Using Load-Cells to Unveil Limitations to the Human Movement System

Although Apparently Versatile and Robust, the Control of Finger Musculature Operates on the Verge of Failure

Master thesis as a requirement to obtain the title MSc in Biomedical Engineering from ETH Zurich

Submitted by:
Richard Lukas Bumann

Supervisors:
Prof. Francisco J. Valero-Cuevas (USC, Los Angeles)
Prof. Robert Riener (ETH Zürich)

Introduction

The human body is an astonishing movement system in its performance and complexity. Every fourth year, athletes from all over the world demonstrate the boundaries of the human performance at the Olympic games. Track and field athletes accomplish incredible force and speed performances when running the 100 m in well below 10 s or shot putting a 16 lb heavy shot farther than 20 m. Moreover, shooters and springboard divers or gymnasts document an amazing level of accuracy and coordination, while fencers impress with extremely fast reactions. Yet, even everyday actions are surprisingly difficult and require a complex interplay of nervous system and muscles. Considerably simple actions such as knitting or even turning over a page in a book consist of a extensive set of atomic movements and complex neural control strategies.

It is still a big challenge to analyze these control mechanisms. There are several non-invasive (surface EMG) and minimal-invasive (fine wire EMG) to measure muscle activity. However, it is very difficult to measure the collectivity and the cooperation of the different muscles. For example, it is well possible to monitor the activity of the seven intrinsic muscles in a single finger. However, the hand is a complex grasping organ and actuating one finger always causes co-actuation of the other fingers. It is even more difficult to measure nervous signals. The neuronal activity in the brain can be assessed e.g. through fMRI or EEG. But these methods capture only a large pool of neurons and leave a lot of space for interpretations. On the other hand, measuring the activity of single neurons neglects the complex circuitry and the interaction of neurons.

For these reasons, we chose the most indirect way to study the neural controller for movements. We simply measured the force as an end product of the movement system and connected these recordings with previous knowledge about the nervous system. In our study presented in the chapters 1 and 2, we assessed the most basic tasks that can be performed with the index finger. We trained subjects until they reached their best possible performance to conclude on the limitations set by the nervous system and the structure of the controller.

In order to be able to draw relevant conclusions, we had to rely on precisely working force measuring devices. Code of practice in the Brain-body dynamics lab has always been the use of six-axis load-cells for this purpose. They can accurately measure forces and torques in all spatial directions. Nevertheless, active usage changes the mechanical properties of the complex system and changes the calibration parameters of the sensor. Even if the manufacturer offers recalibration services, the process is costly and time-consuming. In chapter 3, we describe how we developed an approach to calibrate complex load-cells in-lab with a precision that comes close to industry standards.

These two projects alone would not completely justify the title of this thesis,

even if we used a load-cell to unveil limitations of index finger dexterity. However, numerous studies in the BBDL relied on similar principles, measuring only the output of the human body as a movement system. For example I was profoundly involved in a study, which investigates fatigue mitigating mechanisms during knee extension. Subjects had to maintain low-level knee-extension forces over a long period of time, which was ensured using a visual feedback that was based on load-cell measurements. Measuring the EMG signals of the different knee extension synergists revealed that they are mutually active and inactive for forces below 15% MVC but become all necassary due to biomechanic constraints for higher forces. In addition to that, I was loosely involved in the data collection of a study, which used load-cells and accelerometers to study the balancing mechanisms during slacklining. Both studies are beyond the scope of this thesis. However, they legitimate the holistic title of my thesis, describing the different involvements during my master's internship.

Contents

Writing, to me,
is simply thinking through my fingers.
Isaac Asimov

Chapter 1

Although Apparently Versatile and Robust, the Control of Finger Musculature Operates on the Verge of Failure

