ground are capacitive touch sensors which are low-cost, readily implementable solutions for detecting knob holding behavior. Touch sensing capability for animal work is not in itself novel and has been used in monkey [24][25][26] and rodent behavioral assessment [27][28]. However, this is the first instance that we are aware of where touch detection is incorporated into a forelimb rotation manipulandum for rodents.

1.5 Sensor selection for assessing dynamic manipulandum turning behavior

While touch detection provides relevant information during static holds, high-resolution motor shaft encoder decoding is useful for assessing knob kinematics during dynamic forelimb movements. Encoder resolutions in rodent studies run a wide range, from the lower end of 0.25° in the MotoTrak [20] to the higher end of 0.0055° in the ETH Pattus [22] [23] and 0.0027° in our system. From a cost-comparison standpoint, higher-resolution encoders are also notably more expensive: an encoder similar to that used for the MotoTrak can be found for less than $50 while higher-resolution encoders like the Gurley R119 used in the ETH Pattus can be in excess of $300. The necessity of higher or lower resolution motor shaft position decoding is heavily study-dependent. For the MotoTrak device and similar studies, low decoding resolution is appropriate because the kinematic trajectories are not intended to be heavily scrutinized, only the magnitude of the turning angle performed. On the other hand, a sophisticated robot, such as the ETH Pattus, requires high-resolution decoding for end-effector control algorithms. We are interested in assessing fast timescale compensatory forelimb movements during mechanically perturbed movements, which benefit from the highest resolution decoding possible.

1.6 Selection of a device for applying manipulandum stiffness

Lastly, we explored commonly used techniques to create rotational stiffness, or resistance to turning, in the rotation manipulandum. Stiffnesses can broadly be categorized as real or virtual. Real stiffnesses are defined as a "natural", usually inertia-based resistance to movement, e.g., a counterweight-pulley system (like in the MotoTrak) or movement against a high-friction surface. Real stiffnesses are physical phenomena and do not necessarily require a controller to generate a stiffness. Virtual stiffnesses on the other hand are usually delivered by an electric motor controlled programmatically and are not necessarily stiff without a control system. Virtual stiffnesses have a long history of use in neuroscientific research, largely in motor learning [29][30] and more recently in post-injury rehabilitation robotics [31][32][33][34]. While real stiffnesses have an advantage in providing instantaneous resistance, virtual stiffnesses provide greater flexibility in designing experimental paradigms. Unlike real stiffnesses, virtual stiffnesses are easily dynamically adjusted according to the behavior of a subject and can be used to both resist or aid a user's movements. Here, we opted for a virtual stiffness because of the greater possibilities for behavioral perturbation without adding additional mechanical complexity as would be necessary for a design based on real stiffnesses. Given the very small torques that were successfully used in the MotoTrak using real stiffnesses (less than 2mNm [21]) for rats, a large part of this study is dedicated to characterizing our ability to precisely and rapidly deliver small virtual stiffnesses to the manipulandum.

1.7 Description of behavioral paradigms

Two behavioral paradigms are presented here and were developed both for validation of the behavioral training device and for our long-term goals related to assessing the existence of dynamics in rat motor cortex. The first behavioral task is intended to validate a paradigm that uses rat forelimb in a motor cortically driven way while being able to utilize multiple task conditions. While there are many rodent paradigms that involve skilled forelimb behavior [35][36][37][38], there are significantly fewer than have been shown to require motor cortex following motor learning and fewer still are easily modifiable for different experimental conditions. The paradigm developed here is modeled after the MotoTrak, where a rat was required to supinate their forelimb to turn a knob to a specified target angle. In Meyers, the task was made more difficult with the addition of a real stiffness applied by a counterweight connected to a pulley, which was connected to the knob. *Critically*, this one-dimensional supination task was shown to involve motor cortex both by pyramidal tract lesion and GABA-based inhibition of motor cortex, which significantly reduced the ability of the subjects to reach pre-procedure peak turning angles. The paradigm developed in our study differs from Meyers in that the stiffness is virtual and is delivered by a DC motor. Using a motor over a real stiffness opens the option of a dynamically changing virtual stiffness during or between turning attempts, which the MotoTrak is unable to do. In this study, we are primarily interested in validating the usage of a virtual stiffness instead of a real stiffness in a task nearly identical to that in Meyers 2016 et al.

The second paradigm we trained is a static hold task using a novel touch detection method, where an animal is required to hold the knob without turning or releasing for an adaptive duration based on subject performance. Ideally, this paradigm could be used in conjunction with a task similar to the first behavioral paradigm to create a temporal separation between a subject's initial reach for the knob and any forelimb rotation behavior that follows. Our rationale is that this kinematic separation might also be reflected in motor cortical activity and can help tease apart different movement representations in cortex. Training rats to perform static holds is complicated by two features of rodent behavior. Rodents are not naturally inclined to perform holding behavior and are more likely (from our own observations and discussions with experts) to show impatient-like behavior. Rat motor behaviors also occur on a much faster timescale than similar tasks in humans or monkeys - for example, a monkey performing a center-out reach would typically take a second to perform their movement [7] while a forelimb rotation may only take a fifth of that time [20]. Therefore, a task that requires a rat to perform sustained static holding behavior requires a robust touch-detection method and appropriate difficulty-scaling to maintain subject motivation while still promoting learning.

1.8 A scalable system for rodent behavioral training

Scalable systems for training rodents have been shown to vastly improve data-throughput and minimize supervision time for researchers compared to traditional behavioral training [20][39][40][41][42]. Scalable training systems have several advantages over traditional training. Firstly, and most importantly, data throughput is maximized due to parallelized training. In our system, a single controller machine can manage any number of behavioral training setups. This is of critical importance to our work, as a machine learning algorithm like LFADS creates more accurate estimates based on the quantity of data it uses. Secondly, monetary costs and man-hour burdens are minimized. An automated system does not require active training from research staff, which frees this time for usage in other endeavors. The estimated one-time cost for one of our behavioral systems is less than $1300, and these systems can be modified at low-cost as well. Thirdly, an automated system reduces experimenter error by making the structure of behavioral training identical between animals. Finally, multiple paradigms can be trained simultaneously with an automated system by simply adjusting what paradigm is trained on a certain behavioral setup. This removes a typical limitation in training where lab space is dedicated for one paradigm at a time – using an automated system, each behavioral setup is a compact version of the tools needed for any paradigm if designed properly.

1.8 Long-term goals and future directions

The long-term goal of this project is to probe rodent motor cortex for evidence of a dynamical system using electrophysiological data collected from our behavioral paradigms using our novel behavioral training device. Given the similar anatomical structure of motor cortex in rats compared to primates, it might be expected that a dynamical system will exist in rat as well. Additionally, if rotational dynamics do exist, then LFADS can be expected to model the dynamics effectively which should lead to better kinematics decoding, as has been shown in monkeys and humans.

Regarding future directions of this behavioral device, the usage of a motor to generate virtual stiffnesses opens a wide range of possible research questions. In this study, virtual stiffnesses were always a step-up function to a constant value; however, using more complex torque waveforms is also an option for exploring perturbations to motor cortical dynamics and forelimb kinematics. This would be an analogous approach to using a robotic manipulandum to perturb movement dynamically for motor learning. Further, virtual stiffnesses can be applied to resist *or* aid movement which could be used to assess the short-term compensatory and long-term adaptive plasticity of motor cortex to a task where a subject must turn to a precise range of angles with these bidirectional perturbations applied. Rodents could also be a strong model for understanding the flow of information between brain regions. For example, the subject could be provided visual feedback during a task, similar to the recent study by Burgess et al. 2017 [43] in mice, which used an LCD to display the position of a manipulandum in relation to the target position. A visual transformation could be applied to the feedback without explicitly changing the physical properties of the knob to explore the transfer of information between the visual cortex, thalamus and motor cortex. Or, with a strong model of motor cortical dynamical activity, the mechanisms of recovery following stroke could be assessed by interpreting changes in the initial conditions of a dynamical system as an individual rehabilitates. Finally, translationally the work could be used to develop feedback controllers for neural prosthetics, first tested in rodent models before being translated to primate and humans.

2. MATERIALS AND METHODS

2.1 Subject information and animal care

All subjects were procured from Charles River Labs (Wilmington, MA) at 10 weeks old. Three female Long-Evans rats were chosen for this study, with N=2 for the static holding task and N=1 for the forelimb supination task. Long-Evans rats have been shown to have better success at learning skilled motor tasks compared to other models like the commonly used Sprague-Dawley albino or Fischer-344 breeds [44][45]. Further, females do not appear to differ in performance for other learning tasks, such as the Morris Water Maze [46][36], so selecting females provides the advantages of pair housing and no scent marking compared to males.

Two of the three animals were pair-housed throughout the course of the study. One rat (G) was individually housed in a modified cage without a wire rack because the subject had previously been implanted with an electrode drive. Additionally, out of the three animals, two rats (M, N) were 3 months old at the beginning of behavioral training and one (G) was 8 months old (see Table 7). During the day (approximately 10AM - 6PM), subjects were kept in the research laboratory space for training. At all other times, animals were kept in the Animal Care Facility in the building and were monitored by Division of Animal Resources (DAR) staff at Emory University. Standard 12-hour day-night light cycles were used, with the transition occurring at 7AM (lights on) or 7PM (lights off). When moved between the animal facility and the research laboratory, subjects were transported in their wire rack cages covered with a cloth cage cover on a wheeled cart. The behavioral testing and food restriction schedules were approved in IACUC protocol DAR-3000347-ELMNTS-N.

2.2 Food-restriction schedules

Subjects were food-restricted to induce motivation to perform their assigned behavioral task. Restriction was only enforced during the week (M-F), and the rodents were provided ad lib food during Friday evening and over the weekend (Saturday – Sunday afternoon). Animals were also always provided ad lib access to water, except during the 30-minute behavioral training sessions which were not equipped with water feeding equipment. Food was completely removed from cages on Sunday evenings, and a pre-restriction 'baseline' weight was recorded at that time. During the weekdays, weights were recorded once per day at the end of the training day. If the weight of total food consumed for the day was less than 10g for an animal, then the subject was provided supplemental feeding equal to the difference. The 10g value could be adjusted slightly based on individual rodent size and the rate of weight gain / loss (typically +/- 2.5g). Supplemental food pellets were provided in the form of standard rodent diet food bricks.

Subject weight was typically lowered over Monday and Tuesday, with the target of 85-90% of the baseline weight being typically reached by end-of-day on Tuesday. If the subject weight ever went below 85%, the subject was required to be placed on ad lib feeding until the weight increased to 90%. During this study, no subject went below 85% baseline body weight.

2.3 General training schedules and behavioral setup considerations

Behavioral training was sectioned into 30-minute sessions, with a minimum of 2 hours between sessions to give time for motor skill consolidation [47]. Additionally, spacing the sessions appropriately allowed the animals to regain motivation to perform the task. Typically, one training session would be done prior to noon, and one would be done in the late afternoon. At maximum, each animal was trained twice daily. Each animal was only assigned a single behavioral paradigm. Training was always conducted in the researcher's laboratory space, which was approved for animal use by Emory's IACUC Approval Committee. When not in training, animals remained housed in wire-rack cages. Subjects were only removed from the cages during a training session when moved to behavioral setups for the duration of a session. Behavioral setups were cleaned with 70% ethanol between each animal and were disinfected with Virkon every two weeks. Each behavioral setup was also filled with standard rodent bedding, which was replaced once per week. Feces was manually removed between animals.

2.4 Description of behavioral training system

Subjects were trained individually using a custom-built behavioral training system. The functional requirements in development were to make the systems 1) modular to support a multitude of training paradigms, 2) cost-effective, 3) scalable / parallelizable, 4) readily constructed and deconstructed and 5) easy to clean. Each behavioral setup consisted of a behavioral box, enclosure, and various external input / output (I/O) devices (e.g., pellet feeder, manipulandum, etc.) specific to the behavioral task being trained. A high-level overview of the system can be seen in Figure 1.

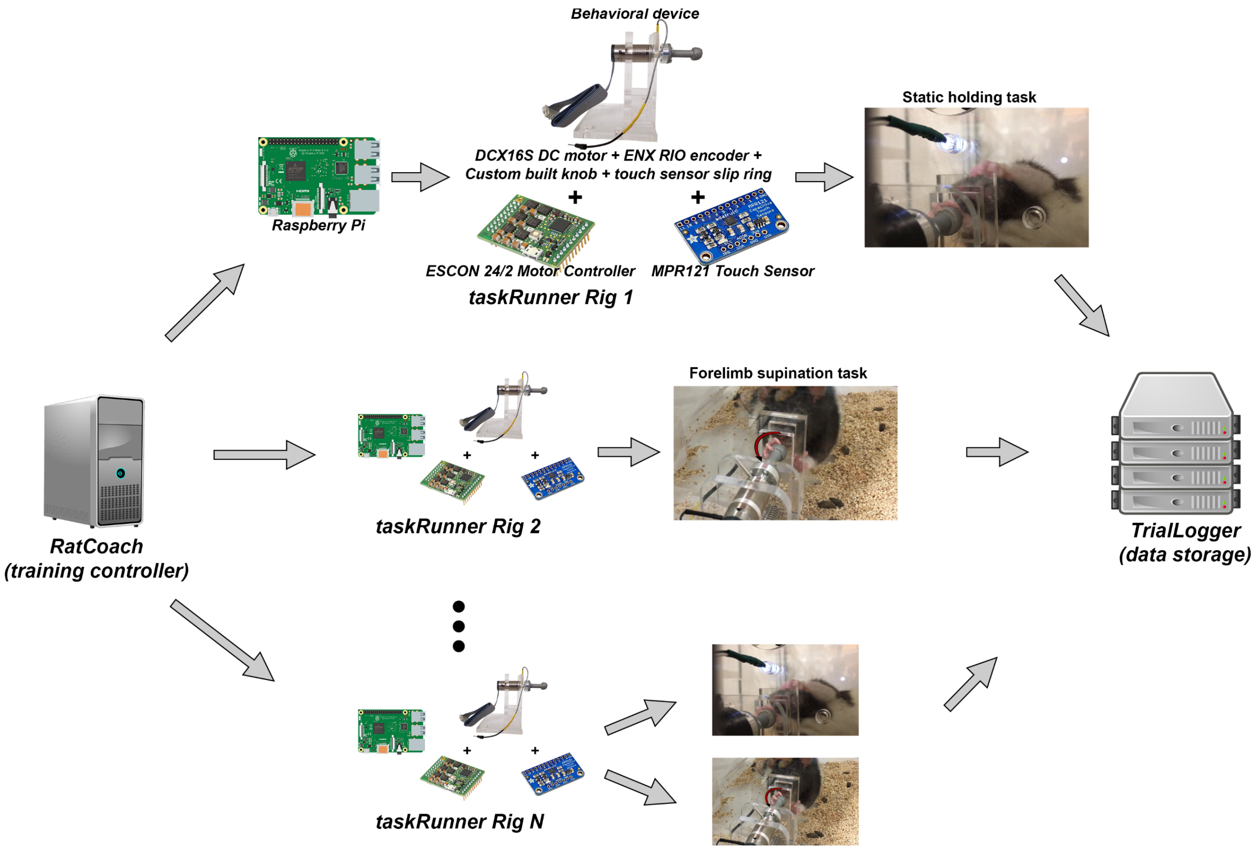


Figure 1. High-level visual representation of experimental equipment and dataflow overview. The system employed in this study allows for scalable, hands-off rat training. Animals were trained in specialized behavioral enclosures (taskRunner) that were equipped with training devices. Each taskRunner "rig" consisted of a Raspberry Pi which controlled our custom-built behavioral device comprised of a DCX16S DC motor + ENX RIO encoder combination with attached custom-built aluminum knob and slip ring. The motor was driven by an ESCON 24/2 motor controller, and the slip ring was connected to the MPR121 touch sensor for touch detection. Each taskRunner rig ran training for a single subject at a time – for example, Rig 1 might have run a session for a static holding task and Rig 2 would simultaneously train the forelimb supination task. ratCoach is a single machine that manages session times for all taskRunner rigs simultaneously and sends individualized task parameters to each rig. It can simultaneously manage sessions for any number of taskRunner rigs, which is symbolized by "taskRunner Rig *N*". trialLogger is a general name for the fileserver that received and parsed data packets from all taskRunner rigs and quickly wrote task-related data to MATLAB .mat files.

To manage costs for scalability, two versions of the behavioral system were used in the study as it was transitioned from a higher-cost proof-of-concept (Version 1) to a lower cost system (Version 2). Comparatively, the Version 2 (~$1250) system is 60% lower cost than the Version 1 system (~$3100). The complete price and part breakdowns are included in Appendix Table 8 (Version 1) and Table 9 (Version 2). The Version 1 enclosures and boxes were based on the ARTS system in [41]. The enclosures in the Version 2 system were like those in Version 1, though were designed to be easier to construct and clean. Behavioral boxes were placed inside of a melamine coated cabinet (IKEA, Leiden, Netherlands, SEKTION Cabinet). Each cabinet was separated into two by additional melamine coated medium-density fiberboard paneling (Home Depot, Atlanta, GA, 461877), and the same paneling was used to create and front and back door for the enclosures. Subjects trained on the static hold task were trained on the Version 1 system and those trained on the forelimb supination with stiffness task were trained on the Version 2 system.

For Version 2 behavioral boxes, the outer portion of the box was a cuboid measuring 11” x 11.5” x 11.5” and was built out of 5/16” thick sheet acrylic (McMaster-Carr, Elmhurst, IL, Item No. 8560K591) joined at the edges. The top and bottom panels of the box were threaded on three edges and were then fastened to the front and side panels. The back panel, which functioned as a door, was fastened to one side piece using two zinc-plated strap hinges (McMaster-Carr, Item No. 1530A31). The door also had a nickel pull handle (Amazon Inc., Seattle, WA, Item No. AB1500-SN-10) and could be latched closed using a steel flip lock (Prime-Line Products, Redlands CA, Item No. U 9888). The box also contained a removable acrylic bedding tray made of ¼” thick acrylic (McMaster-Carr, Item No. 8560K354). When in use, the bedding tray was kept in place by closing the door and activating the lock on the behavioral box. When the box was open, the tray was readily removed, and the contents could be dumped, facilitating easy cleaning (Figure 2c).

The front panel was designed to be readily modifiable to fit the specific requirements of other behavioral paradigms than what was tested here. For the paradigms in this study, the front panel had a built-in food reward pellet trough, an acrylic “reaching window”, and an attached 10.1” LCD (Figure 2) (Version 2) or LEDs (Version 1) for visual task cueing. The food reward pellet trough was built to hold pellets dispensed by a gravity-hopper pellet feeder. The acrylic reaching window functionally limited the subject to the use of a single forelimb during behavioral tasks by restricting the aperture to 15mm x 25mm. The fixed 2cm offset between the reaching window and the front of the panel provided space for any protruding implanted electrophysiology drives from restricting an animal’s ability to perform a behavioral task. The LCD panel (Digikey, Thief River Falls, MN, 1597-1102-ND) provided the capability to flexibly display task-relevant stimuli at a size that would salient to the subject. In Version 1 of the behavioral system, LEDs were used in place of the LCD screens. The behavioral training device was attached to the front panel by a 6” threaded aluminum t-slotted extrusion (McMaster-Carr, 47065T101).

The training device was mounted on an aluminum glide with ultra-high molecular weight (UHMW) plastic rails which traveled along the t-slot and could be driven by a NEMA17 stepper motor with attached lead screw (Figure 2b, Pololu, Las Vegas, NV, Item No. 2268). The stepper motor provided consistent placement of the training device between behavioral setups and was resistant to displacement. In this study, the training device placement was controlled manually, not using a stepper motor controller. The parts and usage of the behavioral training device are explained in greater detail in Section 2.5 in the Methods.

Both versions of the systems had identical functionality relating to the behavioral training tasks. In Version 1, I/O devices were controlled using an NI-6323 PCIe Data Acquisition Card (NIDAQ, National Instruments, Austin, TX), which was connected to a custom-built computer to send and receive data. The NIDAQ connections broke out to two 68-pin connector blocks, and one connector block was used for a single behavioral setup, so that one NIDAQ / computer could be used for two setups. In Version 2, a Raspberry Pi replaced the need for either a NIDAQ or a controlling computer and interacted with I/O devices directly. Both versions used a Simulink model for control of relevant I/O, where Version 1 used Simulink Desktop-Real Time and Version 2 used the Raspberry Pi Support Package for MATLAB and Simulink.

In the Version 2 system, the Raspberry Pi notably lacked both analog I/O and a built-in encoder counter as were included onboard with a NIDAQ. As an alternative, analog output was driven by an MCP4725 digital-to-analog converter chip on a breakout board, controllable via I^{2}C (Adafruit, New York, NY, Item No. 935). The analog output signal in both versions was used to control the current output of an ESCON 24/2 motor controller (Maxon Motors, Sachseln, Switzerland, 466023). The ESCON 24/2 was powered using a regulated variable DC power supply, set to 24V (Volteq, San Jose, CA, HY5003D). Details on the control schema for the ESCON 24/2 are included in Section 2.5 in the Methods. Incremental encoder measurements were made on the Version 2 system using an 32-bit quadrature encoder chip controllable via SPI (LSI/CSI, Melville, NY, Item No. LS7366R). Pellet feeders used in the experiments were either of our own design or commercially available (Lafayette Instrument Company, Lafayette, IN, Item No. 80209-45). Each behavioral setup was also equipped with a power-over-ethernet (POE) wide-angle 30fps, 1080p IP camera that automatically switched to infrared video in low-light conditions (SV3C, Shenzen, China, SV-B01POE-1080P-L) so that rodent behavior could be observed remotely.

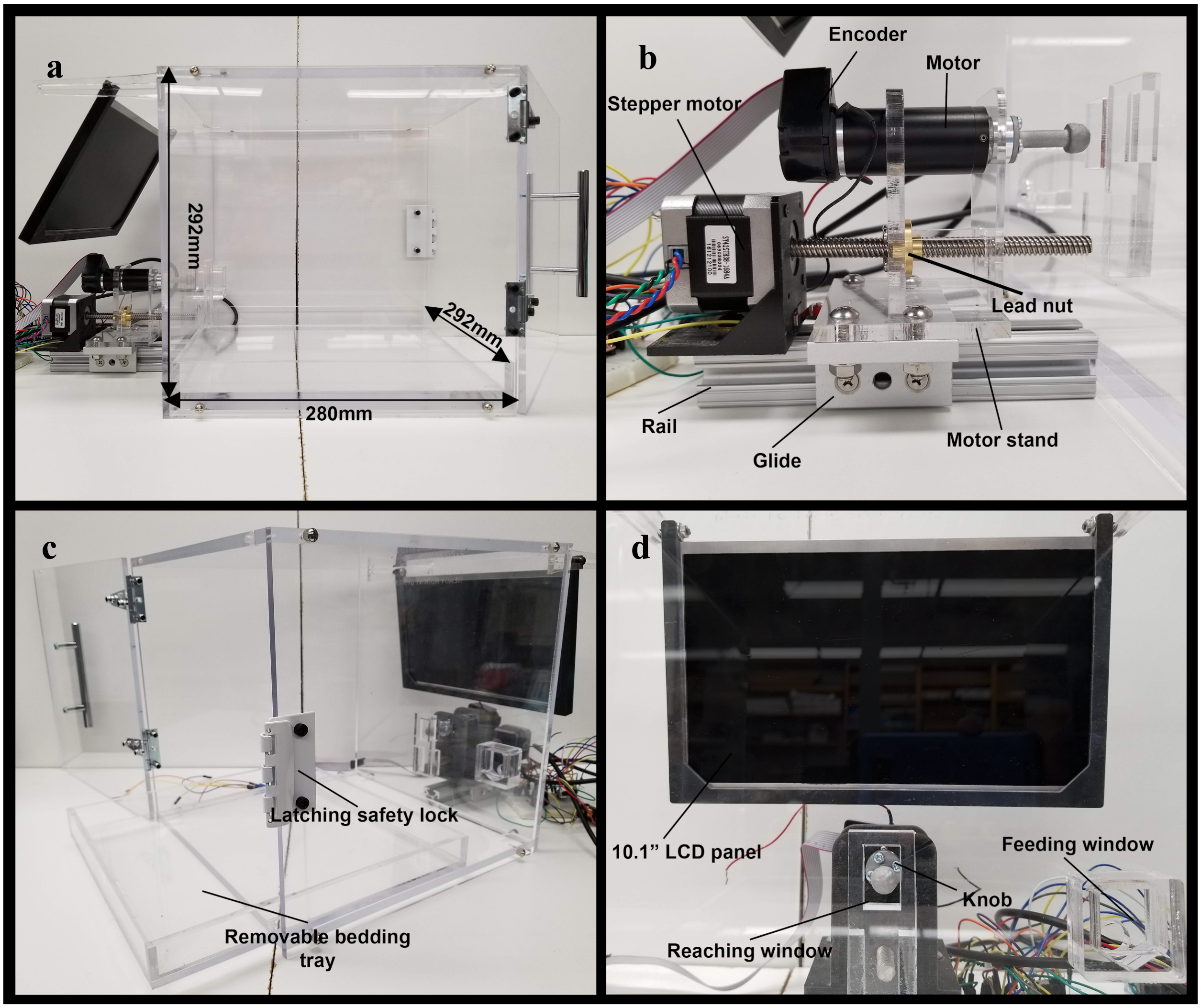


Figure 2. An overview of the "Version 2" taskRunner behavioral setup for rodent training. a) Each behavioral setup was custom built out of laser-cut acrylic. The box dimensions were roughly square, at 280mm x 292mm x 292mm (11" x 11.5" x 11.5"). b) An example manipulandum-stepper system was shown here. The manipulandum (described in detail in Figure 3) could be driven forward and backward using the stepper motor. As the stepper motor rotated, the stationary lead nut (attached to the motor stand) transferred the motor torque into linear motion of the glide along the rail. c) Rodent bedding was placed in the removable bedding tray and facilitated simple post-training waste cleanup. During a training session, the door was closed, and the safety lock was active, preventing animals from escaping while being easy to undo for researchers. d) Inside of the box, the subject needed to reach through a reaching an acrylic reaching window to reach the knob. The reaching window was designed to allow an animal to place their snout against it while performing a task without damaging a skull-implanted electrode drive. In this version of the behavioral setup, cues were provided by a 10.1" LCD panel. Older versions used LEDs (such as the setups used in the holding task). When a trial was completed successfully, a single pellet was dispensed into the feeding window by a TTL-pulse triggered feeder (not shown here).