Human dexterous manipulation has been praised since the dawn of civilization. Traditionally this is attributed to evolutionary adaptations, and the remarkable ability of the nervous system to control the numerous muscles driving the complex mechanism of the hand. While there are well-accepted anatomical and neural limitations for the control of manipulation, these are considered only minor impediments to what is otherwise versatile and robust everyday function. Here we provide evidence contrary to this widespread notion. We demonstrate that the evolutionary adaptations for individuated neural control of finger muscles (and thus fingers) necessary for dexterous manipulation are far from robust or complete. We compare the dynamical performance of two simple and equally necessary index finger tasks for dexterous manipulation-the scaling of the magnitude of a fingertip force vector in a given direction vs. the re-direction of a fingertip force vector of a given magnitude. Our multi-day testing regime shows that the former can be done well enough to exhibit a robust speed-accuracy tradeoff (a trademark of evolutionary mature neuromechanical function). In contrast, the latter can barely be done at all, let alone optimized. Placing these results in the context of other recent work and clinical experience, there is mounting evidence that everyday manipulation actually takes place at the limits of performance and therefore on the verge of failure. Thus our apparently versatile manipulation owes as much to adequate-but incomplete and non-robust-neuromechanical evolutionary adaptations, as to socio-biological co-evolutionary design of objects and tools to

make them forgiving when manipulated.

What gives us the ability to perform dexterous manipulation tasks so necessary for everyday life? Current thinking points to indisputable evidence from the fossil record, comparative anatomy/neuroanatomy, and neuroscience showing clear and specific adaptations for dexterous manipulation in hominids and modern humans [1–3]. These adaptations show a clear trend towards the ability to produce individuated finger motions and forces [4, 5]. After all, being able to dynamically regulate the magnitude and direction of individual fingertip force vectors is critical to any mechanical definition of dexterous manipulation that includes the ability to grasp, lift, hold, reorient and use objects and tools in the presence of perturbations[1].

Direct evidence for the evolutionary trend towards individuated finger control comes from many anatomical and neuroanatomical features of the human hand (e.g., individuated muscle slips of flexors and extensor muscles, dedicated independent muscles of thumb and index finger). The ability of the central nervous system (CNS) to have an imperfect, but nevertheless remarkable, control over the numerous muscles driving the complex mechanism of the hand is indirect evidence for evolutionary adaptations at the neural level favoring individual control of muscles. Therefore, even though there are multiple well-accepted anatomical (e.g., multiple tendinous, intra- and inter-muscle connections) and neural (e.g., short-term synchronization across motor unit pools [4, 5], enslaving and force deficits [6]) limitations for the control of manipulation, these are considered only minor impediments to what is otherwise obviously versatile everyday function [5]. So much so that thinkers since the ancient Greeks [7] have debated whether the human hand is the cause or consequence of superior intelligence[2]. However, if the human hand is so neuromechanically redundant and versatile, why is dexterous manipulation so susceptible to even mild neurological conditions, seldom recovers full functionality after stroke [8] or cerebral palsy [9], takes years to develop in childhood [10], or degrades with healthy aging [11]? Our work is motivated by this apparent paradox.

We focused on a detailed quantification of the dynamic control of the magnitude and direction of fingertip force vectors because it is a fundamental building block for dexterous manipulation [12, 13]. In particular we tested whether healthy, young participants can maintain the finger posture and (i) rapidly scale up the magnitude of the fingertip force vector while maintaining the direction constant; or conversely (ii) rapidly change the direction of the fingertip force vector while maintaining the magnitude constant. These two seemingly simple capabilities are fundamental aspects of multifinger manipulation and come into play when, for example, we respectively increase the grasp force on a tool when preparing for its use, or reorient an object with respect to the gravity vector. In these dynamic manipulation situations the magnitude or direction of the static fingertip force vectors must change quickly (else the object can be dropped), but the posture of the fingers remains constant and static equilibrium is maintained.

[1] Individuated finger motions are necessary in dexterous manipulation to place the fingertips on objects for grasp acquisition, and reconfigure the grasp as needed during manipulation. But the magnitude and direction of individual fingertips are critical to hold the object and produce the grasp task.

[2] Aristotle writes that "Now it is the opinion of Anaxagoras that the possession of these hands is the cause of man being of all animals the most intelligent. But it is more rational to suppose that his endowment with hands is the consequence rather than the cause of his superior intelligence."