2.5 Technical specifications of the behavioral training device

The behavioral training device was developed to flexibly probe rodent forelimb paradigms and was required to fulfill two goals: 1) provide a way evaluate kinematic compensatory changes during rodent forelimb rotation to precise mechanical perturbations or virtual stiffnesses, and 2) allow touch detection of a manipulandum without reliance on kinematic variables. Additionally, the paradigms that were developed using these devices were intended to probe motor cortical activity in future electrophysiological studies.

Goal 1: precise mechanical stiffness / perturbations. This goal was fulfilled by mounting a lightweight (9.2g) custom-built manipulandum (the "knob") to a low-rotational inertia (0.96gcm^2, Table 1), brushed DC motor (Maxon Motors, Sachseln, Switzerland, 24V DCX16S DC motor with precious metal bearings). Relevant specifications of the DCX16S are given in Table 1. The low-inertia rotor allowed for a larger range of potential virtual stiffnesses, as a high-inertia rotor would require significant effort to accelerate even with no torque generated to counter movement. Additionally, the torques generated by this motor were within the range of the MotoTrak [21]. The current output to the motor was controlled by an ESCON 24/2 motor controller, which was directly controlled by the value read on a 16-bit analog input. The ESCON 24/2 was programmable to determine a range of input voltages (e.g., 0 - 10V) that mapped linearly to a range of controllable output currents (e.g., 0 - 150mV). The direction of the motor and enabling the power stage were each controlled by additional separate digital inputs.

Table 1. Relevant Specifications from Manufacturer for 24V DCX16S DC Motor.

DCX16S Specification	Specification Value
Torque Constant (mNm / A)	36
Nominal Voltage (V)	24
Max continuous torque (mNm)	4.95
Max continuous current (mA)	143
Rotor inertia (gcm^2)	0.96
No-load current (mA)	6.91
No-load torque (mNm)	0.248
Total DCX16S + ENX16 RIO mass (g)	39

Rotational position of the knob was measured using a ENX16 RIO Encoder attached to the DCX16S motor body. The ENX16 RIO Encoder is a very high-resolution incremental optical encoder, with 131,072 counts per turn (in quadrature mode), or 0.0027° resolution. This high-resolution allowed us to accurately track the position and velocity of the knob to precisely train any turning behaviors. Additionally, a higher velocity resolution provides a more accurate measurement of low-velocity knob movements, which can indicate when a subject intended to perform a movement more quickly. In the Version 1 system, the onboard programmable function inputs (PFIs) on the NIDAQ were programmed to directly read incremental encoder values. The Version 2 system used a dedicated 32-bit quadrature encoder counter chip (LSI/CSI, Melville, NY, Item No. LS7366R), which was controlled via SPI, which was functionally identical to the PFI setup in Version 1.

Goal 2: touch detection without kinematic variables. One approach to touch detection would be to evaluate minute changes in knob position due to small, uncontrollable movements in a subject's forelimb. However, our goal was to train a stationary holding task and a turning task independently - therefore, detecting touch by kinematics was counterproductive. Additionally, from preliminary data it was found that using kinematics to determine a touch caused the subjects to make purposeful small movements of the knob to confirm touch detection, which inadvertently reinforced stereotyped small turns.

As an alternative, we instead performed touch-detection using an MPR121 capacitive touch sensor pre-soldered to a breakout board (Amazon, Seattle, WA, B00SK8PVNA). Capacitive touch sensors rely on a capacitive measurement change depending on the capacitance of the object touching the sensor electrode. The electrode port of the MPR121 was connected via a signal wire through a slip ring which was mechanically connected and electrically conductive with the knob (Figure 3). This custom-machined brass slip ring allowed for electrical transfer of capacitive changes due to knob touching without rotation of the signal wire. Both the slip ring and knob were mounted on a second-hand purchased 48V RE25 DC motor (Maxon Motors) instead of the DCX16S for the touch task. Ideally, when a subject physically interacted with the knob it would register as a decrease in capacitance. A schematic diagram of the manipulandum and the slip ring are included in Figure 4, and a complete device setup can be found in Figure 3.

The MPR121 used the I^{2}C serial data transfer protocol and communicated with a Raspberry Pi at 400kb/s. I^{2}C was also used for initializing chip registers and changing default parameters. The Raspberry Pi was controlled using a Simulink model in external target mode which compiled the model down to C/C++ and transferred the code to the Raspberry Pi. Although the real-time system sampled at 500Hz , the MPR121 was only capable of updating electrode measurements at 250Hz. The algorithm developed for detecting touches is described in detail in Section 2.6.

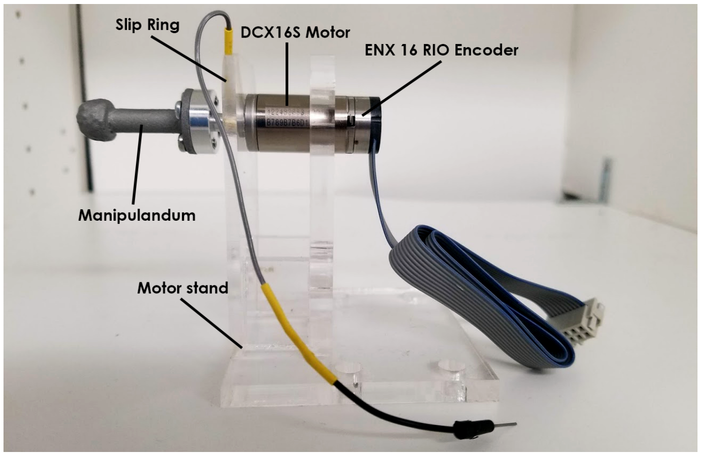


Figure 3. An overview of the behavioral device used for both behavioral tasks in this study. Rodents were trained to interact with the manipulandum either in the stationary holding or turning tasks. Electrical contact of the manipulandum was transferred through the brass slip ring to the touch sensor. The signal wire on the slip ring was soldered in place on the slip ring side and secured with a screw terminal on the touch sensor side to reduce motion noise. Rotational movements were measured using the high-resolution (16-bit) ENX 16 RIO encoder, and virtual stiffnesses were generated using the DCX16S motor to oppose rodent forelimb supination. The motor stand provided a way for the manipulandum to be driven along a track with a stepper motor.

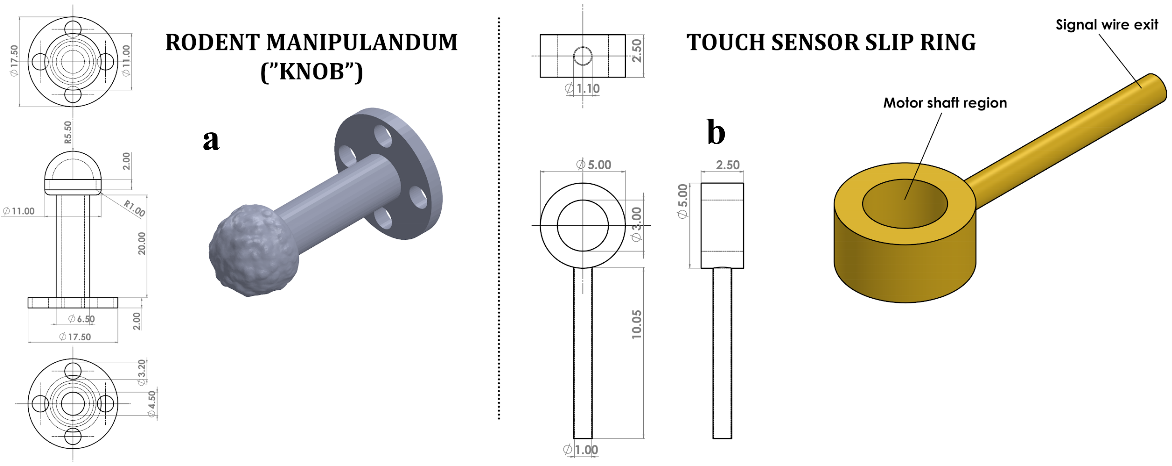


Figure 4. 3D models of the manipulandum and touch sensor slip ring. a) A 3D rendering of the manipulandum ("knob") was shown alongside three different 2D views. The knob end is 11mm in diameter, and the shaft is 2cm in length. The manipulandum was fabricated using selective laser-sintered aluminum. b) A 3D rendering of the slip ring, used to electrically connect the manipulandum to a touch sensor, is shown alongside several 2D renderings. The slip ring was specially constructed by the Emory University machine shop out of machined brass.

2.6 Touch sensing algorithm development

2.6.1 Standard methods and assumptions for capacitance-based touch-detection

Capacitive touch sensor ICs, like the MPR121 used in this study, output a unitless measurement of an electrode's measured capacitance. Accurate touch detection is reliant on a reliable "raw" measurement of this signal. Critically, two assumptions for touch detection using this raw signal are made in commercial touch sensing chips: 1) the signal should reach a steady state value or experience small levels of drift when not being touched to track a steady "baseline" that can be compared to the raw capacitance signal, and 2) biological tissues encountering the sensor should result in a decrease in the value of the raw capacitance signal. Here, a baseline was the value of the sensor when it is not being touched and should be accurately tracked even when the sensor is being touched. From preliminary data collected using our slip-ring to touch sensor setup, two major problems arose with these assumptions. First, the baseline was not constant, and often changed after the motor shaft was turned. Second, while touching the knob did induce a decrease in the sensor output, turning the knob paradoxically caused an *increase* in the sensor value, often above the baseline. These two violations in assumptions standard touch sensing detection necessitated that a novel algorithm be developed for detecting touches with this complex signal.

The algorithm developed here was like those in a commercial capacitive sensing chip like the MPR121, with a few caveats. Three signals derived from the raw MPR121 output were tracked: 1) a filtered version of the MPR121 output, 2) an estimation of the baseline and 3) a subtracted signal of the baseline signal minus the filtered signal. The primary novelty of the algorithm here was the method of properly detecting the baseline even with unexpected capacitive value changes. For actual touch detection, the touch sensing algorithm used a simple thresholding on the difference signal, which was standard amongst capacitive sensing ICs, including the MPR121. Before describing the details of how the baseline was properly tracked, an explanation of how the MPR121 was configured and how the raw output was produced is described next.

2.6.2 Description of "raw" signal obtained from MPR121

The raw output of the MPR121 ($touch_{RAW}$) was read as a 10-bit value (1024 levels) using the I^{2}C serial data transfer protocol and had *two* levels of filtering applied to the signal prior to output from the MPR121. As a note, most touch sensor chips do not give access to the raw signal output for measurement but only output the status of a touch. The first data filter was on the μs timescale, where the number of charge-discharge cycles of the sensing electrode (first filter iterations, or FFIs), total charge-discharge current (CDC) and the charge-discharge cycle time (CDT) determined the filtering employed. Ultimately, the output of the first filter was equal to the average of the capacitance measurements over the FFIs, with the minimum and maximum measurements removed. Through testing, it was determined that the CDT had very little to no effect on the filtered output. As expected, increasing the FFI had a smoothing effect (averaging over more samples), and changing the CDC changed the sensitivity of the filtered output to different capacitance values. Therefore, the FFI was set to 34 (the maximum), the CDC was set to 0.5μs (the minimum) and the CDC was set to 63μA (the maximum, which produced the greatest change in the filtered value when touching the knob). For this document, the average of the FFI will be referred to as AFFI.

The second level of filtering was on the millisecond timescale and was an average of multiple AFFIs, dictated by two parameters: second filter iterations (SFIs), which is number of AFFIs to average, and the electrode sampling interval (ESI), which is the number of milliseconds to wait between AFFI samples. Here, our goal was to maximize sampling frequency of the sensor to provide a subject with the lowest latency response to holding behavior. Therefore, SFI was set to 4 and ESI was set to 1ms (both minimum values) which resulted in a 4ms delay between new filtered sensor values. Note that despite being filtered, because this is the direct output of the MPR121 the signal will be referred to as $touch_{RAW}$, and is further filtered in 2.6.3.

2.6.3 Derivation of filtered (two-sample lowpass) signal from raw MPR121 signal

It was determined by visual inspection that the raw output of the MPR121, though filtered, was still too noisy to use in an effective touch detection algorithm. The filtering solution here was chosen to minimize the number of $touch_{RAW}$ samples required to lower latencies for behavioral training while still reducing the noisiness of the signal. Specifically, a two-sample lowpass filter was applied to the raw data resulting in a filtered signal ($touch_{FILT}$). The two-sample lowpass filter was applied using the equation:

$$touch_{FILT}(t) = u * touch_{FILT}(t-1) + (1-u) * touch_{RAW}(t) \tag{1}$$

Where $touch_{FILT}$ was the two-sample lowpass filtered signal and u was the weighting for the filter. It was found that setting u equal to 0.4 provided a good estimate of the true signal behavior.

2.6.4 Algorithm for baseline signal estimation and touch detection

As discussed in 2.6.1, the behavior of the $touch_{FILT}$ signal in this study was atypical compared to standard touch sensor usage cases due to 1) changing baselines and 2) signal increases compared to baseline, both following knob turning. In order to develop a robust touch detection method, the following assumptions were made after analyzing filtered data. Firstly, the rate of change of $touch_{FILT}$ was larger during the initial holding period *and* the hold release period than during no holding or sustained holding. Secondly, the deviation of the filtered signal due to noise did not vary greatly either within a single touch sensor setup or between setups, which provided a way to determine when the difference between the filtered and baseline signal was large enough to be discriminated as a touch. Thirdly, an increase in the capacitive signal always occurred when the knob was being turned. Using these assumptions, the following pseudocode was developed to accurately track the baseline, evaluated at each sample time.

Baseline estimation and touch detection algorithm (2)

```
IF $touch_{STATUS}$ is FALSE AND $touch_{FILT}$ is decreasing quickly THEN
    $touch_{STATUS}$ = TRUE
ELSEIF the knob is moving THEN
    $touch_{STATUS}$ is TRUE
ELSEIF $touch_{STATUS}$ is TRUE and $touch_{FILT}$ is increasing quickly THEN
    $touch_{STATUS}$ is FALSE
ELSEIF $touch_{DIFF}$ is less than the noise amplitude THEN
    $touch_{STATUS}$ is FALSE
ELSEIF $touch_{DIFF}$ is greater than or equal to the noise amplitude THEN
    $touch_{STATUS}$ is TRUE
END

IF $touch_{STATUS}$ is TRUE THEN
    $touch_{BASE}(t) = touch_{BASE}(t-1)$
ELSE
    $touch_{BASELINE}(t) = u_B * touch_{FILT}(t) + (1 - u_B) * touch_{BASELINE}(t-1)$
END
```

Here, $touch_{STATUS}$ was a Boolean that indicates whether a touch is currently registered, $touch_{BASELINE}$ was a value that tracks the true baseline of $touch_{FILT}$, $touch_{DIFF}$ was equal to $touch_{FILT} - touch_{BASELINE}$ and u_B was the baseline filter scalar. Like with $touch_{FILT}$, a two-sample lowpass filter was used for $touch_{BASELINE}$, and the value of u_B used was 0.03. Note that $touch_{BASELINE}$ was only filtered when the knob was not touched, and otherwise the value was held constant. This was to ensure that a quickly changing baseline could be tracked appropriately. Checking to see if $touch_{FILT}$ was changing quickly was accomplished by taking the numerical derivative of $touch_{FILT}$ and comparing that value to a threshold. The value of $touch_{STATUS}$ was directly used to determine whether a touch was currently registered for the holding task. The results of using this algorithm can be found in Section 3.2.

2.7 DCX16S torque generation characterization

The virtual stiffness of the manipulandum was driven by the torque generated by the DCX16S DC motor, and characterized by the static force / torque Nano17 transducer. The full equation for the torque generated at the Nano17 for static torque characterization was given as:

$$\tau_{COMM} - \tau_{STATIC_{FRIC}} - \tau_{MEAS} = 0 \tag{3}$$

Where τ_{MEAS} was the net torque measured by the Nano17, τ_{COMM} was the torque commanded to the motor and τ_{STATIC_FRIC} was the counteracting static torsional friction to rotation of the shaft. Torque generation characterization of the DCX16S was evaluated using two testing procedures to evaluate both step-response and frequency response characteristics. The ESCON 24/2 was used for motor torque control in current control mode. The ESCON 24/2 is a programmable motor controller capable of doing closed-loop current control using a built-in PI controller. The gains on this controller were not changed in this study from the manufacturer defaults.

A key assumption in this study was that τ_{COMM} would be linearly related to the commanded current (I_{COMM}, mA), which is a well-known equation that holds true for any DC motor. The equation relevant to this study is given by:

$$\tau_{COMM} = I_{COMM}\, \tau_{CONST} \tag{4}$$

Where the manufacturer specified value for the torque constant (τ_{CONST}, mNm / A) is provided in the motor specifications (Table 1). The actual value of the torque constant for a single DCX16S is computed in section 3.3 of the Results. I_{COMM} was controlled by the ESCON 24/2 and was modulated using an analog voltage input command (V_{COMM}) from a NI-6323 PCIe Data Acquisition Card (NIDAQ) controlled by a Simulink model running at 500Hz. The mapping between V_{COMM} and I_{COMM} was linear and was given by the following function:

$$I_{COMM} = I_{MAX} \frac{V_{COMM}}{V_{MAX}} \tag{5}$$

Where V_{MAX} was the maximum input voltage programmed for the ESCON 24/2, set as 5V for this study, and I_{MAX} being the corresponding maximum current output of the ESCON 24/2, set to 75mA. The sign of V_{COMM} also dictated the direction (+Z or -Z) that the motor would generate torque.

Torque measurements were made using a Nano17 6-axis force/torque transducer using the US6-2 calibration, which allowed for torque resolutions of 0.028mNm and maximum torques of 280mNm in all axes [48]. For this study, only torque in the Z-axis of the Nano17 (in the same axis as the motor) were considered. A calibration file, which was used to convert raw gauge voltages to forces and torques, was provided by ATI. The output of each gauge was read in as six differential analog inputs into the NIDAQ. Voltage measurements mapped to instantaneous current measurements made by the ESCON 24/2 were also read in as differential analog inputs into the NIDAQ. All torques used for torque testing were static torques. The torque testing rig is shown and described in Figure 5a. Notably, initial torque tests performed were done using the compliant coupling shown in Figure 5b (design provided by Dr. Frank Hammond, Georgia Tech, Atlanta, GA). The motor was attached rigidly to the outer portion of the coupling and the Nano17 to the inner portion. This coupling was developed to break if the torque or force demands of a motor approached the mechanical limitations of the Nano17, so that the motor would continue to spin without exerting forces or torques on the load cell. After initial torque tests were confirmed to be within safe limits, a more rigid coupling, seen in Figure 5c, was used.

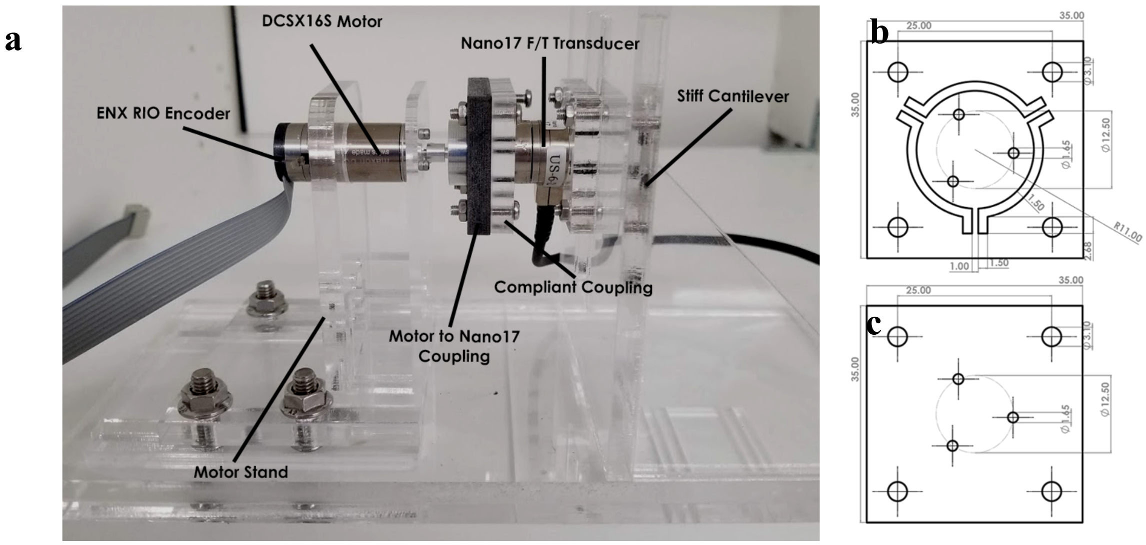


Figure 5. An overview of the torque testing rig used to evaluate torque characteristics of the DCX16S. a) Motor torques were transferred through the motor to Nano17 coupling, which were in turn transferred to the compliant coupling. The Nano17 measured static torques generated by the DCX16S. The Nano17 was cantilevered to the acrylic back panel which would act a rigid member and further increase rig stiffness. Two forms of compliant coupling were used. b) shows the more compliant coupling, which would break to prevent torque transfer in the event of a torque that exceeded Nano17 tolerances. c) was a much stiffer coupling without breakage points used after confirming that the tested torques were within tolerances and reduced low-frequency transients through increased stiffness.

2.7.1 Step-response / steady-state analysis

For step-response analysis, a series step-up / step-down ('on' / 'off') voltage commands (V_{COMM}) were generated and sent to the ESCON 24/2 to control I_{COMM}. The on / off periods were maintained according to the following function:

$$V_{COMM}(t) = \begin{cases} V_{COMM}\,('on') & 0 \leq t \leq 250ms \\ 0\,('off') & 250ms < t < 250ms + 166ms * rand([0\ 1]) \end{cases} \tag{6}$$

This piecewise 'on' and 'off' function was repeated with increasing values of V_{COMM}, starting from 0V and increasing in steps of 0.1V to a maximum of 5V (which would command the ESCON 24/2 to generate an I_{COMM} from 0 to 75mV). The random term was added to reduce 60Hz powerline noise when averaging multiple repeated experiments. A snippet of this on/off strategy can be seen in Figure 6 for a V_{COMM} of 4.9V. The full 0-5V on/off testing sequence was assumed to be repeatable and each 0-5V sequence was considered a ‘trial’. A total of 1000 trials were recorded, time-aligned and then averaged to reduce out-of-phase noise and eliminate the need for filtering.

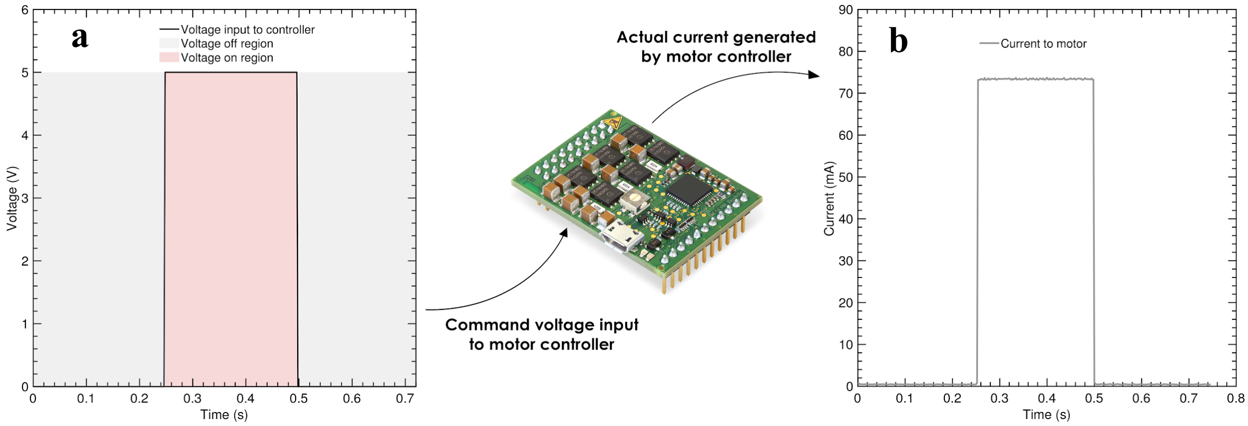


Figure 6. DCX16S steady-state response current control signal generation. a) The voltage signal generated from the NIDAQ, V_{COMM}, sent to the ESCON 24/2 was applied using alternating 250ms on (pink highlight) and off (grey highlight) periods. The transition between V_{COMM} ‘on’ and V_{COMM} ‘off’ are step functions, shown as the black trace b) V_{COMM} was used as an input to the ESCON 24/2 and was mapped to generate a PI-controlled current output (I_{COMM}) which was output to the DCX16S motor. An example measurement of I_{COMM} is shown as the grey trace.

2.7.2 Frequency response analysis

The frequency response of the motor was also assessed to study torque characteristics with input currents of varied frequency. Periods when V_{COMM} was ‘on’ and ‘off’ was controlled in a similar manner to the step-response analysis. The specific function is given by:

$$V_{COMM} = \begin{cases} V_{COMM}\,('on') & 0 \le t \le t_{on} \\ 0\,('off') & t_{on} < t < 250ms + 166ms * rand([0\ 1]) \end{cases} \tag{7}$$

$$t_{on} = \frac{Number\ of\ V_{COMM}\ Cycles}{V_{COMM}\ Frequency\ (Hz)} \tag{8}$$

Here, the 'on' period lasts for a duration t_{on}, which was the time to complete a predetermined number of cycles at the selected V_{COMM} frequency. The frequencies tested, and their associated number of cycles tested are provided in Table 2. Lower frequencies were tested with fewer cycles because it was found that a higher cycle count did not improve post-analysis quality, and greater cycle count greatly increased testing times.

Table 2. Duration of Testing Cycles for Frequency Response Analysis.

Frequency of V_{COMM} (Hz)	Number of cycles tested	Duration of t_{on} (s)
1	10	10
2	15	7.5
4	20	5
8	20	2.5
16	20	1.25
32	20	0.675

Unlike the step-response testing, the value of V_{COMM} changed within the 'on' period. V_{COMM} was generated using the following equation:

$$V_{COMM}(t) = \mathrm{e}^{\frac{-(t-\mu)^2}{2\sigma^2}}\sin(2\pi f t) \tag{9}$$

Where V_{COMM} was equal to a sinusoid with amplitude of one and a frequency described in Table 2, multiplied by a gaussian function. The gaussian's parameters were defined as $\sigma = \frac{3.33}{V_{COMM}\,Frequency\,(Hz)}$ (where 3.33 was empirically determined) and $\mu = \frac{t_{on}}{2}$. By multiplying the sinusoid and the gaussian together, a 'wavelet-like' V_{COMM} function was created. The wavelet was chosen so that the amplitude of V_{COMM} would more slowly increase to a maximum and eliminate potential transients, so that the frequency response could be properly characterized. Next, V_{COMM} was min-max normalized so that one cycle in a trial would have a maximum amplitude (Figure

7a). Finally, the min-maxed V_{COMM} was multiplied by 5V as the voltage output from the NIDAQ to the ESCON 24/2. An example of the functions involved in producing the wavelet for a 4Hz V_{COMM} signal is provided in Figure 7. Identical to the steady-state analysis, 1000 trials were collected at each frequency, time-aligned and then averaged.

Post analysis of the frequency response data was done by computing the magnitude change between the predicted torque from the measured current, τ_{PRED}, and the measured torque, τ_{MEAS}. This change was measured in decibels using the following equation:

$$Magnitude\ (dB)\ =\ 10log_{10}\frac{RMS(\tau_{MEAS})}{RMS(\tau_{PRED})} \qquad (10)$$

Where the RMS was used because the input and output signals were both sinusoidal, and the RMS is an effective measurement for computing the average of sinusoids. RMS values were calculated using built-in MATLAB functions. Cross-correlation analysis in MATLAB was used to compute the time delay between τ_{MEAS} and τ_{PRED}, where a positive delay indicated that τ_{MEAS} occurred after τ_{PRED}.

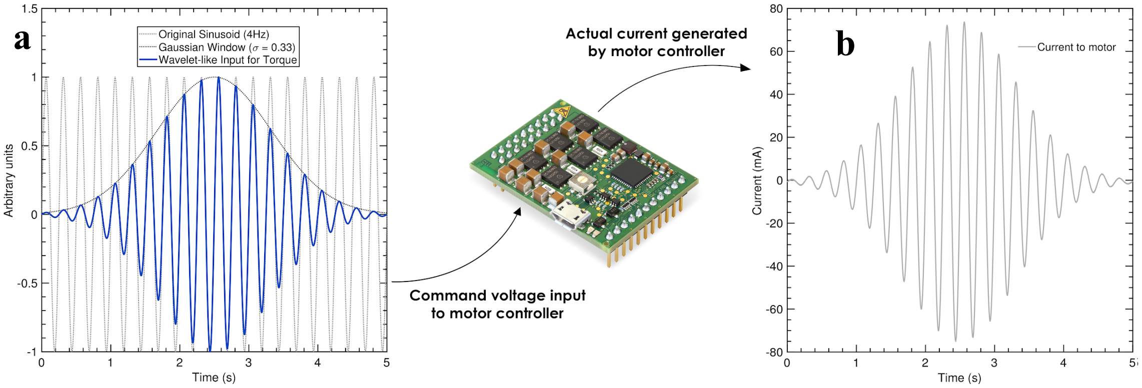


Figure 7. DCX16S frequency response current control signal generation. a) A wavelet-like command voltage (V_{COMM}) from the NIDAQ was used to control the output current of the ESCON 24/2 (I_{COMM}) to the DCX16S. The wavelet was generated by multiplying a gaussian window (dotted black) by a sinusoid (dotted grey). The frequency of the sinusoid was varied based on the frequency to be tested. The resultant product was min-max normalized to generate the voltage

output in arbitrary units (blue). The arbitrary units shown here were multiplied by 5V to obtain V_{COMM}. b) I_{COMM} was controlled using a PI-controller on the ESCON 24/2 and was delivered to the DCX16S. Recorded data for I_{COMM} is shown as the grey trace.

2.8 Manipulandum moment of inertia and motor torsional friction characterization

A complete characterization of relevant parameters to rodent turning behavior included calculating the moment of inertia of the knob (I_{KNOB}) and the dynamic torsional friction for the DCX16S (τ_{FRIC}). A useful outcome of computing these two values is the ability to calculate the total torque on the manipulandum and torque produced by the rat forelimb. The relevant equation to computing the manipulandum net torque was given by:

$$\tau_{KNOB} = I_{KNOB}\alpha_{KNOB} = \tau_{RAT} - \tau_{FRIC} - \tau_{COMM} - \tau_{AIR} \qquad (11)$$

Where τ_{KNOB} was the net torque on the manipulandum, τ_{RAT} was the torque produced by a subject, τ_{FRIC} was the torsional friction opposing the rat produced torque, τ_{AIR} was the air resistance-based friction and τ_{COMM} was the torque commanded to the DCX16S. τ_{COMM} was known because it was the torque value commanded by the user to the ESCON 24/2. τ_{AIR} was assumed to be small and therefore assumed to be negligible compared to the effects of the other torques. The rotational acceleration of the manipulandum, α_{KNOB} was found experimentally by using decoded values from the ENX16 RIO encoder attached to the DCX16S. The remaining unknown variables, I_{KNOB} and τ_{FRIC}, were determined using two separate experiments, outlined in 2.8.1 and 2.8.2.

2.8.1 Calculation of the dynamic torsional friction

The dynamic torsional friction (τ_{FRIC}) was calculated by accelerating the DCX16S shaft (with no knob connected) with an arbitrary initial torque and allowing it to decelerate due to friction. The full equation (neglecting air resistance) is given by:

$$\tau_{ROTOR} = I_{ROTOR}\alpha_{ROTOR} = -\tau_{FRIC} \qquad (12)$$

Where τ_{ROTOR} was the torque on the DCX16S rotor with nothing connected, I_{ROTOR} was the moment of inertia of the DCX16S rotor and α_{ROTOR} is the deceleration of the rotor due to friction. Experimentally, the rotor was pinched between the thumb and forefinger and turned quickly. α_{ROTOR} was computed by decoding the ENX RIO sampled at 500Hz using a program written in MATLAB. The resultant α_{ROTOR} was computed by taking the slope of the measured velocity (rad/s) at two data points. I_{ROTOR} was taken from the manufacturer specification sheet (Table 1). After discussing with an expert on DC motors, it was determined that τ_{FRIC} could be assumed to be a constant value at the rotational speeds for our behavioral tasks.

2.8.2 Calculation of the manipulandum moment of inertia

The moment of inertia was calculated by rotating the knob (connected to the motor shaft) with a known constant torque to produce a constant acceleration. The relevant equation is given by:

$$\tau_{KNOB} = I_{KNOB}\alpha_{KNOB} = \tau_W - \tau_{FRIC} \tag{13}$$

Which is solved for I_{KNOB} to obtain the equation:

$$I_{KNOB} = \frac{\tau_W - \tau_{FRIC}}{\alpha_{KNOB}} \tag{14}$$

Here, the known torque, τ_W, was generated by attaching a 10g calibrated weight to a string. The string was then secured to the shaft of the manipulandum, and the manipulandum was rotated such that the string was wound around the shaft. Then, the knob was released, and the torque generated by the weight caused the knob to "unwind". The string was assumed to have negligible mass and radius. α_{KNOB} was computed by decoding the ENX RIO sampled at 500Hz using a program written in MATLAB, by taking the slope of the knob velocity (rad /s) using two data points. and τ_{KNOB} was computed using $\tau_W = w_{\mathrm{W}} r_{KNOB}$, where w_{W} is the weight of the 10g calibrated weight (N) and r_{KNOB} was the radius of the manipulandum shaft (Table 1) plus the thickness of a piece of tape used for the string [49]. The value of I_{KNOB} was then checked by calculating the radius of gyration using the equation:

$$k_{KNOB} = \sqrt{\frac{I_{KNOB}}{m_{KNOB} + m_{ROTOR}}} \tag{15}$$

Where k_{KNOB} was equal to the distance from the rotational axis of the knob that a point mass with equal moment of inertia would be. No value of m_{ROTOR} was provided by the manufacturer, so it was assumed to be half of the total mass of DCX16S and the ENX16 RIO (Table 1). The radius of gyration should have been greater than the minimum radius of the manipulandum and less than the maximum radius to be considered a reasonable value.

2.9 Pre-behavioral handling

All subjects were handled for a minimum of two 30-minute sessions. Handling involved any human touching or holding. Prior to habituation handling, subjects would exhibit stress-behaviors (defecation, urination, refusal to groom, attempts to burrow, etc.) if handled. Therefore, handling was important for ensuring that each subject would not be stressed when being handled and transferred from cage to behavioral setup. Habituation was conducted by researchers wearing appropriate personal protective equipment (nitrile gloves and either a lab coat or water-resistant gown), where the researcher would place the animal on their arm and allow it to explore freely. A typical endpoint for handling was when an animal would actively groom and ceased handling-induced defecation and urination when handled by a researcher.

2.10 Pre-behavioral testing: Skilled Reaching Task (SRT)

Subjects were required to complete the skilled reaching task (SRT) prior to inclusion in either of the training paradigms. In the SRT, an animal reached with their forelimb through a small (1.5cm x 4cm) aperture and retrieved a food pellet reward (Figure 8) [42]. Initially, pellets were placed directly inside of the aperture to motivate the subjects and habituate them to the chocolate pellets used in this study. Then, pellets were placed at increased distances from the aperture, and was typically retrieved by the subject using the tongue. Finally, the food pellet was placed on a platform located at a distance where the animal could not use its tongue and was required to use its forelimb for retrieval. Pellets placed on the left side of the panel could only be reached with the right hand and vice versa. Pellets were always placed on both the left and right side at an equal distance from the opening to not bias which forelimb the subject would choose. If a subject performed twenty consecutive reaches for reward pellets on any of the training sessions, they were included in the study. In preliminary data collection, one non-learner was identified; however, in this study all subjects used (N = 3) were successful on the SRT. Pellet reaching success rates were not included as criteria for study eligibility.

The SRT had three important uses. Firstly, it allowed the researchers to determine the ‘handedness’ of the subjects, because rats have a limb preference like that in humans. This limb preference was used later in other behavioral training paradigms where a subject was trained according to their limb preference. Secondly, the SRT separated ‘non-learner’ subjects from those that can be expected to learn a complex task. Subjects that could not complete the SRT successfully (e.g., using the tongue only and refusing to use the forelimb) were excluded from the study. Each animal was allowed a maximum of six, 30-minute sessions over three days to perform a forelimb reaching for a reward pellet before being considered a non-learner. Thirdly, it was a subject’s first exposure to retrieving reward pellets as food instead of standard rodent chow, and provided time to overcome any potential food neophobia [50].

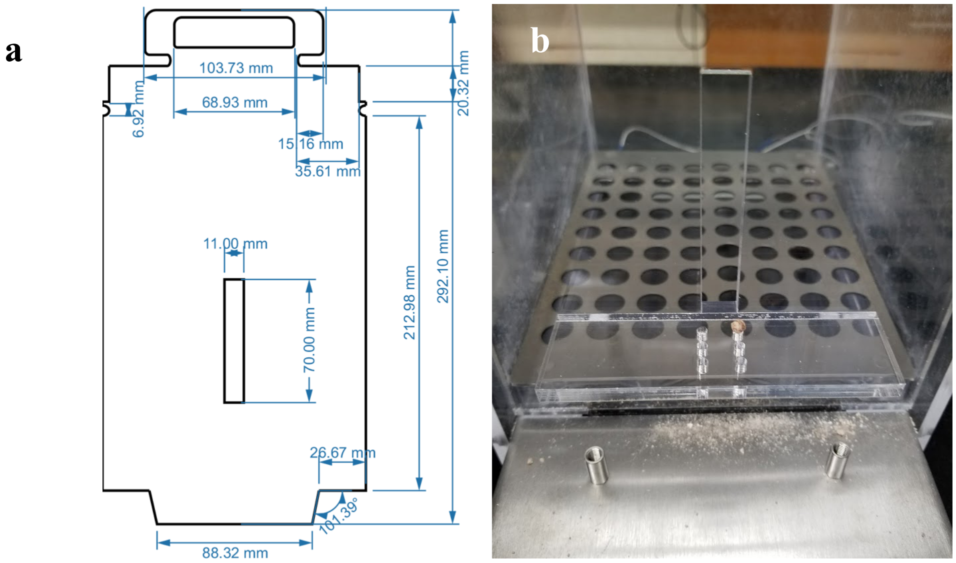


Figure 8. Skilled Reaching Task testing rig physical description. a) A dimensioned schematic of the reaching panel. The SRT panel used in this study has a reaching aperture of 1.1cm x 7cm to only allow a single forelimb through at a time. Other dimensions are included to give a relative size of the printed panel, which was used in a modified Vulintus MotoTrak box. b) Pellets were placed on the platform in either the left or right holes, relative to the point of view of the subject. Here, a single pellet is shown on the left side of the platform. Placing a pellet on the left side required a reach of the right forelimb and the opposite was true for a pellet placed on the right side.

2.11 General structure of training paradigms

Both behavioral paradigms in this study were structured with two guidelines in mind. First, paradigms are broken into discrete, sequential *stages* that were intended to create a minimal number of associations that will lead to the intended behavior. Therefore, each task introduced only one or two associations (e.g., sound-food association) at a time to limit the chance that a subject would create an incorrect association. Secondly, each task had to have discrete endpoints for progression, either to the next task or ramping in difficulty for the same task. This guideline was set both because it made training assessment more objective and because it allowed the tasks to be readily added into an automated training system. A description on how knob kinematics (angular position, velocity, etc.) were measured during both static hold and forelimb supination tasks is included in Figure 11.

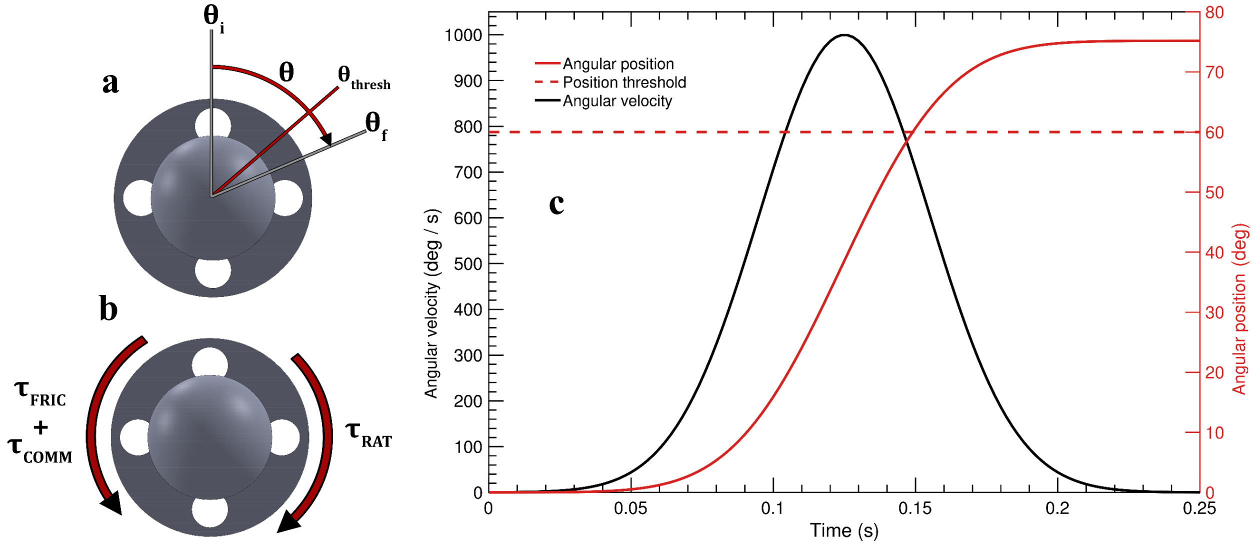


Figure 9. Example of a simulated supination knob turn with a right forelimb that successfully crosses the required position threshold during the supination task. a) Subjects performing forelimb supination tasks were required to turn the knob an angle θ from an initial angular position (θ_i) to a final angular position (θ_f) that exceeded a threshold value (θ_{THRESH}) in a single turning attempt. b) During trials with virtual stiffness, the torque applied by the rat (τ_{RAT}) was opposed by the dynamic torsional motor friction (τ_{FRIC}) and motor torque (τ_{COMM}). c) a simulated rodent turning trial lasting 0.25s was shown. The simulated angular velocity (constructed from a gaussian function, $\sigma = 0.03$s, $\mu = 0.125$s) shown in black and angular position (numerically integrated from velocity) shown in red provided to show the turn trajectory.

2.12 Paradigm 1: static knob hold

An overview of the substages trained in this paradigm is shown in Figure 10.

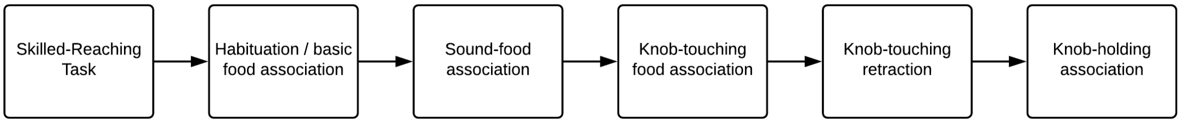


Figure 10. Overview of the substages trained in the static knob holding paradigm. All subjects were initially screened for paw-preference and for separating learners / non-learners with the skilled-reaching task. In habituation, subjects were placed in the behavioral boxes and allowed to explore freely. In sound-food association, subjects learned to associate a 4kHz tone with a food pellet being dispensed and going to the feeding chamber, which marked the end of a trial. In the knob-touching food association, subjects were rewarded for performing very short touches of the manipulandum with either forelimb or snout. In knob-touching retraction, the manipulandum, which was initially inside of the behavioral box, was retracted over several sessions until it was far enough away from a subject that they needed to use their forelimb to touch the knob. In knob-holding association, subjects learned to perform static holds of the knob for increasing durations, scaled based on subject performance.

2.12.1 Habituation / basic food association

Subjects were first habituated to the behavioral setup. This involved two, 30-minute sessions where subjects were placed in the behavioral setup and allowed to explore the testing environment freely. Subjects were then proceeded to simple food association, where food pellets were manually dispensed into the feeding chamber. Once a pellet was retrieved, the next pellet was dispensed on a random interval of 30 – 45s post-retrieval. Association was tested by a prop trial, where feeding was stopped for one minute and if the subject searched the empty food chamber at least twice then the association between food and the feeding chamber was considered learned.

2.12.2 Sound-food association

This association intended to link the subject's hearing of a reward tone and the dispensing of a reward. This sound tone acted as a bridging stimulus for the delay between the completion of a successful trial and the retrieval of a food pellet and always indicated a rewarded trial to the subject for this stage and all future stages. Without this bridging stimulus learned, it would be much more difficult for a subject to learn a behavioral task because the correct sequence of actions to produce a reward could become ambiguous. This stage had the same timing structure as the simple food association but additionally a speaker played a 4kHz tone for 1s at the same time a reward pellet was dispensed. Association was tested by prop trial, where the tone is played and the time it took for the food chamber to be explored was measured. If the exploration time was less than five-seconds on three separate prop trials, the association was learned.

2.12.3 Knob-touching food association

Once the bridging stimulus is associated, the subject learned to associate touching the manipulandum with food. The knob was placed 2.5cm inside of the behavioral box, slightly past the reaching window, to encourage interaction. To further increase the chances of interaction, researchers would crush a reward pellet and sprinkle the dust onto the manipulandum. A hold of 6ms resulted in a successful trial. In this stage, and all stages hereafter, a successful trial rewards a single reward pellet. After each successful trial, the manipulandum was unable to be interacted with again for 4s, known as the inter-trial period. If the subject obtained at least 20 hits in a single session, the association was learned.

2.12.4 Knob-touching retraction

At the end of a successful knob-touching food association session, both subjects on this paradigm (and two other subjects in an earlier version of this paradigm) preferred to interact with the knob with their snout rather than with their paw. To eliminate this unwanted behavior, the knob was slowly retracted over the course of several sessions. In the knob-touching retraction stage, if the subject performed 20 trials within 5 minutes, the knob was retracted by 0.5cm. Waiting to retract the knob until after the subject prefers to heavily interact with it is to maintain encouragement, as the task became more difficult the further away the knob was placed. The final knob distance was set to the minimum distance required to enforce only paw interactions with the knob, which was 2cm for both subjects. Once the subject was rewarded on 20 trials within 5 minutes using only their paw at the maximum distance, the association was learned.

2.12.5 Knob-holding association.

A picture progression of an example trial of this paradigm is provided in Figure 11. This was the end-point of training used in this thesis for this paradigm.

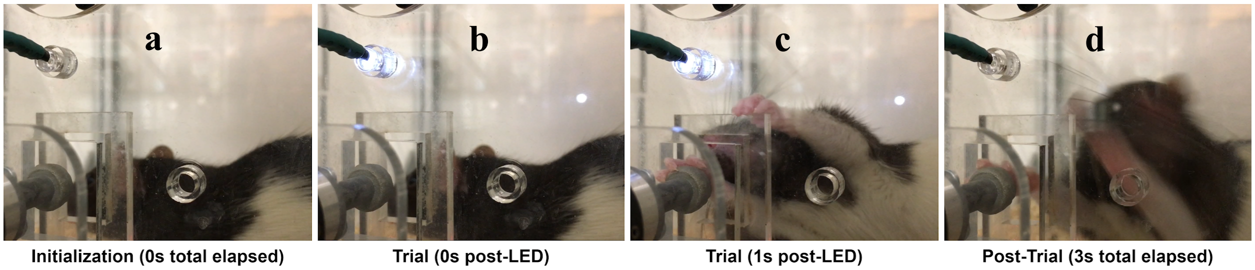

a
b
c
d
Initialization (0s total elapsed)
Trial (0s post-LED)
Trial (1s post-LED)
Post-Trial (3s total elapsed)

Figure 11. An example progression for a single trial of the holding task. a) In the initialization phase, the knob was not available to provide a reward, and would timeout a subject if they touched it. b) The Trial phase began when the cue LED became active and indicated that the knob was available to be interacted with. c) Later in Trial, the subject made a reach and hold of the knob long enough to obtain a reward. d) In Post-Trial, the LED became inactive and in this successful trial, the pellet dispenser fed a single time and a 4kHz reward tone was played (1s). In an unsuccessful trial, no tone was played, and no food was dispensed (not shown).

Following knob-touching retraction, subjects were trained to perform knob holds of increasing duration while also following a rigid trial structure. This stage was necessary to temporally separate the neural activity related to reaching and grasping behavior from any other skilled forelimb behavior. Subjects were rewarded for performing holds that reached a threshold time value. This threshold was adaptive – if the median of the past ten hold times was greater than or equal to the current threshold, then the threshold was increased by 20ms, and if it was less, then it was decreased by 20ms. This ensured that the difficulty of the task scaled with the ability of the subject. The minimum threshold was manually set by the experimenter based on prior subject performance. If the threshold ever went below the minimum value or exceeded 600ms, then it was set to the minimum or 600ms, respectively. In addition to the adaptive threshold, subjects were trained to follow a rigid trial structure, to eliminate some of the natural impatient behavior rats are prone to. This structure was enforced by a white-LED (center LED) above the knob, which turned on to indicate when the knob was available to provide a reward. A successful trial resulted from a single attempt peak hold time of greater than or equal to the threshold, while a failure was caused by a peak hold time lower than the threshold. Either case resulted in the LED deactivating. Following a 4s inter-trial, delay, the LED would reactivate. However, if the knob was touched while the center LED was deactivated, the inter-trial period increased by 4s, and this penalty could be retriggered every 2s since the previous penalty. The combination of the LED-on reward and LED-off penalty ensured that the animal interacted with the knob only when the center LED was active. As an endpoint in this thesis, the goal of training this paradigm was to increase the median hold times of a subject over the course of many sessions. A flowchart showing the possible outcomes for a single trial in this substage is shown in Figure 12.