Our experimental design was driven by the fact that individuation of fingertip forces hinges on the ability of the CNS to control individual motoneuron pools (the populations of α-motoneurons in the spinal cord that activate individual muscles). Decades of research have sought to answer the questions of whether and how well the human CNS has adapted to activate specific motoneuron pools of the hand. Seminal and recent work ([14, 15, 4, 16]; for a review, [17]) has mapped out and identified preferential projections from the brain to the motoneuron pools of hand muscles, which is the neuroanatomical basis of finger individuation. But questions remain about the extent to which these projections are functionally sufficient for individuated control of finger muscles for everyday tasks. For example, recently we have shown that that an involuntary stereotypical grasp tendency pervades voluntary dynamic multifinger manipulation [18]. This functionally detrimental competition in the individuated control of specific motoneuron pools pollutes even simple grasp. It likely stems from competition between the evolutionarily younger targeted corticospinal projections necessary for muscle individuation unique to primates and humans [17] vs. phylogenetically older divergent reticulospinal projections [19] so critical to brachiation or early tool use [14, 17] that would prevent muscle individuation. Therefore, testing whether a person can quickly ramp up a fingertip force vector in a same direction is a direct test of how well the CNS can individually control the intensity of the activation of a specific set of motoneuron pools [20][3]; and testing whether a person can quickly re-direct a fingertip force vector of a same magnitude is a test of whether the CNS can change the proportions of activations across motoneuron pools [12][4].

Seven young and healthy participants performed each task for at least 450 repetitions, over 5 days, emphasizing both speed and accuracy (see Methods). As in prior work [20–22], the participants exerted fingertip force against a rigid surface while we recorded the three-dimensional fingertip force as shown in figure 1.1, a computer monitor displayed the magnitude and direction of the fingertip force vector in real-time, and the target force magnitude and direction for each task. Participants were instructed to go the start target and hold, and then go to the end target as quickly as possible and hold. Figure 1.2 (left) shows that for the force scaling task, the ramp-up speed is defined as the time need to go from 1 N to 2 N, where less time indicates a higher speed. It is customary and acceptable in the context of speed-accuracy tradeoffs to use time as a measure of speed because the endpoints are fixed, and a shorter time is a linear measure of improvement[5]. The accuracy of force scaling was measured by the peak overshoot of the force magnitude (participants never undershot as they were maximizing their speed). Figure 1.2 (right) shows that for the force re-direction task, the speed is defined as the time need to rotate the force vector by 23° (from +0.2 to -0.2 radians with respect to the vertical), where less time indicates a higher speed. The accuracy was the RMSE of the force magnitude (i.e., the deviation from a constant, figure 1.2 (right)) during that same period. Participants performed each task during five days within a week, in 30 bouts

[3] I.e., how well the same muscles can be used at different level of activation, but keeping the proportions of activation similar.

[4] Changing the direction of the fingertip force vector requires changing at least the relative proportions of activation among them, and may even require changing the muscles being used.

[5] In addition, when using 1/time in N/s the reciprocal function 1/x introduces strong nonlinearities that unnecessarily complicate the statistical analysis.