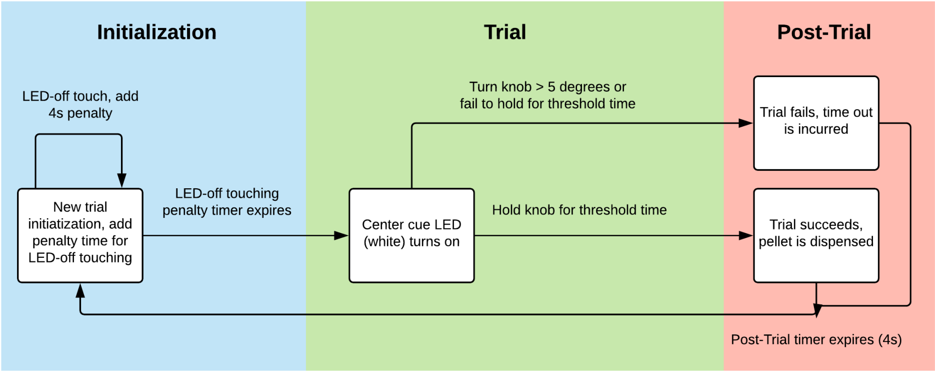

Initialization
Trial
Post-Trial
LED-off touch, add 4s penalty
New trial initiatization, add penalty time for LED-off touching
LED-off touching penalty timer expires
Center cue LED (white) turns on
Turn knob > 5 degrees or fail to hold for threshold time
Hold knob for threshold time
Trial fails, time out is incurred
Trial succeeds, pellet is dispensed
Post-Trial timer expires (4s)

Figure 12. Flowchart of possible outcomes of the static holding task. In Initialization, the subject waited for an LED to turn on to indicate knob availability. If the knob was touched when the LED is off, then a penalty of 4s is incurred (renewable every 2s). When the penalty time resolved, the LED turned on and the subject entered Trial. The subject was required to hold the knob without moving it (less than 5° allowed rotation) until the hold time crossed a threshold. If the subject released the knob prematurely or turned during the holding phase, then the trial failed. A successful or failed trial both entered the subject into Post-Trial, which automatically proceeded back to Initialization following a set Post-Trial timer. The hold time threshold was adaptive based on the median of the past ten trial hold times.

*2.12.6 Cued knob-turning (**not trained in this study**)*

Note: This task was not trained for this thesis but would be considered a possible end-point in training for a subject on this static holding task. Following the holding task the subject would learn to perform a cued knob-turn after a static hold. This stage introduces a green-LED (left LED) and a blue-LED (right LED) which will act as directional cues. Illumination of the left LED indicates a counter-clockwise turn and illumination of the right LED indicates a clockwise turn. Following the 500ms hold in the previous stage, the center-LED will turn off and either the left LED or the right LED will turn on. If the subject performs at least a 5° turn in the correct direction, the trial is successful. Performing a 5° turn in the incorrect direction or releasing the knob without turning will both result in a failed trial. A flowchart for possible outcomes of a single trial of the paradigm is illustrated in Figure 12.

2.13 Paradigm 2: forelimb supination with virtual stiffness

An overview of the substages trained in this paradigm is shown in

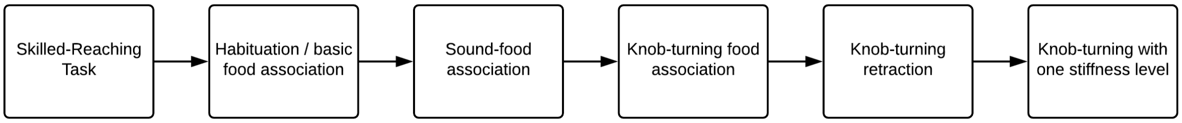

Skilled-Reaching Task
Habituation / basic food association
Sound-food association
Knob-turning food association
Knob-turning retraction
Knob-turning with one stiffness level

Figure 13. Overview of the substages trained in the forelimb supination with virtual stiffness paradigm. All subjects were initially screened for paw-preference and for separating learners / non-learners with the skilled-reaching task. In habituation, subjects were placed in the behavioral boxes and allowed to explore freely. In sound-food association, subjects learned to associate a 4kHz tone with a food pellet being dispensed and going to the feeding chamber, which marked the end of a trial. In the knob-turning food association, subjects were rewarded for performing small turns with either forelimb. In knob-turning retraction, the manipulandum, which was initially inside of the behavioral box, was retracted over several sessions until it was far enough away from a subject that they needed to use a single forelimb to turn the knob. In knob-turning with one stiffness level, subjects learned to perform forelimb supinations of increasing angle against a virtual stiffness.

2.13.1 Habituation / basic food association

See section 2.12.1 in Methods.

2.13.2 Sound-food association

See section 2.12.2 in Methods.

2.13.3 Knob-turning food association (familiarization)

Subjects in the forelimb rotation task were trained to interact with the knob by turning following the sound-food association. This stage, also called knob familiarization, involved placing the end of the knob 2.5cm into the box measured from the box front panel, which was slightly past the reaching window. For further encouragement, pellets were crushed, and the dust was sprinkled onto the knob, so the rat would interact with it. The knob position was initialized at 0°, and a turn of the knob of greater than 2° in either direction resulted in a reward. The inter-trial period was fixed at 4s to limit the rodent from obtaining multiple rewards in a very short period. If the subject obtained 20 rewards within a session, the association was considered learned.

2.13.4 Knob-turning retraction

In the familiarization stage, subjects used various methods to turn the knob (snout, either forelimb, etc.). To standardize behavior, the knob was slowly retracted back towards the front panel. The reward criteria for this stage was based on the subject's forelimb preference determined in the SRT. If a subject preferred their right forelimb, the subject was required to turn the knob clockwise at least 5°. Otherwise, the subject was required to turn the knob counterclockwise at least 5°. In this study, both subjects were right-pawed. The failure condition for this stage was if the subject turned the knob in the wrong direction by at least 5°. In either a rewarded or failed trial, the inter-trial period was 4s.

The knob was retracted by 0.5cm if the subject was able to perform 20 trials within 5-minute intervals. The maximum retraction distance for the knob was based on a position that enforced the subject preferring to use only a single paw. Notably, subjects did not have to have their posture restricted (e.g., physically limiting reaches to either left or right forelimbs) and naturally chose to use only a single forelimb, which matched the forelimb handedness found in the SRT. The endpoint position for both rats in this study was found to be 2cm. Once the subject could perform 20 turns with a single forelimb at the maximum retracted knob distance, retraction was completed.

2.13.5 Knob turning with one stiffness level

A visual overview of this task can be seen in Figure 14. This was the end-point of training used in this thesis for this paradigm.

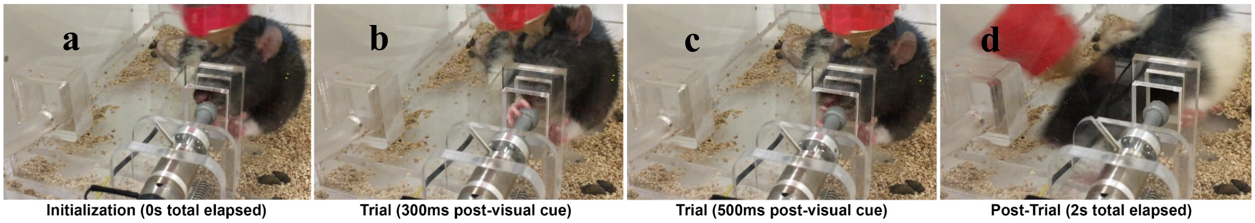

a
b
c
d
Initialization (0s total elapsed)
Trial (300ms post-visual cue)
Trial (500ms post-visual cue)
Post-Trial (2s total elapsed)

Figure 14. An example trial progression of the forelimb supination task with stiffness. a) During the initialization phase, an off-screen LCD panel remained blank until the knob was not turned for 2s consecutively. b) Upon entering a trial, a visual cue appeared on the LCD panel to indicate to the rat that the knob was available to interact with. Within 300ms of the visual cue, the rat began their turn, and ~200ms later (c) the trial was complete. d) Following a successful trial, the subject was given a food pellet as a reward.

Following retraction, subjects were trained to make increasingly large supination attempts at increasing stiffness. Subjects were rewarded for single-attempt knob turns that crossed an adaptive threshold. This threshold was equal to the median of the peak turning angle in the past ten trials. This ensured that the difficulty of the task scaled with the ability of the subject. If the threshold exceeded 60° or went below 5°, then the threshold was set to 60° or 5°, respectively. The thresholding method, including maximum and minimum angles, was identical to that in [20]. Subjects were cued to perform a turn when a large white circle appeared on the 10.1” LCD screen. A single turn was defined as starting when the knob velocity exceeded the minimum velocity threshold (15°/s) and ending when the knob velocity went under the same threshold. Turning the knob when the LCD screen was deactivated would result in a 4 second delay to LCD activation, and this delay could be refreshed every 2 seconds. After stimulus onset, subjects were given a single turn attempt to reach the target threshold. While a turn attempt was active, a motor generated virtual stiffness was also active to counteract the supination attempt. During the adaptive threshold portion of the task, this stiffness was equal to 0.18mNm. Following either a successful or failed turn attempt, the LCD deactivated for an inter-trial period of a minimum of 4s, based on whether the subject interacted with the manipulandum between trials. After three sessions with a greater than 70% success rate on the adaptive task, the subject was moved to the static threshold portion of the task.

In the static threshold subtask, the threshold was fixed at 60° while the virtual stiffness of the manipulandum was increased between sessions. In this study, stiffness torques of 0.18mNm to 1.44mNm were used. All other parameters in the task were the same from the adaptive threshold portion of the task. The goal of using multiple stiffness values was to determine a "calibration curve" of a subject's peak median turning capability at different stiffnesses. These stiffness values would be used to determine the two stiffnesses to use in subtask 2.13.6, which was not trained in this thesis. A flowchart breakdown of the possible progression during a trial for this paradigm is shown in Figure 15.

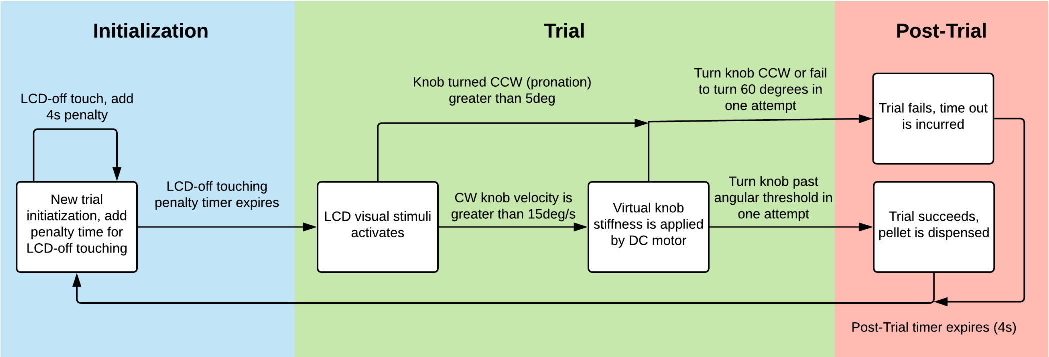

Initialization
Trial
Post-Trial
LCD-off touch, add 4s penalty
New trial initiatization, add penalty time for LCD-off touching
LCD-off touching penalty timer expires
LCD visual stimuli activates
Knob turned CCW (pronation) greater than 5deg
CW knob velocity is greater than 15deg/s
Virtual knob stiffness is applied by DC motor
Turn knob CCW or fail to turn 60 degrees in one attempt
Turn knob past angular threshold in one attempt
Trial fails, time out is incurred
Trial succeeds, pellet is dispensed
Post-Trial timer expires (4s)

Figure 15. Flowchart of possible outcomes of a single trial in the forelimb supination with stiffness task. In Initialization, the subject waited for a visual stimulus to appear on an LCD to indicate knob availability. If the knob was turned (knob velocity > 15°/s) when the visual stimulus is absent, then a penalty of 4s was incurred (renewable every 2s). When the penalty time resolved, the stimulus appeared, and the subject entered Trial. In Trial, the LCD visual stimulus (a salient white circle) cued the rodent to rotate the knob past a certain threshold. A virtual stiffness was applied once the knob velocity exceeded 15°/s. If the subject turned the knob past the position threshold then the trial succeeded and a 4mg food pellet was rewarded and a 4kHz reward tone was played. Otherwise, if the rodent failed to turn the knob the past the threshold or turned the knob in the

opposite direction of the task in a single attempt then the trial failed. Either a successful or failed trial moved the subject into Post-Trial. Following a brief inter-trial delay, Post-Trial automatically proceeded to Initialization. The subject was incrementally trained to turn to the desired target of 60° by initially setting the target threshold to the median of the maximum turn angle of the previous 10 trials.

*2.13.6 Forelimb supination with two stiffness levels (**not trained in this study**)*

Note: This task was not trained for this thesis but would be considered an end-point in training for a subject on this turning task. The final stage for this training was to induce and visualize compensatory kinematics due to changing knob stiffnesses. The criteria for turn angle threshold, reward and failure was the same as with one stiffness level. Here, virtual stiffnesses were applied in 20-trial blocks. In the first block, a lower virtual stiffness would be applied (e.g., 0.72mNm). Then, after 20 trials, a higher stiffness (e.g., 1.44mNm) would be applied, requiring a compensatory response to maintain turn angles above the target threshold. After twenty trials, the stiffness was changed back to the lower level. This cycling continued for the duration of the session.

2.14 Overview of automated system

The automated system described here was used to parallelize and automate our rat behavioral training. The system consists of three major components, 1) a high-level controller, *ratCoach*, 2) a low-level distributed task execution system, *taskRunner*, and 3) a UDP-based data-logging system, *trialLogger*. A schematic of this system is included in Figure 16.

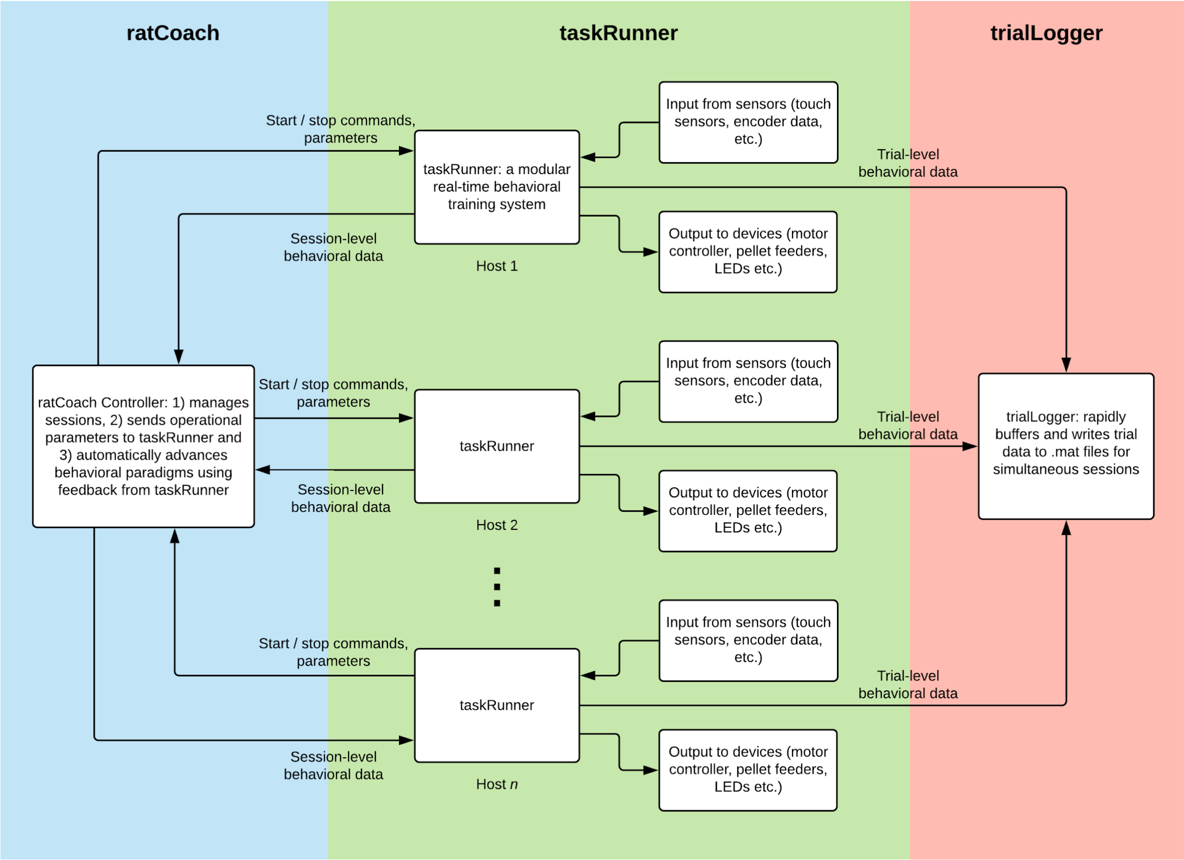

ratCoach
taskRunner
trialLogger
Start / stop commands, parameters
taskRunner: a modular real-time behavioral training system
Input from sensors (touch sensors, encoder data, etc.)
Trial-level behavioral data
Session-level behavioral data
Output to devices (motor controller, pellet feeders, LEDs etc.)
Host 1
ratCoach Controller: 1) manages sessions, 2) sends operational parameters to taskRunner and 3) automatically advances behavioral paradigms using feedback from taskRunner
Start / stop commands, parameters
taskRunner
Input from sensors (touch sensors, encoder data, etc.)
Trial-level behavioral data
trialLogger: rapidly buffers and writes trial data to .mat files for simultaneous sessions
Session-level behavioral data
Output to devices (motor controller, pellet feeders, LEDs etc.)
Host 2
Start / stop commands, parameters
taskRunner
Input from sensors (touch sensors, encoder data, etc.)
Trial-level behavioral data
Session-level behavioral data
Output to devices (motor controller, pellet feeders, LEDs etc.)
Host *n*

Figure 16. Schematic and dataflow diagram between sections of the automated system. ratCoach is a high-level controller that automatically starts training sessions at predetermined times and progresses training paradigms based on previous subject performance. taskRunner is low-level of a behavioral paradigm, where a single instance of taskRunner is linked to a single rodent during training. Each taskRunner setup consists of everything required for one training rig (behavioral enclosures, external I/O, real-time task controllers e.g., Raspberry Pi, NI-DAQs), and sends task-related data back to ratCoach and to trialLogger. trialLogger is a data storage daemon and is made to rapidly buffer and write behavioral data into .mat files.

ratCoach was a behavioral training management system written in Python that simultaneously managed training sessions and automated behavioral training progression based on the past performance of a subject. All the information required to setup an animal for behavioral training (e.g., subject name, name of behavioral paradigm, task-related parameter values, etc.) were accessible via an easy-to-use web application written in Django. Automated behavioral training progression was made possible by analyzing session-level data received back from taskRunner. For example, a rat could be required to obtain a certain hit percentage over a session to progress to either the next task in the paradigm or to increase the difficulty of the current task. In the case of paradigms that were still in-testing, parameters were fine-tuned also using the web interface. taskRunner is a distributed training system that receives high-level task commands from ratCoach to control external devices based on animal behavior during a task. While a single-instance of ratCoach could control distributed behavioral training, the purpose of taskRunner was to run the individualized task for each subject, i.e., each behavioral setup was an instance of taskRunner. Broadly, taskRunner was programmed as a state machine that controlled the necessary code logic and device I/O to train a behavioral paradigm. The main code of taskRunner was written in Simulink using the Simulink Desktop Real-Time Kernel (SLDRT) in MATLAB 2017a. The SLDRT kernel ensured a fixed, real-time sample rate of 500Hz. taskRunner was designed as a "black-box" system, where all the behavioral paradigms were pre-programmed into the Simulink model and ratCoach provided both the appropriate parameters for the specific paradigm to be trained and the commands to start or stop training.

trialLogger was a daemon written in C for Linux distributions by Dan O'Shea [51] that listened for UDP packets sent from taskRunner on a pre-specified port using the associated matUDP communication layer. Using matUDP, trialLogger was triggered in real-time to rapidly buffer incoming packets related to the current behavioral trial and also wrote past trial data into .mat files. matUDP is a bus communication (where in Simulink, a bus is a hard-typed structure) layer, so its goal was to maintain the original bus structure containing organized behavioral data prior during deserialization and regenerated the bus when writing it to file.

2.15 Storage structure and post-analysis of behavioral data

Behavioral data was collected using the matUDP associated trialLogger daemon for all training conducted in this study. The .mat files produced were organized at the time of logging, where the automated system allowed stored subject names to be automatically sent with other task-relevant parameters at the beginning of a training session. The overall structure of a trialLogger saved .mat filepath looked like the following:

Paradigm → Rodent ID → Session Date → Paradigm Revision → saveTag → .mat File

At the top level of the file storage hierarchy was an identifier for the paradigm(s) tested. The next folder was the rodent identifier / name, followed by a session date. While the rodent ID was automatically derived from the ratCoach parameter packet, the session date was automatically applied by trialLogger. At the next level was the paradigm revision number, followed by a 'saveTag', which was a unique identifier of a session number for a specific date. Finally, saved .mat filenames included the subject name, a trial number and a unique timestamp. All of the identifiers in the file structure hierarchy (paradigm name, rodent ID, etc.) were also included as part of a data structure contained within the .mat file. This ensured that if a session needed to be restarted for any reason, even with identical trial numbers or if a filename was somehow altered the timestamp included in the .mat file could be used to organize the data appropriately. An example filepath on a Windows-based machine would look like the following:

.\RATKNOBTASK\ICARUS\2018-09 26\1\saveTag002\ICARUS_1_id000001_time20180926.163141.609.mat

Post-analysis of data was done using a suite of MATLAB scripts. These analysis scripts are included in the associated GitLab repository with this project, included in the Appendix. Additionally, this repository also includes a comprehensive usage document for ratCoach, taskRunner and trialLogger as it pertains to our systems, including: training execution, advanced editing of taskRunner for specific paradigms and large database of troubleshooting tips. Many plots were made using the PlotPub library, created by Masum Habib [52].

3. RESULTS

3.1 Skilled reaching task results

Four female Long-Evans rats were evaluated for this study on the skilled reaching task. Three started training on the SRT simultaneously (M,N,O) and one rat (G) had been trained 4 months prior to the start of the experiment (G). Three rats (G, N, O) were found to be primarily right-pawed. The mean number of sessions for completion of the SRT amongst all subjects was 2 sessions. M did not complete the SRT successfully and was excluded as a non-learner. A summary of these results is found in Table 3.

Table 3. Summary of Reaching Task Results

Subject ID	Age (Months)	Handedness (L/R)	# of sessions to complete SRT
G	8	R	2
M	3	~	~
N	3	R	2
O	3	R	2

3.2 Touch sensor validation

Raw touch sensor data was collected from the MPR121 every 4ms (250Hz). The touch sensor algorithm was validated with preliminary data collection to the static holding task to correctly adjust our algorithm to unexpected MPR121 signal changes (e.g., due to turning, see Section 2.6). While characterizing test data for algorithm development, touch attempts were separated into three categories: 1) holding without turning, 2) holding with turning with the same pre- and post-turn baseline, and 3) holding with turning with different pre- and post-baselines. No trials without turning resulted in a changed baseline pre- and post-hold. Results of applying our algorithm to typical holding and holding / turning attempts are shown in Figure 17.