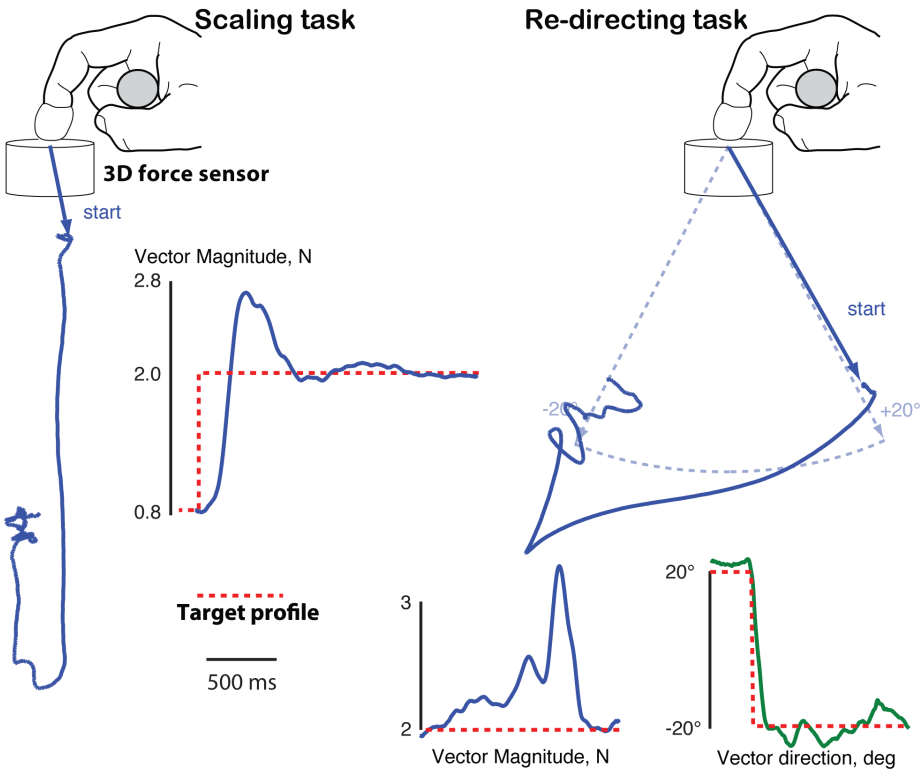

Figure 1.1: We used two different manipulation tasks, scaling and re-directing of the fingertip force, to analyze the fundamental building blocks of dexterous finger control. **Left:** The subject was instructed to exert an initial force of 0.75 N with its index fingertip against a rigid surface. Voluntarily it should ramp-up the force magnitude to 2 N as fast and as accurate as possible. The representative subject tried to follow the target profile (red, dashed line). Eventually it reached the target force but overshot the level at around 0.8 N. The spatial force trace pictures the same behavior. **Right:** The initial instruction was to start at a fingertip force of 2 N, where the direction was tilted at 20° towards the subject. Voluntarily the subject had to redirect the force in a way that the vector pointed 20° away from the subject, whereas the magnitude had to be maintained constant. The subject could very accurately follow the desired angular profile but failed miserably in maintaining the magnitude. The spatial trace of the force visualizes this behavior.

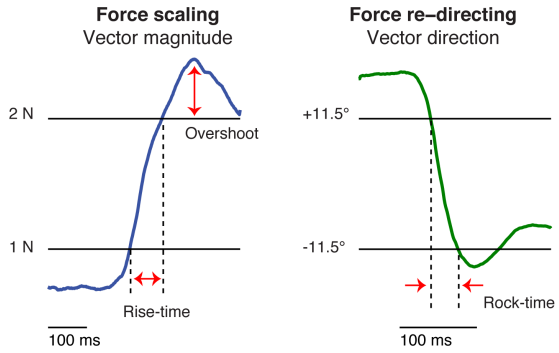

Figure 1.2: For both manipulation tasks, we measured a parameter for speed and accuracy each. **Left:** The rise-time, to change the force magnitude from 1 N to 2 N served as a speed indicator for the force scaling. We used the overshoot at the peak magnitude over the targeted 2 N as accuracy parameter. **Right:** The speed for the force re-directing was indicated by the rock-time to change the angular deflection from $-11.5°$ to $11.5°$ ($11.5°$ to $-11.5°$ in the opposite direction). The RMSE of the magnitude relative to 2 N during this time period served as accuracy parameter.

of three trials each day. The computer screen displayed the trace of their force vector in real time to show them their accuracy. From the second day onward we also gave them a letter grade (A best to D worst) immediately after the trial to encourage speed. Assigning a letter grade to their speed was critical to avoid emphasizing accuracy, but instead perform a fast ballistic motor task without on-line visuomotor control. Moreover, as shown in figure 1.3 (top left), we made the speed grading more stringent on the last two days of testing to further push the participants to their limits of ballistic performance. Over-training the ballistic (i.e., feedforward) performance of these tasks allowed us to quantify the ability of corticospinal motor drive to issue a "motor memory" to control the motoneuron pools of finger muscles. As described in the Methods, 450 trials in bouts of 3 trials distributed over several days is known to be sufficient to reach steady-states overtrained performance of motoneuron pools, especially for finger tasks [23–27]. As such, the repeated and consistent performance of our tasks is analogous to the execution well-trained everyday manipulation tasks that are naturally driven by predictive, feedforward strategies mediated by proprioceptive and tactile afferent signals, and not by visual, cognitive or auditory input (for a review, see [28]). Our multi-day testing regime shows that the force scaling task can be done well enough to exhibit a robust speed-accuracy tradeoff, but the force re-direction task can barely be done at all, let alone optimized (figure 1.3). Participants were able to scale the force magnitude without disrupting its direction, figures 1.2 (left) and 1.3 (top left), but did not improve their overall combined performance (defined as the slope of the sum of normalized speed and normalized accuracy, figure 1.3 (middle left), p=0.7929). In addition, they displayed a robust speed-accuracy tradeoff as per Fitts' law (figure 1.3 (bottom left), R2=0.68, [29]). This demonstrates that rapid scaling of fingertip force vector magnitude with the index finger in the direction of grasp is a well-established motor skill in adulthood. A robust speed-accuracy tradeoff