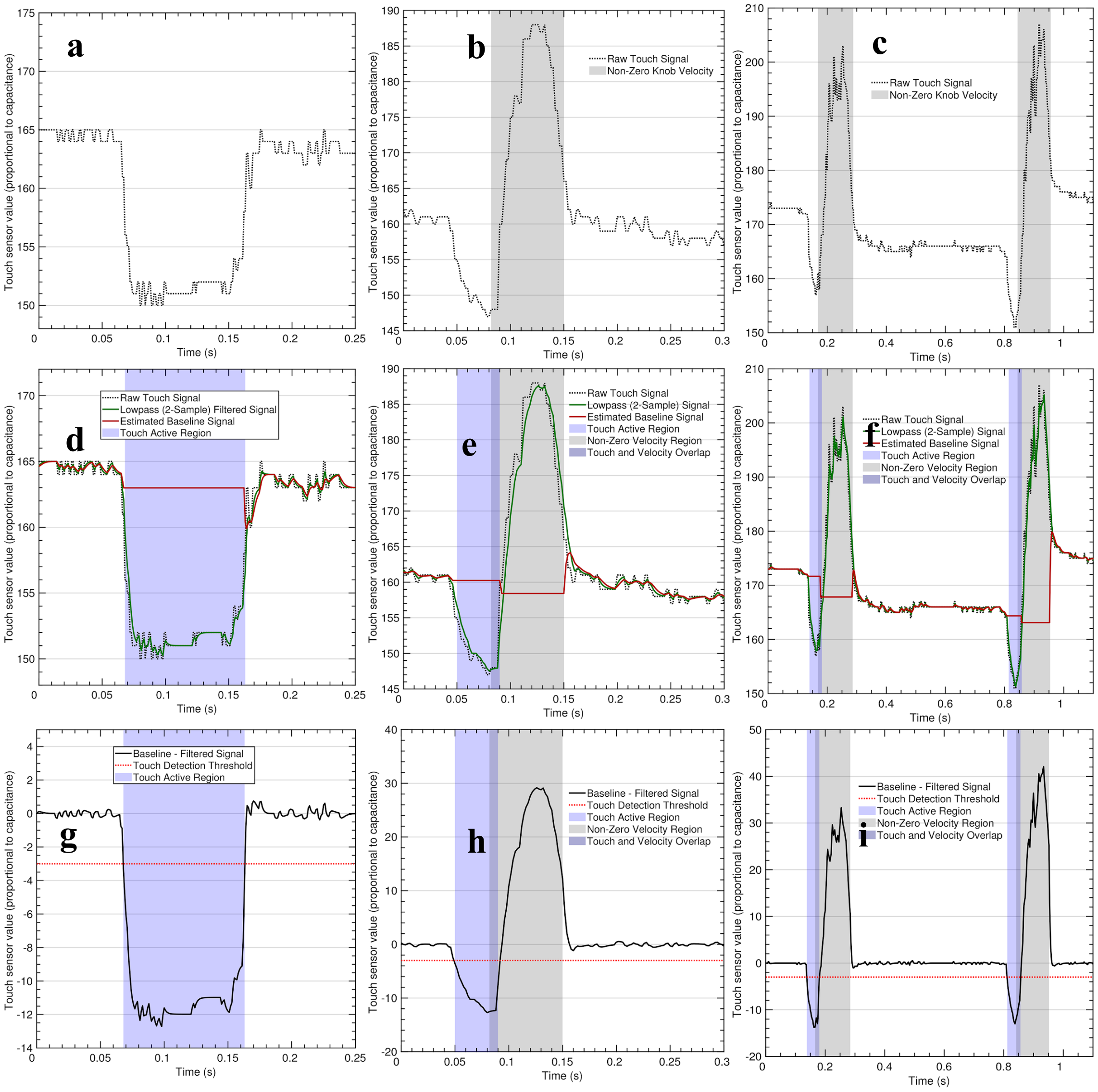


Figure 17. Validation of the touch sensing algorithms on rodent touching and turning data. a - c) show traces for raw touch sensor output in black with periods of non-zero knob rotational velocity are shown in grey as functions of time. The touch signals resulting from a) an ideal knob hold with no turning, b) a knob hold followed by turning of the knob and c), two instances of holds followed by turning and an example of a changing baseline following a turn. In c), there were two instances where the baseline changes value (~0.3s and ~0.95s). d-f) same plots as in a-c now processed with our algorithm, where red tracks the estimated baseline, green is the raw signal with a two-sample lowpass filter applied, and the blue highlighted regions indicate where the rat was detected to have been holding the knob. Grey regions indicate when the knob velocity was non-zero. g-i) show the baseline signal minus the filtered signal from figures in d-f and a static threshold value shown in red. If the subtracted signal crossed the threshold then the knob was considered touched.

The accuracy of the algorithm was manually calculated by assessing individual trials for accurate touch detection. It was found that 96.7% of hold attempts were correctly identified (N = 1102 trials). Algorithm failures were either due to incomplete knob releases or very slow contact speed with the manipulandum (Figure 18).

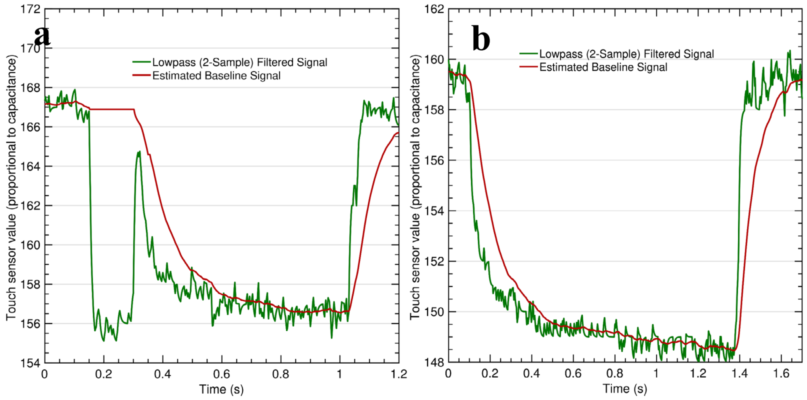


Figure 18. Different hold types that prevented knob holds from being detected. a) Rodents that initiated a good hold followed by an incomplete attempt to release the knob (~0.35s) triggered the release detection portion of the algorithm and caused the remainder of the hold to not be detected. b) Additionally, if a subject was too slow when grabbing the manipulandum, then the grip initiation portion of the detection algorithm failed to register a hold attempt, because it relied on a large rate of change of $touch_{FILT}$.

3.3 Nano17 raw gauge measurements and force / torque conversions

Analog data from six gauges of the Nano17 and I_{MEAS} from the ESCON 24/2 were collected at 500Hz. For steady-state testing (using the protocol described in Section 2.7.1 of the Methods), a total of 1000 trials were used for analysis. Each trial contained Nano17 and I_{MEAS} data for V_{COMM} increased according to Section 2.7.1 (V_{COMM} from 0V to 5V in 0.1V increments). For frequency analysis testing, a total of 1500 trials were used for 32Hz, 16Hz and 8Hz, 1000 trials were used for 4Hz and 2Hz and 500 trials was used for 1Hz. Each trial contained data for a V_{COMM} of 5V, corresponding to a current output of 150mV. V_{COMM} was 20 cycles of a wavelet-like signal for 32Hz, 16Hz, 8Hz, 4Hz and 2Hz and 15 cycles for 1Hz. For both steady-state and frequency tests, trials of the same kind were time-aligned and averaged.

Torques were calculated from Nano17 gauge readings by first zeroing trial-averaged gauge measurements. Zeroing was done by subtracting the average voltage of a 100ms reading of a 0V V_{COMM} included at the start of all trials. These zeroed gauge measurements were then multiplied by a manufacturer supplied calibration matrix, which produced forces and torques in x, y and z. For this study, our interest was only in torques in the Z-axis, which was the same axis as the τ_{COMM}. A summary of these zeroing and conversions steps can be seen in Figure 19.

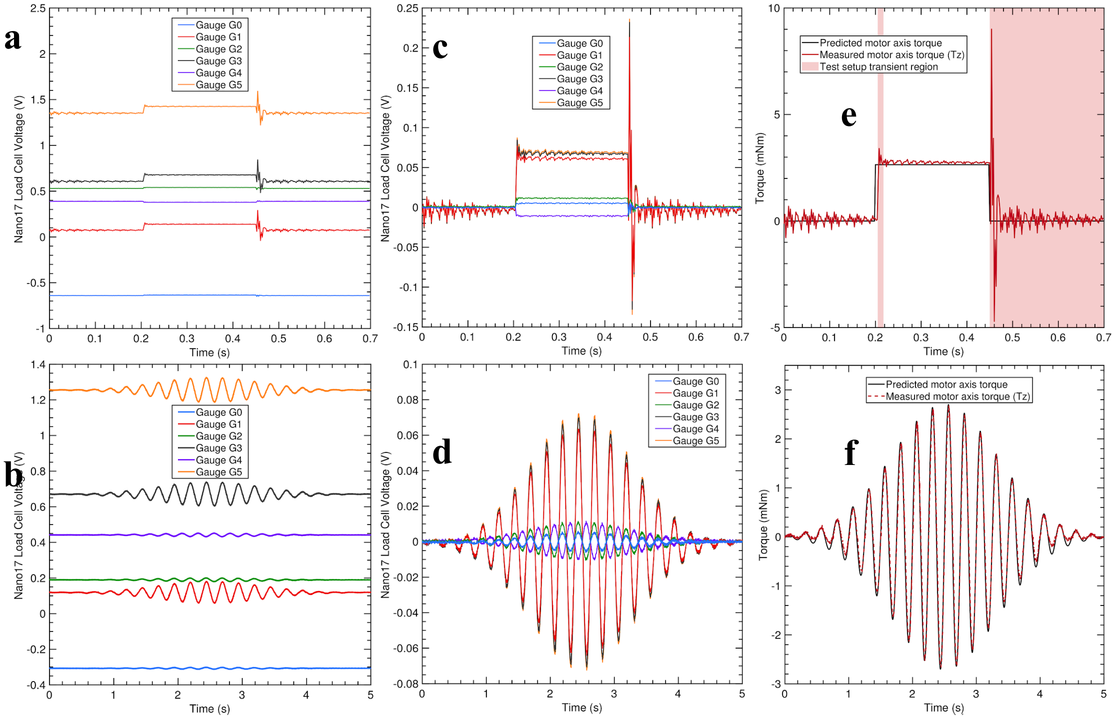


Figure 19. Conversion of Nano17 gauge voltages to torques for both steady state and frequency response analysis. a,b) raw voltages from the six gauges were simultaneously shown for a single desired set of torque commands. Either a) a step input or b) a 4Hz wavelet-like input was used to explore steady-state and frequency response characteristics, respectively. For a) gauge measurements were a result of a 75mA I_{COMM}, and b) was a wavelet-like input with a maximum amplitude of 75mA. Each trace color represents a different gauge on the Nano17. These traces were formed by averaging time-aligned data from repeated experiments (N > 500) to minimize noise. c,d) prior to force and torque conversion, the gauge voltages were zeroed to eliminate offsets. Zeroing was done by subtracting the average of 100ms of V_{COMM} = 0V for each gauge. e,f) Nano17 torque conversions were performed by multiplying gauge values for G0-G5 by a 6x6 calibration matrix. The predicted torque, τ_{PRED}, from current measurements, I_{MEAS} were shown in grey and the measured torque, τ_{MEAS}, were shown in red. In e), an unexpected high-frequency transient torque response was shown by the pink regions.

3.4 Step-response motor torque analysis

Steady-state torque accuracy was first visually assessed by plotting trial-averaged τ_{MEAS} at different levels of τ_{COMM} (Figure 20a). It was shown that unexpected high-frequency transient behavior (overshooting and undershooting oscillations, Figure 20a) persisted for approximately the first 20ms. Additionally, a delay of approximately 6ms was found between the step-up for τ_{PRED} and τ_{MEAS}. Instantaneous measurements of τ_{MEAS} at different times post-current command were plotted together to determine an approximate settling time of the step-response. Empirically, this was determined to be 20ms post-command (Figure 20b), and confirmed by regressing τ_{MEAS} against τ_{COMM} at 20ms, resulting in an $R^2 = 0.9988$ and a regression formula of (Figure 20c):

$$\tau_{MEAS,20ms} = 1.046\tau_{COMM,20ms} - 0.05 \tag{16}$$

The accuracy of the PI-controller for I_{MEAS} was validated by regressing I_{MEAS} against I_{COMM} at 20ms, resulting in an $R^2 = 0.9999$ and a regression equation (Figure 20d):

$$I_{COMM} = I_{MEAS}1.0067 + 0.0003 \tag{17}$$

Next, the average τ_{MEAS} from 20ms – 250ms was regressed against the averaged I_{MEAS} from 20ms to 250ms, resulting in a fit with $R^2 = 0.9987$, and the regression equation given by (Figure 20e):

$$\tau_{MEAS} = 37.29I_{MEAS} - 0.0944 \tag{18}$$

Where τ_{CONST} is provided by the slope of this regression as 37.29mNm/A. The percent error between τ_{CONST} and the specification sheet value (36mNm /A) was 3.58%. (Table 1). The signed error and fractional error for the averaged τ_{MEAS} compared to τ_{COMM} are shown in Figure 20f,g.

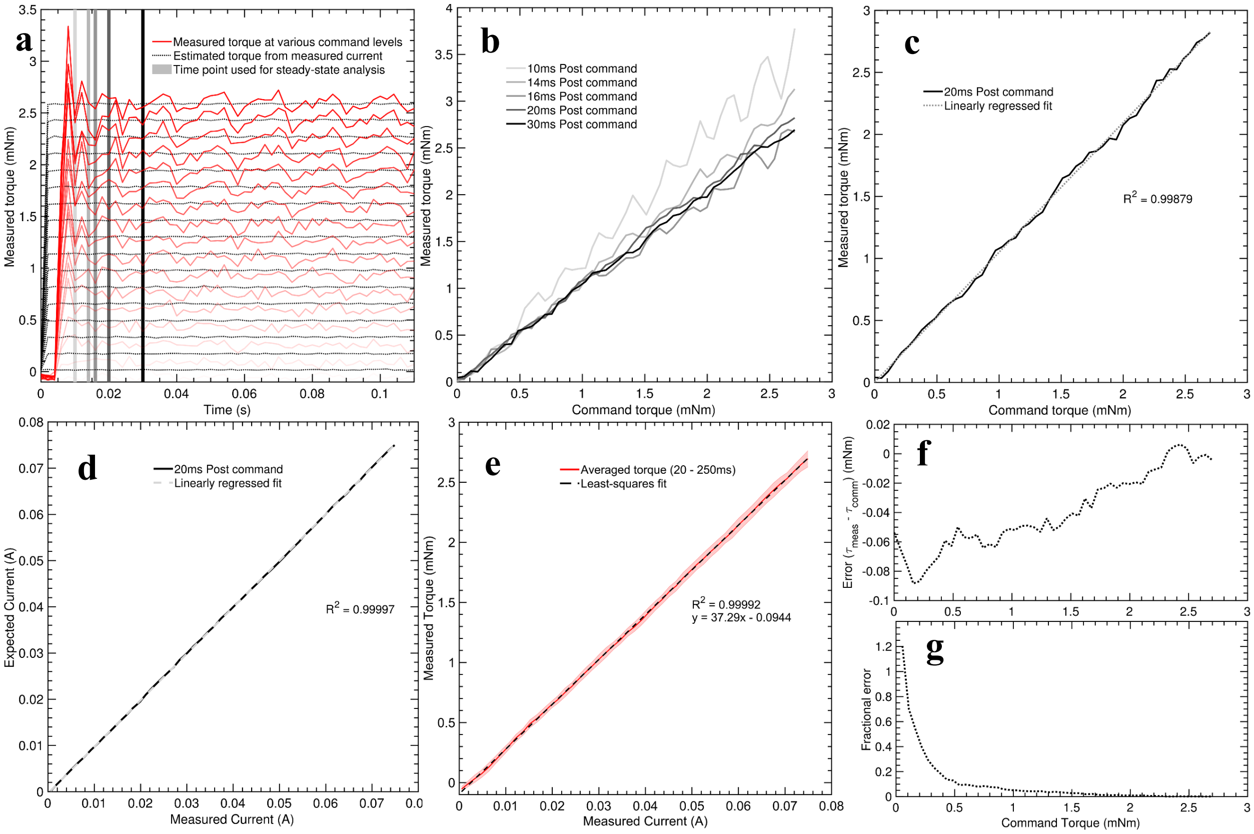


Figure 20. Analysis of the steady state torque behavior of the DCX16S Maxon motor to a step-up torque. a) τ_{MEAS} at various levels of τ_{COMM}, shown as individual red traces. As the τ_{COMM} increased, the traces for τ_{MEAS} were shown as becoming opaquer. τ_{PRED}, estimated from I_{MEAS}, was shown by the black dotted traces. Greyscale highlighted strips show time periods where instantaneous torque measurements were used for b), where more opaque regions are further from step onset, which is time zero. b) Instantaneous torque measurements extracted from a) plotted against τ_{COMM}. c) τ_{COMM} was linearly regressed against τ_{MEAS} at 20ms post command and the line of best fit was shown ($R^2 = 0.9988$). d) The accuracy of the PI-controller on the ESCON 24/2 was tested by regressing I_{MEAS} against I_{COMM} ($R^2 = 0.9999$). e) The averaged τ_{MEAS} from 20-250ms of the step-on period at every voltage increment from 0 – 5V (0.1V steps) was regressed against I_{MEAS} averaged in the same period to compute τ_{CONST} = 37.29mNm/A using the slope of the regression curve ($R^2 = 0.9999$). f,g) Valued errors (f) and fractional errors (g) comparing the averaged τ_{MEAS} to τ_{COMM}.

3.5 Frequency response analysis of the DCX16S

The high-frequency transient behavior in the step-response necessitated following up with frequency response analysis to more closely assess the torque characteristics. In this case, the input signal was the predicted torque from the measured current, τ_{PRED} and the output signal was τ_{MEAS}. For our analysis, both a Bode magnitude and Bode phase plot were produced at the chosen frequencies (1 - 32Hz at 1Hz, 2Hz, 4Hz, 8Hz, 16Hz and 32Hz). Frequency response trial-averaged calculations of τ_{MEAS}, τ_{CURR} and τ_{COMM} for these frequencies can be found in Figure 21.

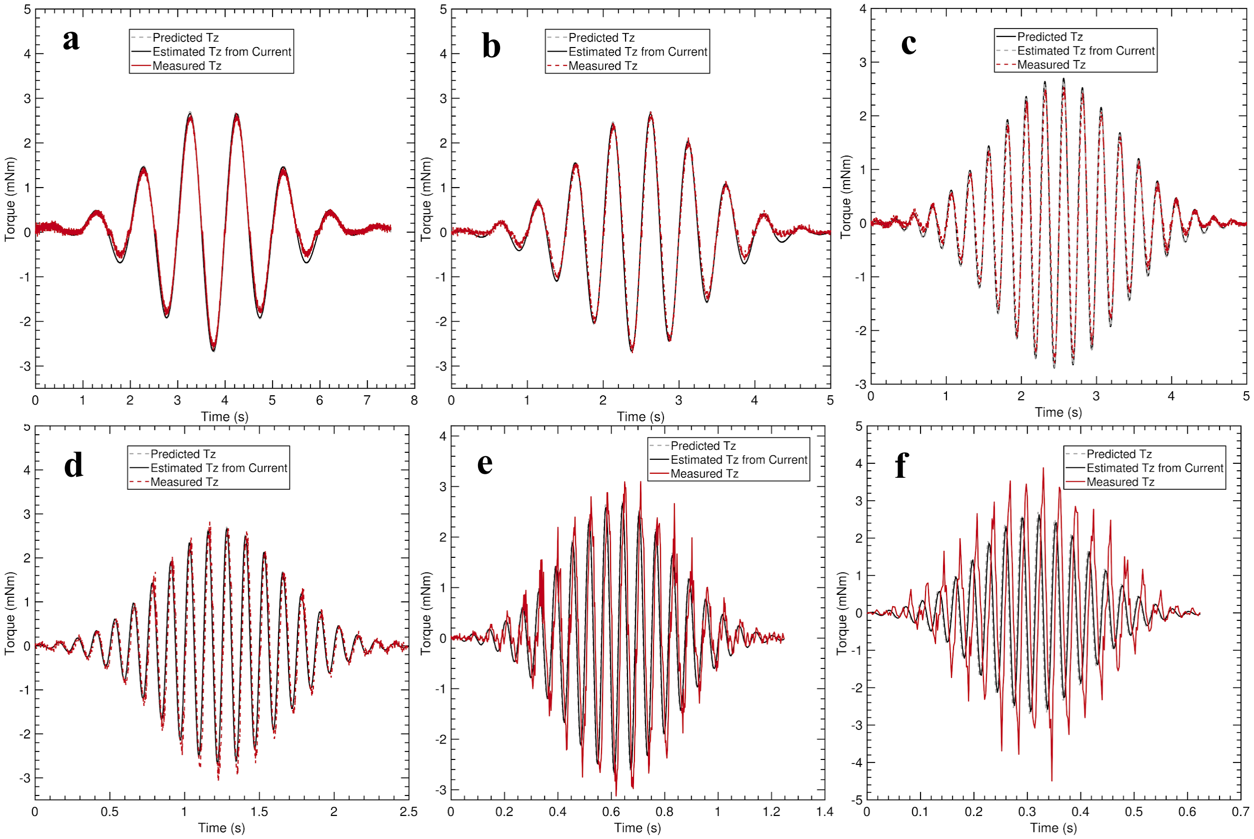


Figure 21. Frequency response analysis raw data of the DCX16S. a) τ_{MEAS} (red) is plotted simultaneously with τ_{COMM} (dotted grey) and τ_{PRED} (solid black) at 1Hz. Testing was completed at six frequency levels, a) 1Hz b) 2Hz c) 4Hz d) 8Hz e) 16Hz and f) 32Hz.

The average phase delay was computed by calculating the cross-correlation between the measured current and torque, and the phase delay was found to increase over the testing frequency range from 6ms to 22ms at maximum. The cross-correlation results are summarized in Figure 22.

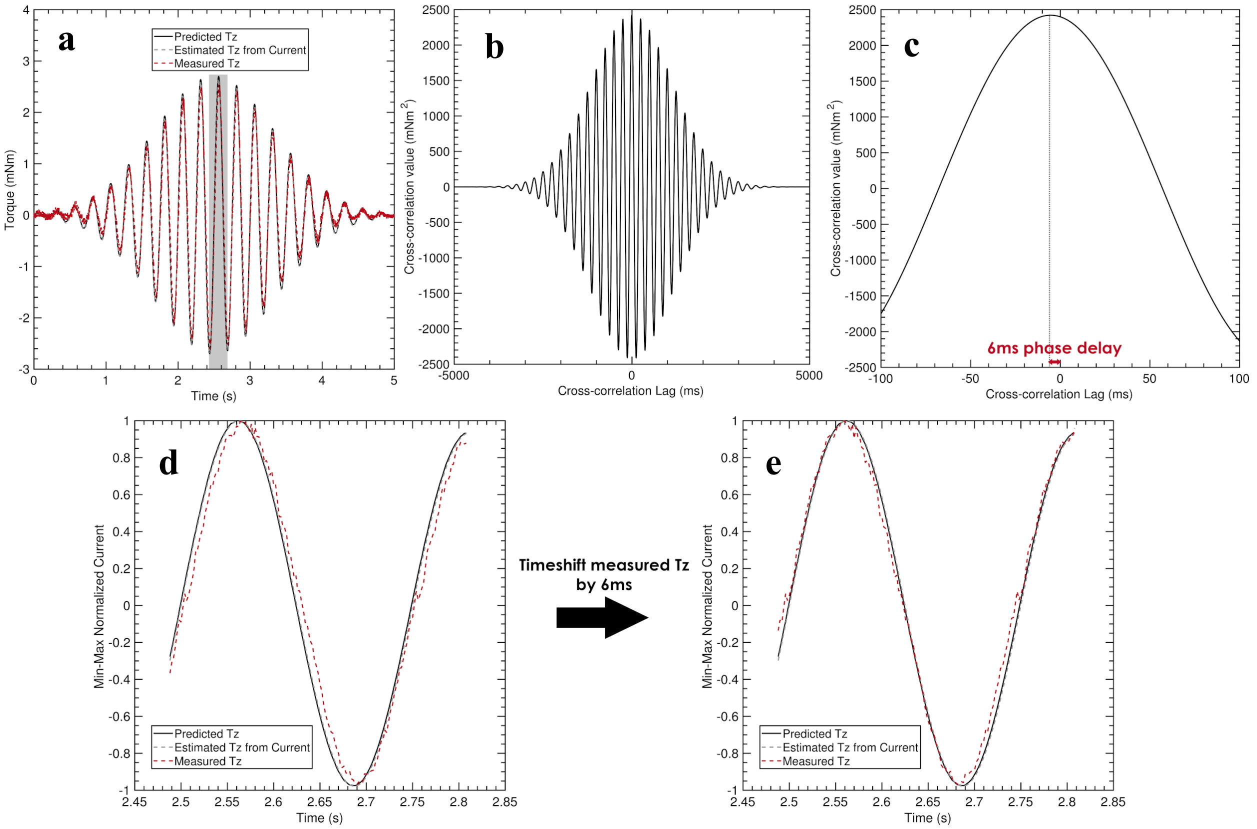


Figure 22. Cross-correlation analysis of the latency between torque command onset and τ_{MEAS}. a) τ_{MEAS} (dotted red), τ_{PRED} (dotted grey) and τ_{COMM} (solid black) for one testing cycle of the wavelet input, V_{COMM} at 4Hz. τ_{PRED} and τ_{COMM} are completely in phase. However, τ_{MEAS} and τ_{PRED} have a non-significant latency compared to one another, which can be seen in greater detail in the highlighted region grey region, which is shown in d). b) The cross-correlation between τ_{PRED} and τ_{MEAS} was computed, which showed an average of a 6ms phase delay between the two signals. c) A magnified plot of the relevant cross-correlation section (around 0ms of cross-correlation) to more easily show the delay between the largest cross-correlation peak and 0. d) A magnified section of a) to show the phase delay more clearly between estimated and measured torques. e) The result of applying a timeshift equal to the measured phase delay to the measured torque to visually assess the accuracy of the change in phase of τ_{MEAS}.

The decibel magnitude changes between $\boldsymbol{\tau_{MEAS}}$ and $\boldsymbol{\tau_{PRED}}$ were calculated using Equation (10) using RMS values computed in MATLAB. The resultant RMS values, decibel gains and phase delays for each input wavelet frequency are given in Table 4. The magnitude changes and phase delays were plotted against the log of the frequency to create a Bode plot shown in Figure 23.

Table 4. Frequency Response Gain and Phase Delay Results using Non-Compliant Coupling.