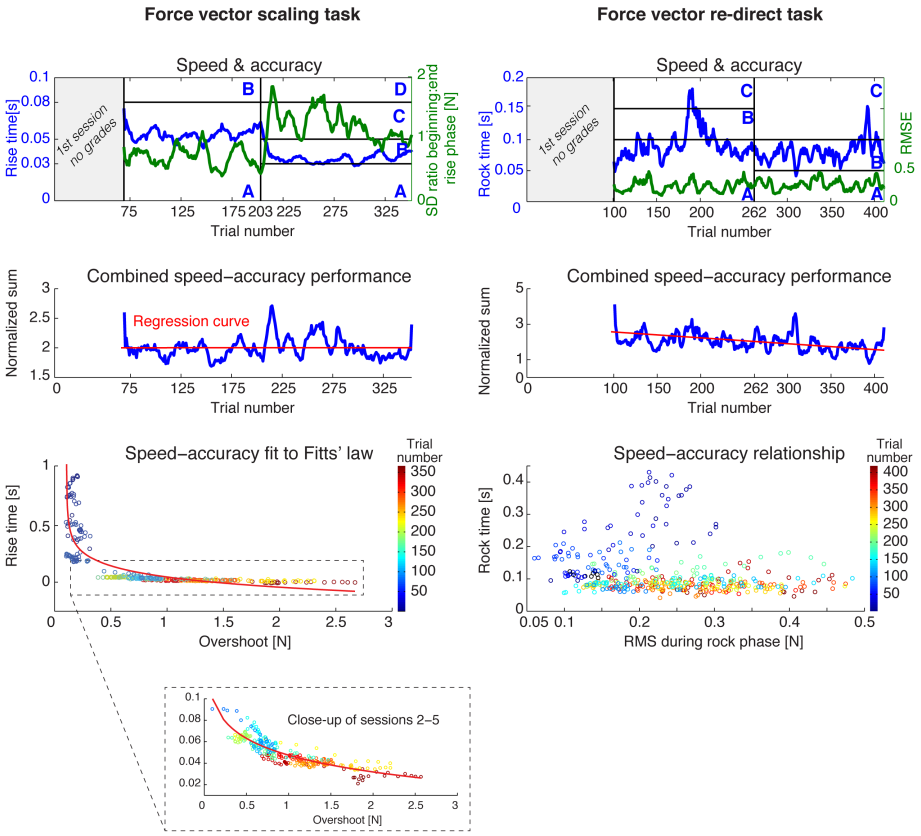

Figure 1.3: Typical experimental results of a representative subject for the force scaling task (right column) and force re-directing task (left column). **Top left:** The plot combines speed and accuracy in on graph. The black lines indicate the limits for the different grades. The subject could decrease the rise-time when the grading became more stringent between Sessions 3&4 (trial 203). Even though it was not able to further increase the speed. The gain in speed was immediately connected with a decrease in accuracy. **Middle left:** The graph shows a parameter, which combines the performance in speed and accuracy. The regression line visualizes that, even though there are fluctuations, the subject could not improve its performance. **Bottom left:** The tradeoff between speed and accuracy is color-coded corresponding to the trial number. With advancing training, the rise-time decreased, where as the overshoot increased. The data can be fitted very well to a Fitts' law like semi-log distribution with an $R^2 = 0.63$. The inset shows that the speed actually doesn't saturate during the training but maintains the distribution. **Top right:** As for the force scaling, the subject eventually reached a minimum rock-time for re-directing its fingertip force. In contrary to the latter case, the accuracy was not connected to the speed. **Middle right:** The decreasing regression line visualizes that the overall performance for this task increased statistically significant (p=0.0129). **Bottom right:** The scatter plot shows the speed-accuracy relationship for the force re-directing. Apparently the rise-time was not connected to the RMSE at all. Additionally the trial number didn't influence the position of a data point.

is the hallmark of motor performance with enough neural degrees of freedom to seek to optimize the feedforward drive to multiple motoneuron pools [29–31]. In contrast, the force-redirection task showed quite the opposite. Participants were not able to rotate the fingertip force vector without also disrupting its magnitude, figures 1.2 (right) and 1.3 (top right), and as testing progressed their performance showed improvement (p=0.0129), figure 1.3 (middle right). In addition, they displayed no tendency towards displaying, or even developing, a speed-accuracy tradeoff, figure 1.3 (bottom right). The application of the nonparametric Spearman's rank correlation coefficient, which detects trends in datasets with minimal assumptions, to the speed-accuracy tradeoff clearly contrasts the differences across tasks. The mean value of -0.78 for the scaling task indicates a clear negative trend of the speed-accuracy tradeoff, whereas the mean value of 0.005 strikingly points out that there is no relation at all between speed and accuracy for the re-directing task, figure 1.4. This leads us to conclude that

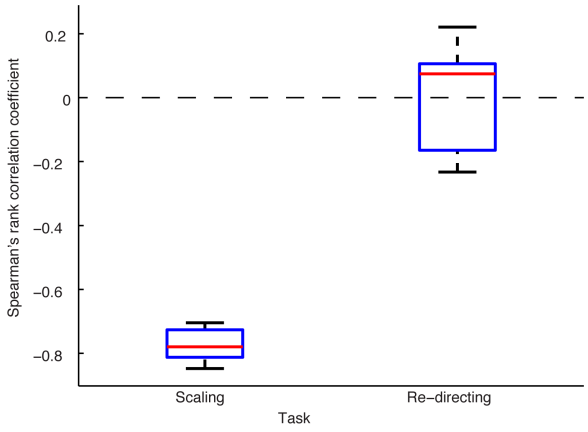

Figure 1.4: Spearman's rank correlation coefficient reveals trends in datasets. A value close to zero indicates no existing trend, where as values close to ± 1 indicate increasing or decreasing behavior. The box plot proves that the tradeoff between speed and accuracy for the force scaling is strongly negative ($\rho = 0.78 \pm 0.05$), where as there is no connection between the parameters for the re-directing (($\rho = 0.05 \pm 0.17$). These results are highly significant (p=7.5E-8).

the that rapid re-direction of a fingertip force vector is not a well-established motor skill in adulthood, and in fact lacks sufficient neural control degrees of freedom to properly plan and execute-let alone optimize-the changes in neural drive across individual motoneuron pools.

Our results for the re-direction of fingertip force demonstrate that the evolutionary adaptations for individuated control of finger muscles for this fundamental task are far from robust or complete, and result in concrete and substantial functional limitations. Our experimental design mitigated potential confounds and allows us to attribute these functional limitations to the phylogenetic immaturity of the individuated control of hand muscles-most likely at the level of the spinal cord. Firstly, by studying isometric finger tasks in the middle of the range of motion we ensured that (i) the anatomical interconnections among tendons and muscles do not pollute the production of fingertip forces, and (ii)