Wavelet Frequency (Hz)	Estimated Torque RMS (mNm)	Meas. Torque RMS (mNm)	Gain (dB)	Phase delay (ms)
1	1.0665	0.9902	-0.322	6
2	1.0325	0.9859	-0.201	6
4	1.0303	0.9894	-0.176	6
8	1.0297	1.0617	0.133	6
16	1.0258	1.1216	0.388	8
32	1.0154	1.3646	1.284	22

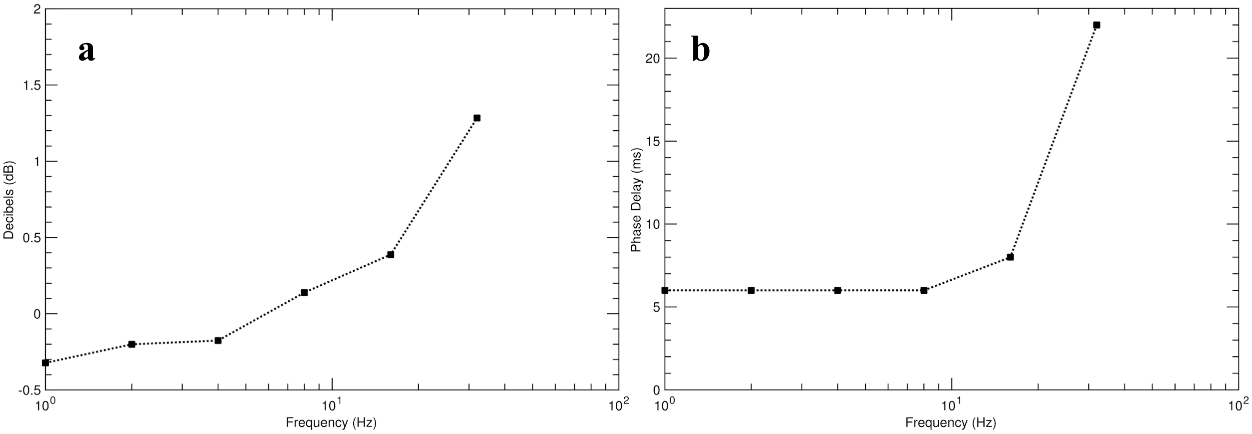


Figure 23. Bode magnitude and phase lag plots for DCX16S torque frequency response characterization. a) The magnitude changes in decibels comparing τ_{PRED} as the input function to τ_{MEAS} as the output function for wavelet commands from 1Hz to 32Hz. The frequency axis is shown in log-frequency. Decibel changes ranged from -0.322dB to 1.284dB. b) For the same input-output functions as in a), the phase lag in milliseconds showed a constant 6ms delay until a 16Hz wavelet input. At 16Hz, the delay increased to 8ms and increased further to 22ms at 32Hz.

3.6 Manipulandum torsional friction and moment of inertia and characterization

Several parameters of our testing apparatus were measured to compute the moment of inertia, radius of gyration and torsional dynamic friction of the manipulandum. Section 2.8 contains all relevant equations for computing I_{KNOB}, k_{KNOB} and τ_{FRIC}. Example plots of the rotational velocity of the knob used to compute the rotational acceleration in each experiment in section 2.8.1 (α_{KNOB}) and 2.8.2 (α_{ROTOR}) are shown in Figure 24. The mean value of α_{KNOB} was found to be 125.19 rad / s^2 ($\sigma = 12.85$), and the mean value of α_{ROTOR} was found to be -2575.9 rad / s^2 ($\sigma = 10.35$). The mass of the manipulandum was determined to be 9.2g, the radius of the knob shaft was found to be 3.25mm. A summary of these findings is shown in Table 5.

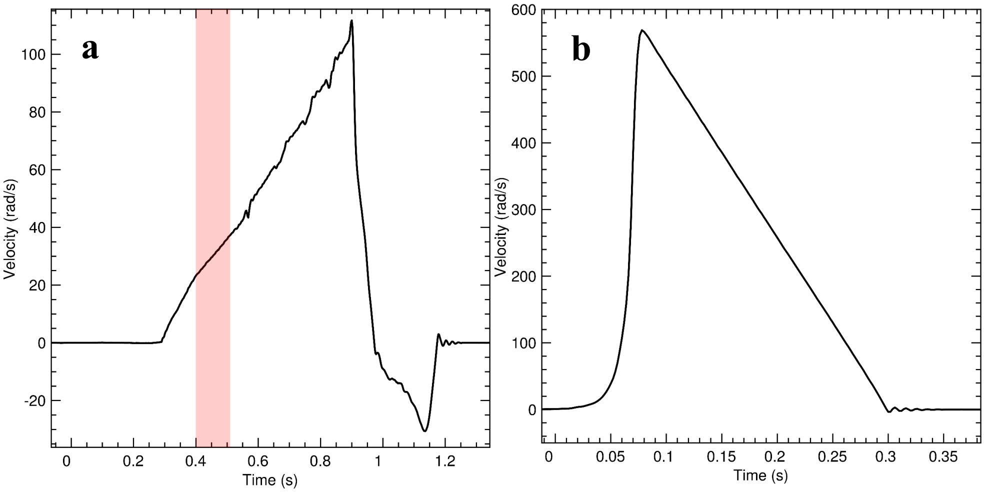


Figure 24. Rotational velocity plots used to estimate rotational acceleration for computing I_{KNOB} and τ_{FRIC}. a) The rotational velocity of the knob due to a weight tied to a string around the knob shaft was shown. The positive velocity region from ~0.3s to ~0.9s was due to the string unwinding and accelerating the knob. The rapid deceleration and negative velocity directly after that was due to the string completely unwinding and decelerating the knob. The slope of the velocity remained largely constant past ~ 0.4s to ~0.9s. The red highlighted region indicated the section of the velocity curve used to calculate the rotational acceleration. b) The rotational velocity of the motor rotor was accelerated using an arbitrary torque and decelerated only by the unknown rotor friction, τ_{FRIC}. The increasing velocity from ~0.05s to ~0.1s was due to the arbitrary torque (applied by spinning the knob between an experimenter's thumb and forefinger), and the roughly linear decrease in velocity was due to the rotor friction decelerating the rotor. Because of the reasonably constant slope during deceleration, an arbitrary region was selected here for each trial to estimate the rotational acceleration.

Table 5. Measured parameters used for DCX16S physical characterization.

Parameter	Value
Measured weight of 10g weight (N)	0.0991
Radius from center of rotation (r_{KNOB}, m)	0.00380
Torque due to 10g weight (τ_{10g}, Nm)	0.00037
Mass of knob (kg)	0.0092
Acceleration of knob with 10g weight (α_{KNOB}, rad/s^2)	125.19
Acceleration of rotor by friction (α_{ROTOR}, rad/s^2)	-2575.9

Using the calculated values for α_{KNOB} and τ_{10g}, I_{KNOB} was computed to be 9.661gcm^2. As a check for the reasonableness of this calculation, k_{KNOB} was also computed with a value of 5.675mm. The friction-deceleration experiment yielded a value for τ_{FRIC} of 0.247mNm using the specification sheet provided I_{ROTOR} (0.96gcm^2, Table 1) and the roughly constant α_{ROTOR} (Table 5). These computed parameters are summarized in Table 6.

Table 6. Computed parameters of DCX16S motor with attached manipulandum.

Parameter	Value
Moment of inertia of knob (I_{KNOB}, gcm^2)	9.661
Radius of gyration of knob (k_{KNOB}, mm)	5.675
Torsional dynamic friction of knob (τ_{FRIC}, mNm)	0.247

3.7 Static holding paradigm

Subjects (N = 2) were trained to perform the paradigm provided in the Methods from 2.11.1 through 2.11.5. Both subjects in this task preferred their right paw. The mean number of trials performed was 91.5 trials by Subject O (σ = 42.3) and 130 trials for Subject N (σ = 40.08) per session.

Over approximately two weeks, Subject O and N were trained over 8 and 10 sessions, respectively, to perform static holds. For Subject O, the median of the maximum hold time during a session went from 40ms (IQR = 39.5ms) in the 1st session to 391ms (IQR = 502ms) in the 8th session ($p < 0.001$). The median of the hold time threshold went from 6ms (IQR = 0ms) in the 1st session to 275ms (IQR = 140ms) in the 8th session ($p < 0.001$). Similarly, for Subject N, the median of the maximum hold time went from 72ms (IQR = 287) in the 1st session to 456ms (IQR = 537.5ms) in the 10th session ($p < 0.001$). The median of the hold time threshold went from 66ms (IQR = 120ms) in the 1st session to 400ms (IQR = 200) in the 10th session ($p < 0.001$). The raw data and summary plots for the holding task are included in Figure 25.

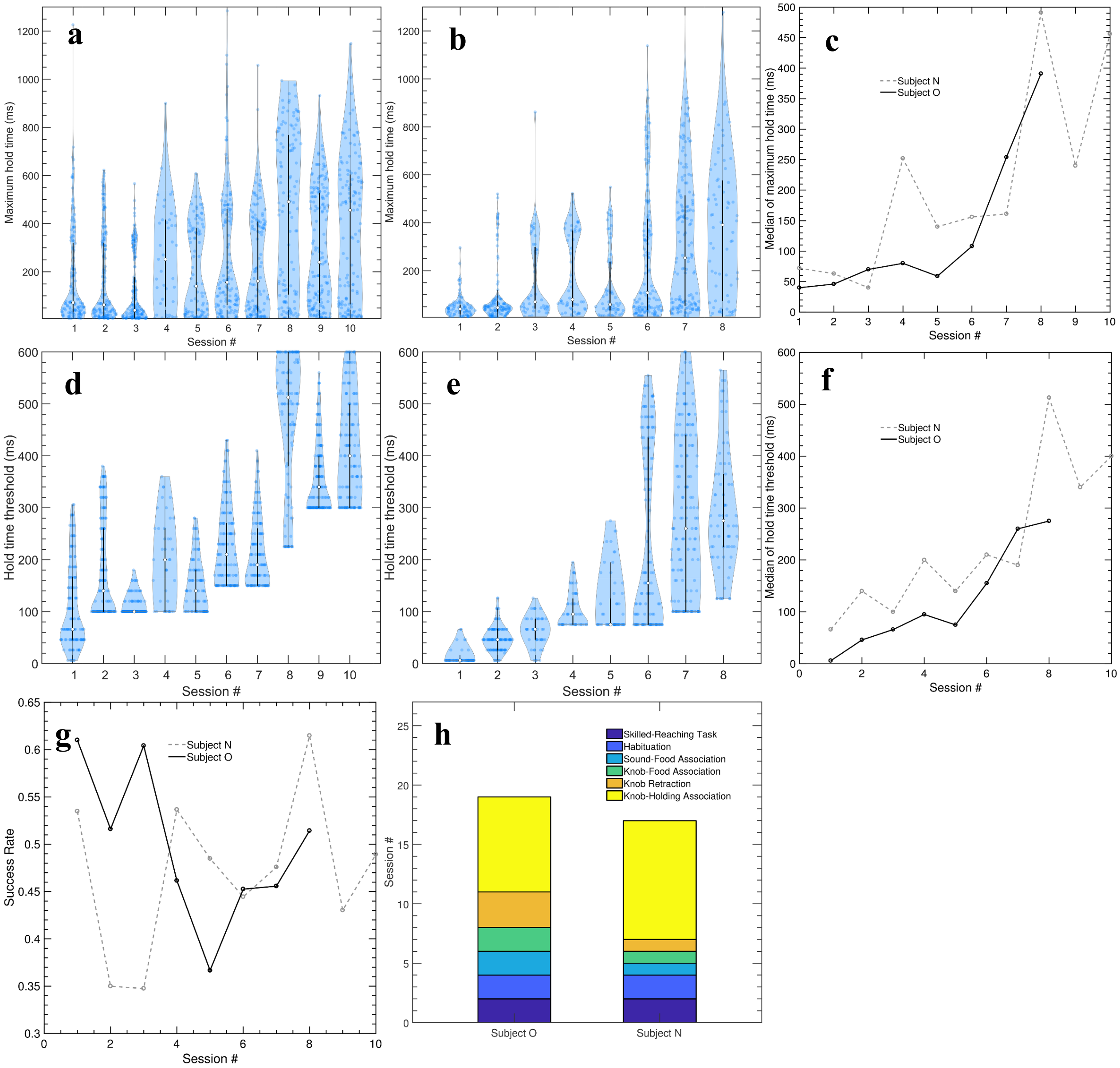


Figure 25. Results of the static holding task in two subjects. The maximum hold times of a) subject N and b) Subject O over 10 and 8 sessions, respectively. The violin plot is a combination of a boxplot and a visualization of the distribution of the raw data. The median of a session is shown with the white dot and the interquartile range is shown with the thick black bar. Raw data points are shown as dark blue dots and a rough distribution of data is shown in light blue. c) The median of each session is plotted to show task progression. The same analysis is performed for the hold time thresholds in figures d-f. g) Successful trial rates are shown for both subjects over their training sessions. h) Shows the time required for each subject to move through the separate sub-stages of the behavioral paradigm. Each block indicates which sessions and the number of sessions needed for the association.

3.8 Forelimb supination with stiffness paradigm

A single subject (G) was trained to perform the paradigm outlined in section 2.12 up to 2.12.5 (forelimb supination with one stiffness level). This subject had been previously trained on and off for the adaptive median-threshold turning task for this study and for preliminary testing. An example of Subject G performing the rotation with stiffness (2.13.6) task is shown in Figure 14. The mean number of trials performed in 2.12.6 with one stiffness was 252.6 trials ($\sigma = 92.73$).

Subject G required 3 sessions to obtain a 70% hit rate with 0.18mNm command torque. Over the following six sessions, increasing stiffnesses were tested to determine whether the changing stiffness had a noticeable effect on rodent behavior. During these six sessions, the threshold was fixed at 60 degrees. The median of the maximum angles turned during a session trended downward as the command torque was increased ($R^2 = 0.91$), with a maximum at 0.18mNm of 83.31° (IQR = 51.79°) and a minimum at 1.41mNm of 44.02° (IQR = 47.03°). Similarly, the hit rate showed a general decline as the command torque increased, where the hit rate was 71.2% at 0.18mNm and 34.7% at 1.44mNm. The raw data and a summary plot of the stiffness testing can be seen in Figure 26. The mean peak turning angle, velocity and torques at 0.18mNm were compared to the same parameter at each other stiffness level using one-way ANOVA and post-hoc Tukey HSD. Peak turning angles at 0.72mNm ($p = 0.0075$), 1.08mNm ($p < 0.001$) and 1.44mNm ($p < 0.001$) differed significantly from 0.18mNm. Similarly, peak velocities differed from 0.18mNm at 0.72mNm ($p=0.0035$), 1.08mNm ($p < 0.001$) and 1.44mNm ($p < 0.001$). For peak net knob torques, differences were significantly different from 0.18mNm at 0.72mNm ($p < 0.001$) and 1.08mNm ($p < 0.001$).

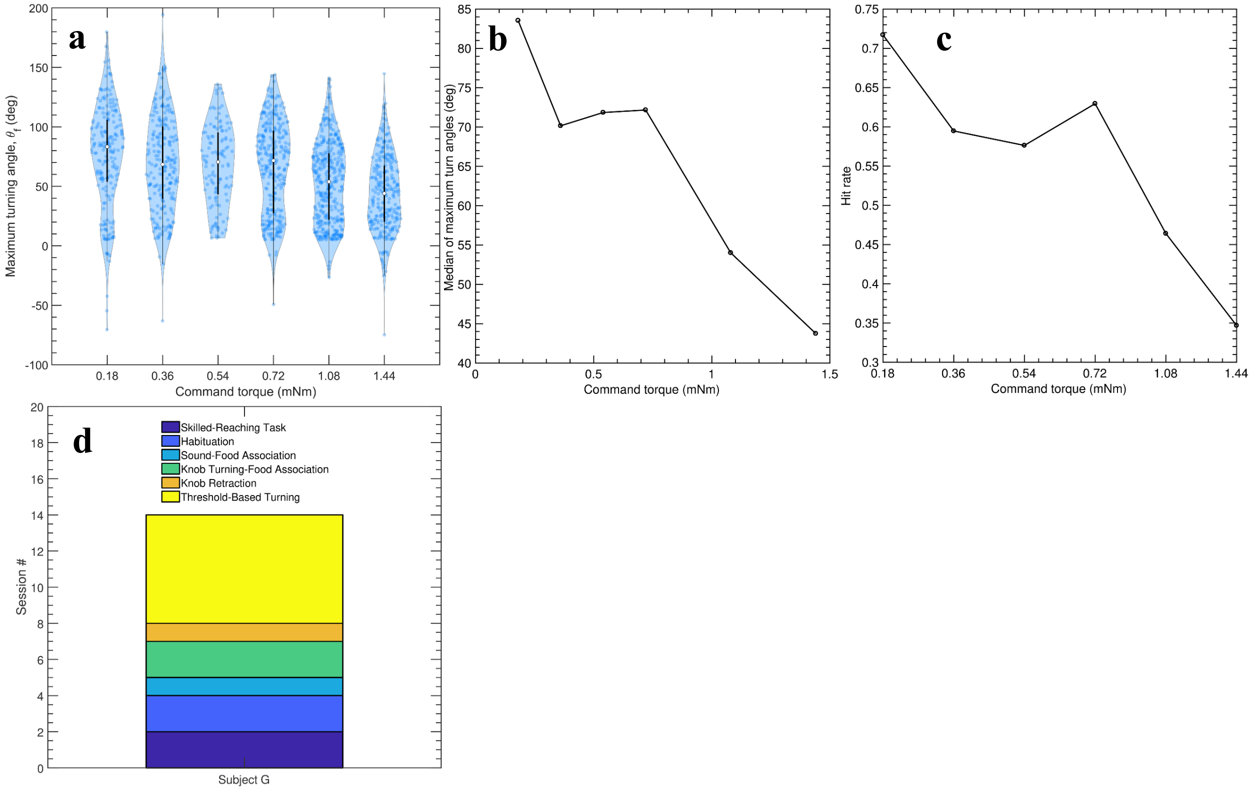


Figure 26. Results of testing various motor stiffnesses on turning behavior. a) Violin plots for subject G's maximum turning angle achieved in trial over six sessions with varied commanded motor stiffnesses, from 0.18mNm to 1.44mNm. Each session delivered only the torque specified on the x-axis to the manipulandum. In each violin plot, the median angle is represented by the white dot and the interquartile range is shown by the thick black bar. The individual data points are shown in dark blue and the estimated distribution based on the raw data is shown in light blue, where a thicker region indicates a higher density of data. b) The median of the maximum angles turned are shown as a function of command torque, and a similarly relationship to command torque is shown for hit rate over the six sessions in c). d) Shows the total number of sessions spent in each of the six sub-tasks in training the skilled turning paradigm, where each block represents the total number of days for that sub-task. Time flows from Session 0 to Session 14.

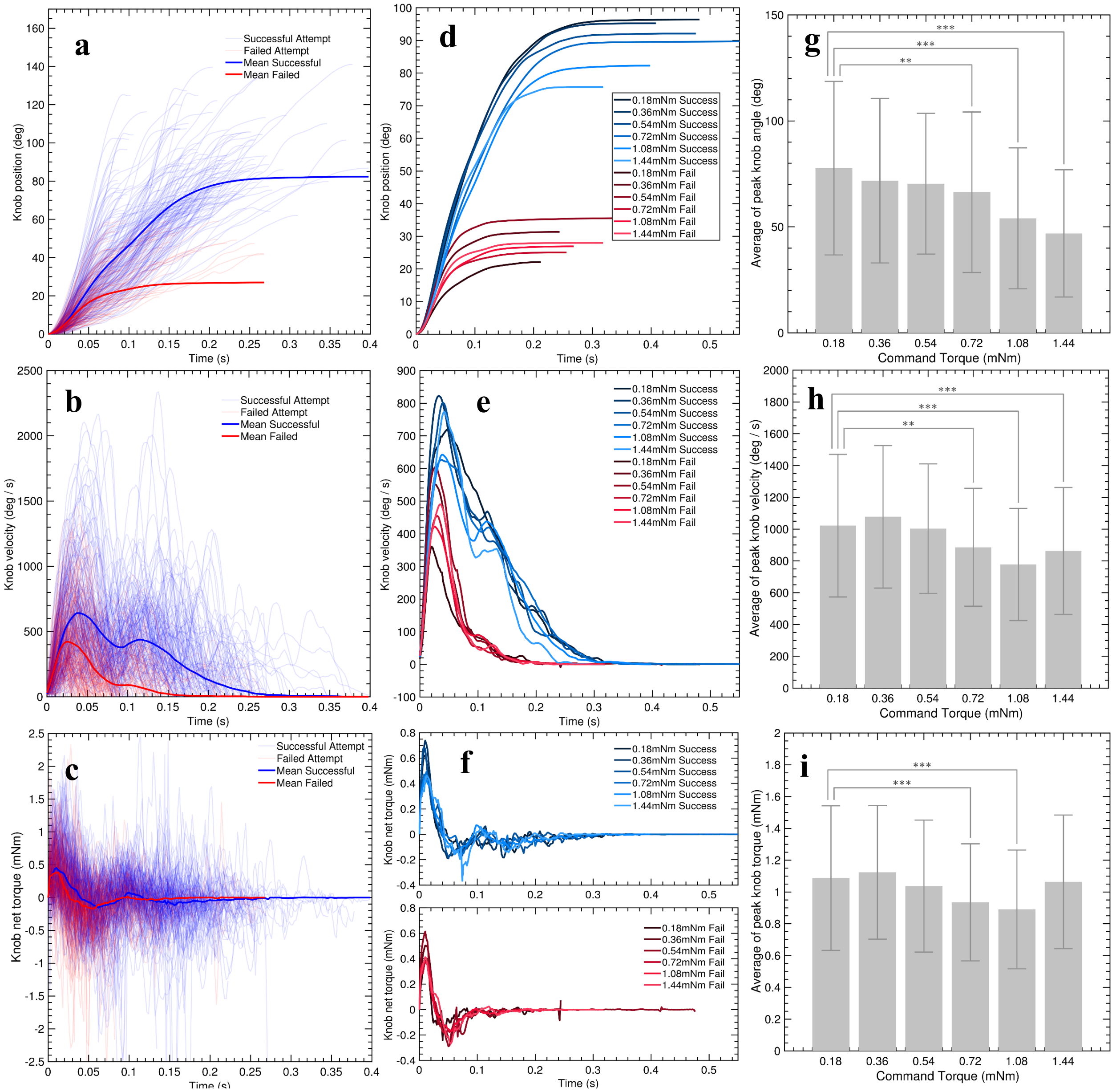


Figure 27. Analysis of the knob turning kinematics at varied stiffness levels. a) The position traces during the 0.72mNm command torque session were time-aligned at turn-initiation and split into successful (blue) and failed (red) turning attempts (N = 305 trials). The traces were shown to end at the completion of a turning attempt for visibility but have real values past what is shown. The position traces in the successful and failed trials were averaged and the resultant trace is plotted in dark blue and dark red, respectively. b) and c) show the same analysis done with knob velocity and torque, respectively, using numerical derivatives. Torque was computed by multiplying the known I_{KNOB} and experimental rotational acceleration. d-f) show the same analysis as done in a-c), but only show the averaged traces for successful and failed trials and include all six stiffness levels tested. g) The peak turning angles for the 0.18mNm session were statistically compared to each other session using a one-way ANOVA with post-hoc Tukey HSD (α=0.05). Means of the peak turning angles are shown with standard deviations as errors. The same analysis was performed for velocity and torque in h-i. **= $p < 0.05$, ***= $p < 0.001$.

Lastly, torques generated by the rat forelimb were estimated by substituting known values into Equation (11). The data for τ_{RAT} is plotted against both time and knob position in Figure 28.

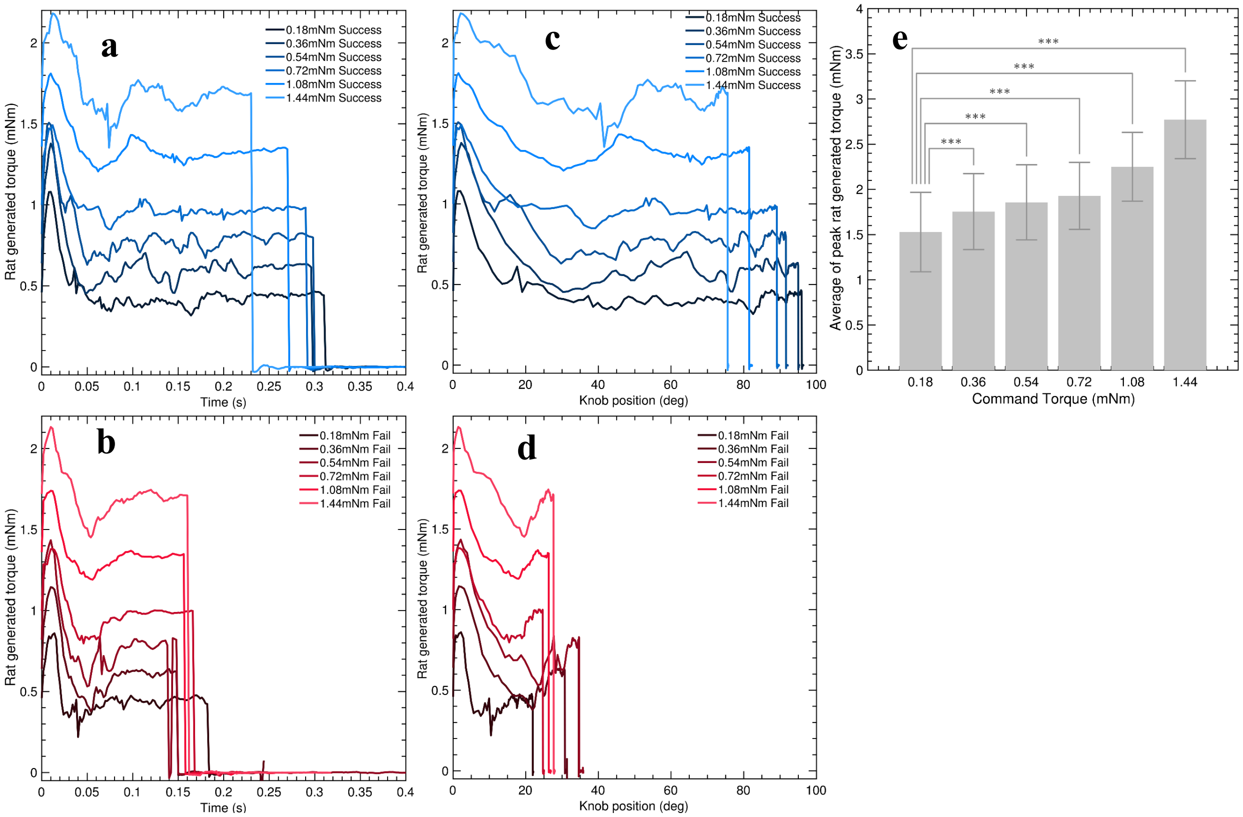


Figure 28. Rat forelimb generated torques, τ_{RAT} during the supination task at varied stiffness levels. a) Time trajectories of the torque generated by subject G for six different sessions at increasing knob stiffness are shown in blue, where lighter blue traces indicate a greater motor stiffness. b) shows the same time trajectories, but for failed trials. c) and d) were similar to a) and b) but instead show the relationship between the forelimb torque and the knob position. e) Statistical comparison (one-way ANOVA with post-hoc Tukey HSD, α=0.05) on the peak averaged rat generated torques for each of the six stiffness conditions, including both successful and failed trials together. Pairwise comparisons were only evaluated comparing 0.18mNm to each other condition, and all comparisons were significant ($p < 0.001$). *** = $p < 0.001$.