do not invite nonlinearities in force production due to force-length and force-velocity properties of muscle. Secondly, we defined the target force levels of both tasks to be very low (between 1.9 N to 2.1 N, respectively around 9% of maximal force [12, 20, 32] specifically to be compatible with low force levels typical of light everyday manipulation tasks [33], and heavily favor rate coding as the mechanism of force modulation (i.e., little if any recruitment of additional motoneurons [34]). This is critical as it rules out activation-contraction dynamics [35] and signal dependent noise [36] as leading causes of the disruption of fingertip force vectors. In fact, the relatively better performance of the scaling task shows that the poorer performance for the re-directing task cannot be explained by rate-coding, activation-contraction dynamics or signal dependent noise as in both cases similarly low levels of force are used. In addition, our mathematical models, fine-wire recordings, and anatomical studies show that the muscle coordination patters needed to re-direct the fingertip force vector of low magnitude in that range from $+20^\circ$ to -20° that we tested do not require the turning on or off of multiple muscles [37, 12, 20, 38]. That is, there exist biomechanically feasible coordination patterns that can re-direct the force vector without turning any muscles on or off, especially after multiple days of testing, that the CNS could use if it had individuated control over motoneuron pools. In addition, fixed synergistic interactions among muscles are also not likely the potential explanation: we have recently reported the absence synergistic muscle activations when recording fine-wire EMG during similar tasks [39]; and others have shown the index finger to have detectable, but not functionally significant, correlation across motoneuron activity during isometric forces of low magnitudes [4]. Thus our results strongly suggest that it is the inability of the CNS to dynamically modulate the relative activation across already active motoneuron pools (i.e., muscle individuation) that is the likely cause of the inability to re-direct fingertip forces accurately.

Having ruled out muscle-physiological confounds, one may appeal to cognitive load or perceptual differences between our two tasks as the impediment to the independent control of fingertip force magnitude and direction. However, Prinz and colleagues [40], have shown that such perceptual limitations to motor function apply when the task is primarily a bi-manual visual or auditory one, as during asynchronous drumming or cyclic motion. In addition, as mentioned above, the index finger together with thumb enjoys the greatest neuromechanical independence, its motoneuron pools are least susceptible to enslaving and contextual weakening [6], and thus fixed motor primitives in the traditional sense [40] do not apply. Moreover, we chose our uni-manual tasks specifically because the need to re-direct fingertip forces is a ubiquitous, everyday task that is almost exclusively executed under proprioceptive feedback [10, 28] with clear functional consequences (i.e., objects should not be dropped during manipulation), and our multi-day testing paradigm was designed specifically to retain that independence from visual feedback and the establishment of a predictable feedforward motor command.

A simpler explanation exists, which is strongly supported by other work and clinical experience. It is that there is no a-priori reason to believe that the evolution of individuated muscle control should be complete or robust in modern humans. As a consequence we see that, contrary to our everyday assumption, the human hand might have barely enough neuromechanical (as opposed to strictly mechanical) degrees of freedom to be able to individually regulate the activity

across motoneuron pools of the fingers. Scaling the fingertip force vector only requires regulating the common drive to all muscles in a specific combination [20]. Redirecting the force tasks, however, reveals functionally relevant limitations in the speed and accuracy with which the CNS can change the proportions of activations across muscles (i.e., their individuation)-which likely comes from competition within and among cortico-spinal projections and interneuronal interactions in the spinal cord [18]. Such interpretation is supported by the detection of the inability of the CNS to execute apparently simple finger tasks in other highly functional contexts. These include the ability to individually move the fingers during static grasp [18], measurable errors in the transition from the control of motion and force-or the combination of motion and force-[22, 21]. In addition, our results clearly show that the CNS has does not have the ability to individually control of changes in activations across motoneuron pools even for the simple ecological tasks of fingertip force production. These independent lines of evidence strongly suggests that the human hand operates on the verge of failure because it has barely enough neuromechanical degrees of freedom to meet the multiple simultaneous mechanical demands of even simple ecological tasks.