4. DISCUSSION

The goal of this thesis was to develop and characterize a one-dimensional rodent training manipulandum that could assess forelimb holding and turning behavior while also having the capability to precisely deliver a virtual stiffness to resist rotation of a manipulandum. Following a thorough characterization of both our touch detection system and torque delivery capabilities using a DC motor, we validated our device using two behavioral training paradigms. The first paradigm assessed the compensatory response of the subject when the stiffness of the manipulandum was changed during a training session. The second paradigm trained subjects to reach and perform a static hold of the manipulandum for increasing periods of time. Ultimately, we hope that this work can inform future studies by showing that the rodent is an effective model to study motor cortically driven tasks and may be a suitable model to explore the existence of motor cortical dynamics.

4.1 The skilled reaching task is useful for pre-training screening test

All subjects used in this study were able to complete the SRT within 2 total sessions (Table 3), with one subject being unable to complete the SRT in the 3 sessions allotted. Typically, the SRT is used as a standalone motor learning behavioral task and not as a precursor to other types of motor training [35][53][18] and is evaluated by both the number of success rate of reaches for pellets. In this study, we were only interested in the number of reaches in a session to get subjects familiar with reward pellets, determine paw preference and identify non-learners. Comparatively, the number of sessions to see progress was on par with what was shown in [53], where subjects made inter-session progress within 1-4 training sessions. The SRT was accurate in determining which paw the subject would prefer after being placed on their respective behavioral task and could be useful in future studies where paw preference might need to be strictly enforced using physical barriers. Additionally, all subjects in the ‘learner’ category were able to perform their assigned behavioral task (to varying degrees), while the 'non-learner' subject was unable to complete the sound-food association. Thus, the SRT is an effective tool for categorizing subjects prior to behavioral training and getting preliminary information on future motor learning potential.

4.2 The touch detection algorithm is robust and accurate

Capacitive touch sensors are effective, low-cost options for low-latency detection of interactions with an electrically conductive manipulandum. Importantly, all capacitive touch sensor detection methods for skin contact rely on accurate measurements of capacitive changes. In this thesis, detection was further complicated by the fact that a slip ring provided electrical contact by acting as a brush between the manipulandum, motor shaft and the touch sensor. An MPR121 capacitive touch sensor was first evaluated for its ability to visually show enough differences between 'touched' and 'non-touched' states for a detection algorithm to be useful. Then, based on these output signals, we developed a novel algorithm that was able to handle unexpected capacitive signal changes due to turning behavior of the knob.

Raw touch sensor output signals clearly showed a change between a 'touched' and 'non-touched' state (Figure 17a-c). The expected change in a capacitive touch sensor with skin contact is a decrease in the signal, which aligns with our findings. This also served as validation that touch detection using our custom-made slip rings would be feasible. The average standard deviation of the filtered sensor signal during non-touch behavior was low (1.91 units, N = 1102) compared to the typical decrease in signal turning a touch which was on the scale of 15-20 units, which provided an adequate signal-to-noise ratio (Figure 17a-c).

The touch sensing algorithm used was developed to handle unexpected positive increases in the raw signal when the knob was turned. Standard algorithms for touch sensing are typically threshold-based only: a baseline signal (i.e., the sensor signal when a touch is *not* present) and a filtered-version of the raw signal are subtracted from one another to produce a difference signal. If this difference signal exceeded a threshold then a touch was detected. Here, our algorithm used both a threshold and the rate of change of the sensor signal (Equation (2)) to detect touches to combat the capacitive changes induced by turning the knob. A full rationale for algorithm development is included in Section 2.6. Following a search of the literature, capacitive measurement changes following knob turning do not appear to be an issue with any other capacitive touch systems that involve a slip ring. Further, upon testing a new set of motors with an identical setup for touch sensing, these motors did not have the same capacitive changing problem. The motors used in the static touch task (48V RE25) were not the same as the motors used for stiffness training (DCX16S) and were obtained secondhand. It is possible that these RE25 motors were somehow physically damaged, leading to unstable capacitive behavior upon rotation of the motor shaft. Even with these detection difficulties, the algorithm was able to discriminate touching behavior in both trials with and without knob turning with 96.7% of touches accurately detected (Figure 17d-i).

Regarding incorrectly detected trials, the two cases were shown in Figure 18. The case shown in Figure 18b is a clear misdetection caused by the rate of change of the touch signal at the touch onset being too slow. Future iterations of the algorithm will likely have to be adjusted for these kinds of holds by changing the threshold of the $touch_{FILT}$ rate of change. The case shown in Figure 18a is correctly detecting the knob release between the two hold attempts and therefore does not detect the second hold. However, if the researcher wishes to count this as a single hold, then the time constant of the baseline filtering should be decreased so that the baseline resets quickly enough between the hold attempts to properly register the second attempt. Overall, the detection capabilities of the algorithm were effective for training static holding behavior but certainly could be improved in future iterations.

4.3 Behavioral device characterization

Prior to training any animals with the behavioral system, the torque generation capabilities of our motors needed to be characterized. Here, we tested the step / steady-state response and frequency-response characteristics of the DCX16S. Step-response testing was able to accurately characterize the steady-state behavior of the system compared to manufacturer specifications but also was confounded by unexpected transient torques on I_{MEAS} step-up. To supplement the step-response data, frequency response testing revealed important phase and magnitude information about appropriate frequencies for I_{COMM} and future steps to validate higher frequencies.

4.3.1 The DCX16S has predictable and accurate steady-state characteristics

For the behavioral paradigms evaluated here, our goal was for the virtual stiffness to be as indistinguishable as possible from a real stiffness, to eliminate unwanted behaviors from being trained. For example, the virtual stiffness should not have unwanted transient torques for I_{COMM} of reasonable frequencies and have accurate steady-state characteristics compared to manufacturer specifications. Theoretically, the current-torque relationship for an electric motor dictates that the output torque of that motor should be linearly related to the input current multiplied by a torque constant, where the torque constant is given in a motor’s specification sheet. However, the real relationship can be more complicated. Among other factors, torque generation inaccuracies can be the result of a poorly controlled motor current, or a motor may not be accurate to specifications. Additionally, torque measurements can be confounded by factors such as testing rig resonance or a poor coupling between the motor shaft and the measurement device. A major effort here is to identify where actual motor behavior deviates from theoretical expectations and whether those deviations are due to torque production or measurement.

Using our step-up torque paradigm, we first assessed the steady-state characteristics of the current and torque measurements compared to expected values. The current controller accuracy was assessed by regressing the I_{MEAS} against I_{COMM}. The slope of this regression was expected to be 1 and the offset was expected to be zero for a perfect input-to-output current relationship. The real values were found to be very accurate (Equation (17)), where the slope was 1.0067 and the offset was 0.0003. Additionally, there was zero phase lag between the two signals. Therefore, the errors between expected and measured values were low enough to conclude that practically speaking I_{MEAS} and I_{PRED} were identical, and thus the ESCON 24/2 was an extremely accurate current controller. This effectively allows us to rule out the ESCON 24/2 as a confound for any downstream inconsistencies in torque production.

Next, we analyzed τ_{MEAS} of the DCX16S as a response to a step-up I_{COMM}. It was determined by visual assessment that steady-state τ_{MEAS} occurred at approximately 20ms post-command (Figure 20b). This was validated by regressing the instantaneous τ_{MEAS} at 20ms post-command at all levels of V_{COMM} (0 -5V in 0.1V increments) against the corresponding τ_{COMM}, which resulted in an $R^2 = 0.999$ with a slope of 1.046 and offset of 0.05 (Equation (16)). At steady-state, the slope should be approximately 1 and the offset 0 in an ideal situation, so this is likely into the settled time. A possibility for this long settling time is a result of the high-frequency step-up I_{COMM} causing unexpected transient torques in the testing rig. This problem is explored further in the frequency response analysis, in Section 4.3.2. The torque constant of the motor was empirically determined by regressing the average τ_{MEAS} against the averaged I_{MEAS} from 20ms to 250ms (the duration of the step-up period). Here, the $R^2 = 0.999$ and the torque constant was found to be 37.29 mNm/A. Two conclusions can be made from this: firstly, given the large R^2, the expected linear current-torque relationship is accurate. Secondly, because this is only a 3.58% error compared to the specification sheet for this motor (36mNm/A), the motor's steady-state torque production is very accurate. The manufacturer does not provide acceptable tolerances for the torque constant to compare to, but this percent error is acceptable for this study.

Lastly, the errors between the averaged τ_{MEAS} and τ_{COMM} were assessed to determine useful ranges for commanded stiffness (Figure 20f, g) using our system. According to Figure 20f, the magnitude of the measured error largely decreases as the commanded torque increases, with a maximum error of 0.087mNm and a minimum error of 0.008mNm. Given the theoretical current-torque relationship, it might be expected that the error would be a constant value due to manufacturing tolerances. Thus, a changing error could be a real phenomenon of the motor (e.g., non-linear torque production at lower currents, or unpredictable static friction interactions) or it might reflect more accurate torque measurement at higher torques with the Nano17. Regarding the percent error, the first time a value of below 10% occurs when τ_{COMM} is equal to 0.54mNm (Figure 20g). However, the magnitude of the error remains relatively constant (0.05 – 0.08mNm) from 0 to 0.54mNm. Therefore, using a virtual stiffness of around 0.54mNm as a minimum value is likely acceptable. In this study, we used lower than 0.54mNm (lowest τ_{COMM} of 0.18mNm) to test the effect of lower torques, discussed in Section 4.5.

4.3.2 Frequency-response motor characteristics

While the step-response was an effective way to assess the steady-state behavior of our system, the high-frequency input current (250Hz) also caused unexpected transient behavior directly following the step-up command. As seen in Figure 20a, the amplitude of τ_{MEAS} showed large amplitude overshooting and undershooting around τ_{COMM}. Practically speaking, large oscillations following current step-up could cause the knob stiffness to feel unpredictable to a subject during a turning task and could be detrimental to training consistent behavior. Further, these transients make determining the time delay between the onset of I_{COMM} and the settling time of τ_{COMM} to be unpredictable. It was important to try and distinguish whether this transient behavior was a function of the input frequency or a fundamental characteristic of the DCX16S. If it was a function of the input frequency, then the solution could be to ramp I_{COMM} to reduce transients. If it was a quality of the motor, then it is possible that the DCX16S is unsuitable for our studies. Thus, frequency-response analysis was conducted to assess the response characteristics of our testing setup at different frequencies of I_{COMM}.

Frequency testing was completed at six frequencies (1-32Hz, doubling for each test, Figure 21). These frequencies were selected because from our own testing, a single turn performed by a rodent typically lasts between 100 - 300ms, which would correspond to turning frequencies of 10Hz - 3.33Hz. Additionally, for frequency response analysis, the torque predicted by the current command (τ_{PRED}) was selected as the "input" and the measured torque (τ_{MEAS}) was selected as the "output", as the measured current was a control signal for the generated torque. The resultant Bode plot from this analysis (Figure 23) indicated that the magnitude and phase response of τ_{MEAS} for an input of τ_{PRED} remained stable until the frequency of I_{PRED} was 16Hz or 32Hz. From

1-8Hz, the latency between τ_{PRED} and τ_{MEAS} remained constant at 6ms but increased to 8ms and 22ms at 16Hz and 32Hz. Similarly, the magnitude change remained around 0dB from 1- 16Hz but jumped from 0.388dB to 1.284dB from 16Hz to 32Hz (Figure 23). This indicates that at the frequencies higher than 16Hz, the system no longer exhibits stable or easily predictable torque generation behavior, which can be seen visually in the raw response plot in Figure 22f, where very clear "spiking" of τ_{MEAS} occurs at 32Hz.

A likely cause for this behavior is resonance due to the low-stiffness of the testing rig. Our test setup has multiple serially connected couplers and adapters, build from a mixture of various plastics and aluminum (Figure 5). Additionally, the pieces with screw holes have tolerance that theoretically allow for axial play. Overall, this almost certainly produces a less stiff system than a comparable testing rig with fewer individual pieces made of stiffer materials (e.g., all aluminum). The next step in frequency testing would be to use a single adapter between the motor shaft and the Nano17 made from a stiff material like aluminum or steel. The frequency response testing would be repeated with the same parameters, and if transient torques were still present, then the transients are likely an inherent property of the DCX16S. We considered briefly testing the more compliant coupling (Figure 5b), however it was assumed that this coupling would result in resonance-related transients at even lower frequencies, due to the lower stiffness. Despite these frequency-response findings, we continued to use a step-up current during the forelimb rotation with virtual stiffness task, as one of our initial goals was to compare the usage of a virtual stiffness to a real stiffness in the turning task.

Regardless of the gaps in frequency-response knowledge without a stiffer coupling, there are still some conclusions that can be made from the data obtained here. The frequency response within our selected rodent turning frequencies (3.33Hz – 10Hz) were stable with testing in both phase and magnitude. Additionally, we were able to validate the lower bound delay between the onset of I_{COMM} and τ_{MEAS} to be 6ms, which is acceptable as the subject tested in this study on the stiffness task had moved the knob less than 5° in that time (Figure 27). It is also interesting to note that there is such a long delay between I_{COMM} and τ_{MEAS}, as the two should be directly linearly related. It is likely I_{COMM} measures the current *to be controlled* based on V_{COMM}, not the *actual value* of the controlled current (which is how the ESCON 24/2 manual describes the value). A simple experiment to validate this would be to place a shunt resistor in series with the motor and directly measure the voltage drop over the resistor. This voltage drop would be directly related to the current flowing through the resistor by Ohm's Law. A useful behavioral task consideration from this data is to linearly relate the value of I_{COMM} to a kinematic variable of rodent turning behavior, such as knob position or velocity. Because the frequencies of rodent turning are within the stable range in the Bode plot, these "ramping" techniques should significantly reduce the chance of resonance-related torque spiking, assuming they are not due to the testing setup.

4.3.3 Rodent forelimb torques can be estimated by calculating I_{KNOB} and τ_{FRIC}

A final characterization of the behavioral device included computing I_{KNOB} and τ_{FRIC}. Accurate measurements of these physical characteristics of the device allowed us to compute the torque generated by the rat, τ_{RAT}, during the supination task using Equation (11). τ_{FRIC} was computed to be 0.247mNm after computing the relevant parameters to use in Equation (12). We are confident in this result because the deceleration curve in Figure 24 was very stereotyped on every trial. Additionally, the closest comparison value to this is the no-load current, which is the current at which the DCX16S is likely to overcome bearing friction and spin, which was 6.91mA. This corresponds to a no-load torque of 0.248mNm, which means our error was equal to 0.4%, which means our measurement accuracy was very good.

A value of I_{KNOB} of 9.661gcm^2 was calculated using the computed τ_{FRIC} and Equation (14). Because of the very small magnitude of this result, it is reported with some uncertainty. For example, computation of τ_W using only the radius of the manipulandum shaft results in a moment of inertia calculation that is different by nearly 50% (4.72gcm^2), even though including the radius of the tape used to hold the 10g weight only adds 0.41mm to the radius computation. Additionally, Maxon does not provide a mass of the rotor, only the total mass of the motor and encoder. Our radius of gyration calculation of 5.67mm is reasonable, though again, it is possible that our assumption of the mass of the rotor (50% of the total unit mass) is incorrect. Finally, another source of uncertainty was in the estimation of α_{KNOB}, as it was not as "clean" as the calculation for α_{ROTOR} (Figure 24). In the future, additional experiments must be done to estimate the motor of inertia in other ways, such as accelerating the knob using the ESCON 24/2 with a known torque or using CAD software estimation algorithms.

4.4 Rats can be trained using our device to perform static holds of increasing duration

The two subjects in this study on the static holding task showed positive improvements in holding behavior following training. The median hold time of Subject O increased by nearly 350ms (41ms to 390ms) while Subject N saw an increase of 382ms (72ms to 456ms) over 8 and 10 sessions, respectively (Figure 25). The median hold time threshold also increased for both subjects, with an increase of over 250ms for Subject O (6ms to 275ms) and nearly 350ms for Subject N (66ms to

400ms, Figure 25). The median was chosen as a good metric for task progress, as the distributions of holding times are not gaussian, as can be seen in the violin plots in Figure 25a, b. Each of these metrics reflect different qualities of task progression. The peak hold times show the changes in raw holding capability of a subject, including the bimodal distribution between longer and shorter holds. It can be seen in the violin plots in Figure 25a,b that the distribution of longer duration holds increases as a subject moves forward in training, indicating that the subjects are more willing to perform longer holds. The threshold is a more stable evaluation of progress as it can only change in smaller increments of 20ms. If a subject’s hold times fluctuate often between above and below threshold, then the threshold would not be expected to increase above the minimum. However, because both metrics increased for both subjects, the behavioral paradigm presented here is clearly able to increase both the holding capability and consistency of static holds of increased duration.

Each subject appeared to learn the task criteria at different rates, as Subject O did not see marked improvements in either the maximum hold time or hold time threshold until approximately session 5 while Subject N showed improved behavior consistently throughout training. It is common knowledge that there is natural variation between individuals during motor-related tasks, but other factors may also contribute. The factors for these differential learning rates are important to distinguish because they may be more pronounced with a larger sample size of subjects.

A potential contributor to these differences could be due to the method of adjusting the minimum hold time threshold. Because of the median-based thresholding method employed here, ideally rats would continue to hold for longer periods to continue receiving a high rate of rewards without having to adjust the minimum hold time. However, both subjects appeared to still be motivated to perform the task even if the hit rates in the task were near 50% based on the number of trials performed in a session (mean of 91.5 for Subject O and 131 for Subject N) and a relatively unchanging hold time threshold. Consistently improved holding behavior only occurred when the minimum hold time threshold was increased. This is possibly because the subject would only perform holds of the minimum duration required – after many trials at the same minimum threshold, a subject could simply learn a timing rather than learn the behavioral task, which would still provide a reward roughly 50% of the time. Increasing the minimum threshold would necessitate adaptation to the changing task requirements. It has been shown that rodents are able to produce precisely timed forelimb taps on a lever, which could be a similar case to what is observed here [38]. It is possible that in the future, increasing both the current hold time threshold and the minimum threshold as a training session proceeds could make improvements in behavior more consistent.

4.5 A virtual stiffness can be used to alter rodent forelimb supination behavior

With the one subject trained in this study on the forelimb stiffness task, we showed that our system could train forelimb rotation behavior and that this turning behavior could be affected by a virtual stiffness. Six stiffnesses of increasing value (0.18mNm through 1.44mNm) were tested to assess their impact on rodent turning kinematics. As shown in Figure 26a, median peak turning angles trended to decrease as the stiffness was increased. Like the peak hold times in Section 4.4, the median was selected as a good holistic metric of behavioral ability because the distribution of peak turning angles were not gaussian. The effect size of the decrease in median peak turning angle from 0.18mNm (83.31°) to 41mNm (44.02°) was large, but there was no point in increasing the stiffness that turning capabilities were eliminated. This means that our behavioral apparatus was able modulate a stiffness within the turning capabilities of the rat. Statistically, the effect of stiffness on peak turning angle compared to 0.18mNm was significant ($\alpha = 0.05$) at 0.72mNm ($p = 0.0075$), 1.08mNm ($p < 0.001$) and 1.44mNm ($p < 0.001$). To our knowledge, this is the first study to show that a virtual stiffness can produce quantifiable differences in rodent forelimb supination behavior. We note, however, that the subject chosen in this task was previously trained on a supination task and had been taken off training for 3 months prior to being used for this thesis. Therefore, the subject's turning strategies may be very different compared to naïve subjects. Further, only a single session was used for data at each stiffness, and each higher stiffness was tested in increasing order. For future studies, it is imperative to repeat the same stiffness testing completed here and in a random order to ensure that the findings of this thesis were not obtained by coincidence.

A longer-term outcome of this work is to use the "calibration" data from testing at various stiffnesses to select two appropriate stiffness values for a two-stiffness task (described in section 2.13.6). This task would switch between two stiffness values after a set number of trials (a "block") until a training session completed. Ideally, after switching to a higher stiffness, a subject's performance would decrease briefly until it was able to compensate for the higher stiffness value. While this was paradigm was not tested in this thesis, the effect of stiffness on peak turning angles found here are a good jumping-off point for assessing this paradigm in the future.

Despite the inability to train the two-stiffness task, the kinematics from the one-stiffness task trained here did reveal some notable time-related effects of varied stiffness. In Figure 27e, the averaged velocities for both failed and successful trials are plotted at each stiffness level. In successful trials, there appears to be a reduction in velocity at approximately 100ms into a turn attempt, followed by a slight increase in velocity until the turn attempt is complete. This drop does not appear in the averaged trace for the lowest stiffness (0.18mNm), nor in any of the failed attempt averages. This multiphasic turning is possibly due to the sensory detection and compensation for the motor torque or the subject's forelimb being unable to generate large torques in certain postures, or a combination of the two. Evidence for possible compensatory efforts can be seen in Figure 28, where most successful and failed average traces for rat generated torque experience a sharp decrease at approximately 50-75ms post initiation followed by an increase shortly after. This increase could indicate that the subject is attempting to compensate for the effect of the motor stiffness and is further supported by the lack of a torque decrease and compensatory response for the successful trace at 0.18mNm (the lowest torque). Overall, reflex times in the rat are much faster than those in humans, which provides circumstantial evidence that this response could be compensatory. For example, the H-reflex, which is an electrically induced reflex that bypasses the stretch receptor, is around 6-12ms in the rat tibia [54][55] and 45ms in human tibias [56]. It is therefore possible that a longer-latency reflex may occur within 50ms of turn onset. Long-latency reflex latencies for forelimb behavior were not found in a search of the literature. To our knowledge, this would be the first study to see a compensatory kinematic response related to forelimb rotation stiffnesses in the rat. Ideally, our work can provide a baseline for other researchers to select appropriate virtual or real stiffnesses in their own work.

Regarding posture-related torque generation differences, there was no significant evidence to show that any particular knob position resulted in lower rat generated torques (Figure 28c,d). Importantly, the positions in Figure 28 are of the knob, and do not necessarily indicate the true posture of the rat. The subject could have used a different starting paw posture depending on the difficulty of the session to generate a larger initial torque. More accurate assessment of posture will require the use of high-speed camera data to analyze the relationship between paw posture and generated torque. Overall, the data in Figure 28e indicated that Subject G produced greater peak torques with increasing stiffness, resulting in the same net knob torque (Figure 27). Future studies will require a greater sample size of subjects to determine whether posture-dependent or somatosensory-related compensatory responses are consistent across multiple animals.

While the virtual stiffness clearly influences the turning capabilities of the subject tested here, it is important to try and distinguish some potential effects on turning behavior using a virtual stiffness instead of a real stiffness. At the highest level, the MotoTrak experiments showed higher task success rates (~90% [21]) for their subjects on the rotation task compared to our subject's peak of 72% (Figure 26c). This is possibly due to two major factors. Firstly, the subjects on the MotoTrak were overtrained for several months (11 weeks on average), while our subject has only been trained on the supination task for slightly over one month. Secondly, the MotoTrak paradigm allowed for as many turn attempts as possible in a two-second window, while our paradigm aims for precision and only allows a single attempt. Therefore, our hit rates are reasonable given the comparative difficulty of the task and the lack of overtraining. Comparing kinematics, the MotoTrak subjects had a mean turn angle of 84.8° while our subject had a peak mean of 75.2°. The difference here is likely due to the number of attempts allowed on the MotoTrak, as the peak mean of the failed trials was much lower for our subject (20.7°) which would lower the average. The kinematic variable that seemed to vary the most between our study and the MotoTrak studies was velocity – the average peak velocities found in [21] were less than 400 deg/s, while in our subjects the peak velocities were found to be twice that (~800 deg/s, Figure 27i) even at the highest stiffness level, which was stiffer than the MotoTrak manipulandum. It is possible that this is at least in part due to the delay between turn onset and motor torque onset in our system. In a real stiffness apparatus like the MotoTrak, the subject is immediately informed of the difficulty of a task due to the increased torque countering any turning attempts of the manipulandum. In our task, the subject may not necessarily detect the increased stiffness immediately, as there is a delay of at least 6ms post turn initiation for the motor torque to become active. Therefore, subjects may resort to performing initial turns of larger torque to overcome the inevitable delayed stiffness generated by the motor. It is also feasible that this the turning strategy employed is highly subject-specific, and that other subjects trained on this task may not exhibit such rapid turns. Overall, this was the starkest difference between the behavioral variables in our system and the MotoTrak.

5. CONCLUSIONS

5.1.1 The behavioral device is an effective tool for training holding and turning behavior

The rat is an understudied model compared to non-human primates and humans with regards to skilled forelimb behavior. Part of this appears to be due to a lack of appropriate tools to properly train motor cortically driven tasks. Our novel behavioral device was shown here to be a powerful tool for scalable training of skilled forelimb behaviors in rats.

Our novel application of a touch sensor for rodent work confirms that a low-latency (4ms), high accuracy (96.7%) touch detection method on a rotation manipulandum designed for rodents is possible. Additionally, the holding task provided preliminary evidence of a method to rapidly train long-duration static holds in rats (N=2). This is a vital building block for providing a kinematic separation between reaching behavior and turning behavior for the manipulandum. Further, there is some evidence that motor cortex is potentially involved in behavioral timing delays in rats [38] and holding behavior in mice [43]. It is possible that the static holding behavior trained here have a distinct representation in motor cortical data. Ideally, the task shown in [43] can be adapted for our behavioral setups for a high-throughput way to assess cued turning behaviors.

Characterization of our torque delivery showed our chosen motor controller (ESCON 24/2) and motor (24V DCX16S) combination to be appropriate for providing virtual stiffnesses on the scale of rodent forelimb torques. With only a 3.58% error in torque production at steady-state compared to manufacturer specifications, our system is extremely precise, and with a delay of 6ms to torque onset post-current command (between 1-16Hz I_{COMM} frequency), it is also fast. Further, we were able to characterize the moment of inertia of the manipulandum and the torsional dynamic friction of the rotor so that we can compute net torques on the knob *and* torques produced by the subjects. Additional efforts are necessary to model the rat generated torque accurately, however this study lays a strong foundation for assessing rat forelimb torques, which have never been characterized in the literature. Finally, using this fully characterized system we were able to train a forelimb rotation task at varied knob stiffness levels similar to that trained in [21]. Peak turn angles for the one subject tested here were decreased as knob stiffness increased, as expected. There was also preliminary evidence of compensatory forelimb torque generation in the rat at higher stiffnesses. In the future, we hope to train more rodents on this task and proceed to a two-stiffness paradigm to further analyze compensatory efforts by the subjects with changing task difficulty.

5.1.2 Exploring motor cortical dynamical systems

Elucidating the existence of dynamics in rat motor cortex in future experiments would require electrophysiological implantation into rat motor cortex. Electrode implants would be placed into caudal forelimb area and rostral forelimb area and electrophysiological data would be collected while subjects perform the full behavioral paradigms trained here. Then, the data could be analyzed for dynamics using well-established tools like jPCA [7] and kinematics can be decoded using denoised neural estimates with LFADS. With greater options for optogenetic control, disease-specific models and strain variation, rodents open a wide range of possibilities to study questions in both basic and translational medicine, validating the rat as a powerful model to study motor cortex could provide a cost-effective, high-throughput alternative to primate research.

APPENDIX

6.1 Usage documentation on taskRunner, ratCoach and trialLogger

Extensive usage notes and documentation on taskRunner, ratCoach and trialLogger can be found in the Google document link in [57]. The link includes setup notes for all systems, commands for starting a training session and basic debugging protocols.

Table 7. Subject Demographics and Experimental Usage.

Subject ID	Sex (M/F)	Age at training onset (mo.)	Included in Study? (Y/N)	Paradigms trained
G	F	8	Y	Turning w/ stiffness
I	F	3	N	Preliminary data
J	F	3	N	Preliminary data
M	F	~	N	Failed SRT
N	F	4	Y	Knob-holding
O	F	4	Y	Knob-holding

Table 8. Cost-Breakdown for Supplies in Behavioral System Version 1.

Item Description	**Supplier**	**Model No.**	**Qty**	**Unit Cost ($)**	**Total (Qty x Unit Cost) ($)**
NIDAQ controlling computer (one computer per 2 boxes)			**1/2**	**1041.13**	**520.57**
ASUS B85M-E/CSM DDR3 1600 LGA 1150 Motherboard	Amazon	B85M-E/CSM	1	93.80	93.80
Intel Core i7-4790K @ 4.00GHz + Cooler	Amazon	i7-4790K	1	373.90	373.90
Crucial 16GB Single DDR4	Amazon	CT16G4DFD824A	2	163.50	163.50
Rosewill rack-mountable chassis	Amazon	RSV-R4000	1	84.99	84.99
SeaSonic 650W Power	Amazon	SS-650KM3	1	135.00	135.00

Supply					

Table 8 (continued).

ViewSonic 27-inch 75Hz 1080p monitor	Amazon	VX2757-MHD	1	170.00	170.00
USB Wired Keyboard	Amazon	KU-0833	1	12.95	12.95
USB Wired Mouse	Amazon	MSU0939	1	6.99	6.99
Box Materials			**1**	**394.15**	**394.15**
Clear Cast Acrylic Sheet (12” x 12” x ¼”)	McMaster-Carr	8560K354	8	17.34	138.72
Clear Cast Acrylic Sheet (12” x 12” x 1/8”)	McMaster-Carr	8560K239	1	9.15	9.15
T-slotted extrusion (1” x 1” x 8 ft.)	McMaster-Carr	47065T101	2	23.57	47.14
T-slot corner brackets (1” x 3”) 47065T267	McMaster-Carr	47065T267	16	7.52	120.32
Linear sleeve bearing carriage **(cut in half per box)**	McMaster-Carr	47065T961	1/2	59.41	29.70
M6 1mm x 16mm Pitch screws (8 pc)	Amazon	B00JDU1S7C	1/2	4.98	4.98
M1.6x4mmx0.35mm Pitch Hex screws (100 pcs)	Amazon	a15040700ux0216	1	8.44	8.44
Misc. M3 screws (6mm/8mm/10mm/12mm, 290 pcs)	Amazon	M306081012	1	5.99	5.99
I/O controller			**1**	**722.49**	**722.49**
NI-6323 PCIe Multifunction I/O Device **(one card per 2 boxes)**	National Instruments	781045-01	1/2	961.00	480.50
CB-68LPR I/O Connector Block	National Instruments	777145-02	1	95.00	95.00
RC68-68 Ribbon Cable, 68 D-Type to 68 VHDCI, 1m	National Instruments	187252-01	1	92.00	92.00

Table 8 (continued).

Raspberry Pi 3 B+ kit w/ 2.5A power supply, heatsinks and case	Amazon	B07BC7BMHY	1	54.99	54.99
Enclosure Materials			**1**	**76.69**	**89.06**
Melamine White Panel (¾” x 4 ft. x 8 ft)	Home Depot	461877	2	28.64	57.28
L-Brackets w/ screws (40mm x 40mm brackets, ⅝” long screws)	Amazon	TNT523	1	9.99	9.99
Surface-Mount Piano Hinge with Holes, Zinc-Plated Steel, 1-1/16" Wide, 0.174" Knuckle Diameter, 1 ft. Long	McMaster-Carr	1608A41	2	4.71	9.42
Cabinet Handle (1/2" Diameter), 5.88" Length (3.5" Hole Center, 10 pcs)	Amazon	B01MS9NR0I	1	12.37	12.37
External Devices			**1**	**1449.72**	**1349.74**
Aluminum knob (SLS printed)	iMaterialise	N/A	1	32.00	32.00
Stepper Motor with 28cm Lead Screw: Bipolar, 200 Steps/Rev, 42×38mm, 2.8V, 1.7 A/Phase	Pololu	2267	1	49.99	49.99
Pellet Dispenser With 45mg Interchangeable Pellet Size Wheel And Optional Stand	Lafayette Instruments	80209-45	1	595.00	595.00
24V power supply	Amazon	3206-24V	1	16.00	16.00
ESCON Module 24/2, 4-Q servo controller for DC/EC motors, 2/6 A, 10-24 VDC	Maxon Motors	466023	1	98.38	98.38
ESCON Module 24/2 Motherboard	Maxon Motors	486400	1	95.25	95.25

Table 8 (continued).

DCX 16 S Ø16 mm, Precious Metal Brushes CLL, ball bearings (24V) + Encoder ENC 16 RIO, 32768 counts per turn, 3-channel, with RS 422 line driver	Maxon Motors	B789B7B6D173	1	302.05	302.05
Raspberry Pi 10.1" LCD Display	Digikey	1597-1102-ND	1	90.78	90.78
Speaker 8ohm 2W Top Port 80dB	Digikey	102-1299-ND	1	5.95	5.95
SV3C Security Camera, 1080P POE (Power Over Ethernet) IP Camera	Amazon	SV-B01POE-1080P-L	1	39.99	39.99
Adafruit MPR121 12-key capacitive touch sensor w/ breakout board	Amazon	B00SK8PVNA	1	9.35	9.35
TP-Link 8-Port Gigabit Ethernet PoE Desktop Switch with 4-PoE Ports **(4 ports total, so ¼ cost per box)**	Amazon	B00BP0SSAS	1/4	59.99	15.00
GRAND TOTAL					**3076.01**

Table 9. Cost-Breakdown for Supplies in Behavioral System Version 2.

Item Description	**Supplier**	**Model No.**	**Qty**	**Unit Cost ($)**	**Total (Qty x Unit Cost) ($)**
I/O Controller			**1**	**109.98**	**109.98**
Raspberry Pi 3 B+ kit w/ 2.5A power supply	Amazon	B07BC7BMHY	2	54.99	109.98
Box Materials			**1**	**260.62**	**256.12**
Clear Cast Acrylic Sheet (12” x 12” x 5/16”)	McMaster-Carr	8560K591	6	20.25	121.50
Clear Cast Acrylic Sheet (12” x 12” x 1/8”)	McMaster-Carr	8560K239	1	9.15	9.15

Table 9 (continued).

Clear Cast Acrylic Sheet (12” x 12” x ¼”)	McMaster-Carr	8560K354	2	17.34	34.68
M4 x 10mm x 0.7mm thumbscrews	Amazon	a15082600ux0181	1	18.08	18.08
Linear sleeve bearing carriage **(cut in half per box)**	McMaster-Carr	47065T961	1/2	59.41	29.70
1/4-20 x 3/4" countersunk screws	Amazon	B01LZNGJAM	1	15.38	15.38
Surface-Mount Piano Hinge with Holes, Zinc-Plated Steel, 1-1/16" Wide, 0.174" Knuckle Diameter, 1 ft. Long	McMaster-Carr	1608A41	2	4.71	9.42
Cabinet Handle (1/2" Diameter), 5.88" Length (3.5" Hole Center, 10 pcs)	Amazon	B01MS9NR0I	1	12.37	12.37
T-slotted extrusion (1” x 1” x 1 ft.)	McMaster-Carr	47065T101	1	5.84	5.84
Enclosure Materials			**1**	**96.06**	**96.06**
IKEA SEKTION cabinet	IKEA	202.655.13	1	58.00	58.00
Melamine White Panel (¾” x 4 ft. x 8 ft)	Home Depot	461877	1	28.64	28.64
Surface-Mount Piano Hinge with Holes, Zinc-Plated Steel, 1-1/16" Wide, 0.174" Knuckle Diameter, 1 ft. Long	McMaster-Carr	1608A41	2	4.71	9.42
External Devices			**1**	**762.38**	**762.38**
Aluminum knob (SLS printed)	iMaterialise	N/A	1	32.00	32.00

Table 9 (continued).

Stepper Motor with 28cm Lead Screw: Bipolar, 200 Steps/Rev, 42×38mm, 2.8V, 1.7 A/Phase	Pololu	2267	1	49.99	49.99
ESCON Module 24/2 Motherboard	Maxon Motors	486400	1	95.25	95.25
DCX 16 S Ø16 mm, Precious Metal Brushes CLL, ball bearings (24V) + Encoder ENC 16 RIO, 32768 counts per turn, 3-channel, with RS 422 line driver	Maxon Motors	B789B7B6D173	1	302.05	302.05
Raspberry Pi 10.1" LCD Display	Digikey	1597-1102-ND	1	90.78	90.78
Speaker 8ohm 2W Top Port 80dB	Digikey	102-1299-ND	1	5.95	5.95
SV3C Security Camera, 1080P POE (Power Over Ethernet) IP Camera	Amazon	SV-B01POE-1080P-L	1	39.99	39.99
Adafruit MPR121 12-key capacitive touch sensor w/ breakout board	Amazon	1982	1	9.35	9.35
TP-Link 8-Port Gigabit Ethernet PoE Desktop Switch with 4-PoE Ports **(4 ports total, so ¼ cost per box)**	Amazon	B00BP0SSAS	1/4	59.99	15.00
Custom-made gravity-hopper feeder (3D Printed)	3D Printed	N/A	1	80	80
Delrin Acetal Resin Sheet (12” x 12” x 1/16”)	McMaster-Carr	8575K111	1	10.34	10.34
Stepper Motor: Unipolar/Bipolar, 200 Steps/Rev, 42×48mm, 4V, 1.2 A/Phase	Pololu	1200	1	19.95	19.95
Adafruit MCP4725 Breakout Board - 12-Bit DAC	Amazon	B00SK8MBXI	1	7.77	7.77

Table 9 (continued).

32-Bit Multi-Mode Counter with Serial Interface	LSI/CSI	LS7366R	1	3.96	3.96
GRAND TOTAL					**1224.54**

REFERENCES

[1] J. C. Kao, P. Nuyujukian, S. I. Ryu, M. M. Churchland, J. P. Cunningham, and K. V. Shenoy, "Single-trial dynamics of motor cortex and their applications to brain-machine interfaces," *Nat. Commun.*, vol. 6, no. May, p. 7759, 2015.

[2] V. Gilja *et al.*, "Clinical translation of a high-performance neural prosthesis.," *Nat. Med.*, vol. 21, no. 10, pp. 6–8, 2015.

[3] C. Pandarinath *et al.*, "High Performance communication by people with paralysis using an intracortical brain-computer interface," pp. 1–27, 2016.

[4] M. Schiefer, D. Tan, S. M. Sidek, and D. J. Tyler, "Sensory feedback by peripheral nerve stimulation improves task performance in individuals with upper limb loss using a myoelectric prosthesis.," *J. Neural Eng.*, vol. 13, no. 1, p. 016001, 2016.

[5] P. A. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex," *J.Neurosci.*, vol. 2(11), no. 11, pp. 1527–1537, 1982.

[6] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," *Science (80-.).*, vol. 233, no. 4771, pp. 1416–1419, 1986.

[7] M. M. Churchland *et al.*, "Neural population dynamics during reaching," *Nature*, 2012.

[8] C. Pandarinath *et al.*, "Neural population dynamics in human motor cortex during movements in people with ALS," *Elife*, vol. 4, no. JUNE, 2015.

[9] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, S. I. Ryu, and K. V. Shenoy, "Cortical Preparatory Activity: Representation of Movement or First Cog in a Dynamical Machine?," *Neuron*, vol. 68, no. 3, pp. 387–400, 2010.

[10] H. Makino *et al.*, "Transformation of Cortex-wide Emergent Properties during Motor Learning," *Neuron*, vol. 94, no. 4, 2017.

[11] a. D. Grosmark and G. Buzsaki, "Diversity in neural firing dynamics supports both rigid and learned hippocampal sequences," *Science (80-.).*, vol. 351, no. 6280, pp. 1440–1443, 2016.

[12] S. Vyas, N. Even-Chen, S. D. Stavisky, S. I. Ryu, P. Nuyujukian, and K. V. Shenoy, "Neural Population Dynamics Underlying Motor Learning Transfer," *Neuron*, vol. 97, no. 5, 2018.

[13] J. A. Gallego, M. G. Perich, L. E. Miller, and S. A. Solla, "Neural Manifolds for the Control of Movement," *Neuron*, vol. 94, no. 5. 2017.

[14] H. R. Sheahan, D. W. Franklin, and D. M. Wolpert, "Motor Planning, Not Execution, Separates Motor Memories," *Neuron*, vol. 92, no. 4, 2016.

[15] C. Pandarinath *et al.*, "Inferring single-trial neural population dynamics using sequential auto-encoders," *Nat. Methods*, vol. 15, no. 10, pp. 805–815, 2018.

[16] K. Morandell and D. Huber, "The role of forelimb motor cortex areas in goal directed action in mice," *Sci. Rep.*, vol. 7, no. 1, 2017.

[17] X. Wang *et al.*, "Deconstruction of Corticospinal Circuits for Goal-Directed Motor Skills," *Cell*, vol. 171, no. 2, p. 440–455.e14, 2017.

[18] J. A. Kleim, S. Barbay, and R. Nudo, "Functional reorganization of the rat motor cortex following motor skill learning," *J. Neurophysiol.*, vol. 80, pp. 3321–3325, 1998.

[19] A. R. Brown and G. C. Teskey, "Motor Cortex Is Functionally Organized as a Set of Spatially Distinct Representations for Complex Movements," *J. Neurosci.*, vol. 34, no. 41, pp. 13574–13585, 2014.

[20] E. Meyers *et al.*, "The supination assessment task: An automated method for quantifying forelimb rotational function in rats," *J. Neurosci. Methods*, vol. 266, pp. 11–20, 2016.

[21] A. Sindhurakar *et al.*, "An Automated Test of Rat Forelimb Supination Quantifies Motor Function Loss and Recovery After Corticospinal Injury.," *Neurorehabil. Neural Repair*, p. 1545968316662528, Aug. 2016.

[22] B. C. Vigaru *et al.*, "A robotic platform to assess, guide and perturb rat forelimb movements," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 5, pp. 796–805, 2013.

[23] B. Vigaru *et al.*, "A small-scale robotic manipulandum for motor training in stroke rats," *IEEE Int. Conf. Rehabil. Robot.*, 2011.

[24] P. B. Yin and S. Kitazawa, "Long-lasting aftereffects of prism adaptation in the monkey," *Exp. Brain Res.*, vol. 141, no. 2, pp. 250–253, 2001.

[25] A. Takemoto, A. Izumi, M. Miwa, and K. Nakamura, "Development of a compact and general-purpose experimental apparatus with a touch-sensitive screen for use in evaluating cognitive functions in common marmosets," *J. Neurosci. Methods*, vol. 199, no. 1, pp. 82–86, 2011.

[26] T. F. Elsmore, J. K. Parkinson, and R. L. Mellgren, "Video touch-screen stimulus-response surface for use with primates," *Bull. Psychon. Soc.*, vol. 27, no. 1, pp. 60–63, 1989.

[27] M. Longley, E. L. Willis, C. X. Tay, and H. Chen, "An open source device for operant licking in rats," *PeerJ*, vol. 5, p. e2981, 2017.

[28] A. E. Yang, M. J. Z. Hartmann, and S. Bergbreiter, "Contact-Resistive Sensing of Touch and Airflow Using A Rat Whisker," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, 2018, pp. 1187–1192.

[29] P. Tommasino, A. Melendez-Calderon, E. Burdet, and D. Campolo, "Motor adaptation with passive machines: A first study on the effect of real and virtual stiffness," *Comput. Methods Programs Biomed.*, vol. 116, no. 2, pp. 145–155, 2014.

[30] J. R. Flanagan, E. Nakano, H. Imamizu, R. Osu, T. Yoshioka, and M. Kawato, "Composition and decomposition of internal models in motor learning under altered kinematic and dynamic environments.," *J. Neurosci.*, vol. 19, no. 20, p. RC34, 1999.

[31] R. F. Boian, J. E. Deutsch, C. S. Lee, G. C. Burdea, and J. Lewis, "Haptic effects for virtual reality-based post-stroke rehabilitation," *11Th Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst. - Haptics 2003, Proc.*, no. Figure 1, pp. 247–253, 2003.

[32] K. International, J. Oblak, and I. Cikajlo, "A universal haptic device for arm and wrist rehabilitation," *Work*, vol. 1, no. 3, pp. 436–441, 2009.

[33] J.-C. Metzger, O. Lambercy, and R. Gassert, "Performance comparison of interaction control strategies on a hand rehabilitation robot," *2015 IEEE Int. Conf. Rehabil. Robot.*, pp. 846–851, 2015.

[34] R. C. V. Loureiro and W. S. Harwin, "Reach & grasp therapy: Design and control of a 9-DOF robotic neuro-rehabilitation system," *2007 IEEE 10th Int. Conf. Rehabil. Robot. ICORR'07*, vol. 00, no. c, pp. 757–763, 2007.

[35] A. Zemmar, B. Kast, K. Lussi, A. R. Luft, and M. E. Schwab, "Acquisition of a High-precision Skilled Forelimb Reaching Task in Rats," *J. Vis. Exp.*, no. 100, pp. 1–7, 2015.

[36] I. Q. Whishaw, B. Gorny, A. Foroud, and J. A. Kleim, "Long-Evans and Sprague-Dawley rats have similar skilled reaching success and limb representations in motor cortex but different movements: Some cautionary insights into the selection of rat strains for neurobiological motor research," *Behav. Brain Res.*, vol. 145, no. 1–2, pp. 221–232, 2003.

[37] S. A. Hays *et al.*, “The isometric pull task: A novel automated method for quantifying forelimb force generation in rats,” *J. Neurosci. Methods*, vol. 212, no. 2, pp. 329–337, 2013.

[38] R. Kawai *et al.*, “Motor Cortex Is Required for Learning but Not for Executing a Motor Skill,” *Neuron*, vol. 86, no. 3, pp. 800–812, 2015.

[39] B. P. Ö. Ashesh K. Dhawale, Rajesh Poddar, Evi Kopelowitz, Valentin Normand, Steffen B. E. Wolff, “Automated long-term recording and analysis of neural activity in behaving animals,” 2015.

[40] C. C. Wong, D. S. Ramanathan, T. Gulati, S. J. Won, and K. Ganguly, “An automated behavioral box to assess forelimb function in rats,” *J. Neurosci. Methods*, vol. 246, pp. 30–37, 2015.

[41] R. Poddar, R. Kawai, and B. P. Ölveczky, “A fully automated high-throughput training system for rodents,” *PLoS One*, vol. 8, no. 12, pp. 1–10, 2013.

[42] D. J. Ellens *et al.*, “An automated rat single pellet reaching system with high-speed video capture,” *J. Neurosci. Methods*, vol. 271, pp. 119–127, 2016.

[43] C. P. Burgess *et al.*, “High-Yield Methods for Accurate Two-Alternative Visual Psychophysics in Head-Fixed Mice,” *Cell Rep.*, vol. 20, no. 10, pp. 2513–2524, 2017.

[44] J. Tonkiss, P. Shultz, and J. R. Galler, “Long-evans and sprague-dawley rats differ in their spatial navigation performance during ontogeny and at maturity,” *Dev. Psychobiol.*, vol. 25, no. 8, pp. 567–579, 1992.

[45] P. M. VandenBerg, T. M. Hogg, J. A. Kleim, and I. Q. Whishaw, “Long-Evans rats have a larger cortical topographic representation of movement than Fischer-344 rats: A microstimulation study of motor cortex in naïve and skilled reaching-trained rats,” *Brain Res. Bull.*, vol. 59, no. 3, pp. 197–203, 2002.

[46] J. Faraji, G. A. Metz, and R. J. Sutherland, “Characterization of spatial performance in male and female Long-Evans rats by means of the Morris water task and the ziggurat task,” *Brain Res. Bull.*, vol. 81, no. 1, pp. 164–172, 2010.

[47] A. Karni *et al.*, “The acquisition of skilled motor performance: Fast and slow experience-driven changes in primary motor cortex,” *Proc. Natl. Acad. Sci.*, vol. 95, no. 3, pp. 861–868, 1998.

[48] ATI, “Nano17 Specifications.,” 2014. [Online]. Available: https://www.ati-ia.com/products/ft/ft_models.aspx?id=Nano17.

[49] Farnell, “Gorilla tape technical data sheet,” 2016. [Online]. Available: http://www.farnell.com/datasheets/1684009.pdf.

[50] S. A. Barnett, “Experiments on neophobia in wild and laboratory rats,” *Br. J. Psychol.*, vol. 90, no. 2, pp. 190–197, 1958.

[51] O’Shea Dan, “matUDP: A Bus Communication Layer and Data Logger for Simulink Real Time,” 2018. [Online]. Available: https://github.com/djoshea/matudp.

[52] H. Masum, “PlotPub: Publication Quality Graphs in MATLAB.,” 2018. [Online]. Available: https://github.com/masumhabib/PlotPub.

[53] M. M. Buitrago, T. Ringer, J. B. Schulz, J. Dichgans, and A. R. Luft, “Characterization of motor skill and instrumental learning time scales in a skilled reaching task in rat,” *Behav. Brain Res.*, vol. 155, no. 2, pp. 249–256, 2004.

[54] E. F. Stanley, “Sensory and motor nerve conduction velocities and the latency of the H reflex during growth of the rat,” *Exp. Neurol.*, vol. 71, no. 3, pp. 497–506, 1981.

[55] M. Gozariu, V. Roth, F. Keime, D. Le Bars, and J. C. Willer, “An electrophysiological

investigation into the monosynaptic H-reflex in the rat," *Brain Res.*, vol. 782, no. 1–2, pp. 343–347, 1998.

[56] W. R. Jankus, L. R. Robinson, and J. W. Little, "Normal limits of side-to-side H-reflex amplitude variability," *Arch. Phys. Med. Rehabil.*, vol. 75, no. 1, pp. 3–7, 1994.

[57] A. Corsten, "taskRunner, trialLogger and ratCoach User Guide," *2018*. [Online]. Available: https://docs.google.com/document/d/1z2nGqebKEROe0iEHg176ljJCVerPGdu6rx0nrTOMFl4/edit?usp=sharing.