This conclusion raises the paradox of why these limitations in manipulation do not seem apparent in everyday life. In reality, however, examples of this lack of individuation and robustness in the control of hand musculature are all around us. Primary among them is the clinical reality that fine control of hand musculature operates on the verge of failure because manipulation is so susceptible to even mild neurological conditions, and seldom recovers full functionality after stroke, cerebral palsy, spinal cord injury or many other neuropathies as cited above. In fact, clinicians know that much functional recovery is due to redundancy and adaptations at the task level [41], This is likely due to the fact that muscle redundancy does not imply robustness to muscle dysfunction [38]. That is, patients adapt to their diminished neural control of hand musculature by using task-adaptive strategies that change the kinematics, use other grasps, tools or success criteria. From the developmental perspective, fully mature and dynamic dexterous manipulation is one of the motor functions that continue to mature into late childhood and adolescence [42, 43], likely because the neural structures necessary for manipulation at the level of the brain [44, 45] -and thus also likely at the level of the spinal cord-take years to develop [46]. So much so that true prestidigitation and manual skill for musical performance or surgical technique are uncommon and admired skills that depend on constant practice. From a more ecological perspective, we propose that the design of objects and tools is in fact an example of socio-biological coevolution where cultures promote the development of tools in general [3], but inevitably favor those designs that are more robustly or easily handled given the inherent limitations of the human hand. This is the realm of ergonomic design, such as the recent evolution of the computer mouse or touch screen, but older examples abound such as the knurling of precision surgical tools, frets for string instruments, the zipper to replace buttons, or the lever door handle to replace the round one. This socio-biological co-evolution even targets subsets of the population to accommodate for development or aging such as connected chop-sticks or Velcro shoe closures for children and older adults-let alone assistive technology for individuals with disabilities. Thus our apparent versatility during manipulation is as much a function of adequate-but incomplete and non-robust-neuromechanical

evolutionary adaptations, as of socio-biological co-evolutionary design of objects and tools to make them forgiving when manipulated.

1.1 Materials and Methods

1.1.1 Experimental Setup

Subjects wrapped the thumb and unused fingers of their dominant hand around a fixed horizontal dowel to produce force against the flat surface of a quadratic pedestal, mounted on a six-axis load cell (model 20E12A-I25; JR3, Woodland, CA). The subject's forearm was supported by a vertically adjustable armrest. Armrest and load-cell were attached to a magnetic optical plate and vertical and horizontal positions were adjusted such that the index finger was in ad-abduction posture. Subjects wore a custom-molded thimble (thermoplastic) with a spherical Teflon ball embedded in its tip. The custom-molded cover (i) helped removing the discontinuity at the fingernail and (i) defined a unique contact point and friction cone for force direction. The force plate was coated with 300 grit sandpaper to avoid slipping. The force signal was sampled at 1000 Hz and recorded with a data acquisition card (PCI-6225, National Instruments Corporation, Austin, Texas). A Matlab-GUI stored the acquired data and provided the subjects real-time visual feedback on speed and accuracy.

1.1.2 Experimental Procedure

Each subject was trained in two different tasks during 5 sessions on 5 different days. For the force scaling the subjects were instructed to start from a preloaded condition (0.75 N magnitude) and ramp up their force to 2 N as fast and as precise as possible, where they should maintain the magnitude for 5 s. One trial consisted of three repetitions. For the second experiment the subjects had to start from a preloaded condition (2 N force magnitude, $+20°$ force deflection from the vertical axis in distal direction) and tilt the force vector twice to $+20°$ in proximal direction and back, while maintaining a constant magnitude. The subjects were instructed to maintain the force magnitude and direction inside the target areas for a short time to clearly separate re-direction phases. The physical and mental state of the subject was monitored daily by means of a questionnaire, which reported concentration, stress, recovery as well as motivation before and during the experiment.

Experiment 1: Force Scaling

Two distinct processes drive motor learning. A process with weak response to error retains information well and is largely diminished after 400 trials, whereas the other process responds strongly to error but has poor retention (vanishes after 30 trials). 30 trials were sufficient to train people in reaching a target with their fingertip. A similar study used 252 trials, whereas subjects satisfactorily learned arm reaching movements within 144 to 192 repetitions We used 30 trials per session (3x30x5=450 repetitions) in order to minimize both learning processes and reach steady-state overtrained performance. Session 1 was performed without grading on speed to get used to the task, while we used different grading schemes for sessions 2 and 3 and sessions 4 and 5, respectively. Grading

schemes for sessions 2&3 (transition time from 1 N to 2 N; school grades; t < 80 ms → A, 80 ms < t < 150 ms → B, 150 ms < t < 250 ms → C, t > 250 ms → D) to mitigate the possibility of supraspinal modulation of the descending motor command or visuomotor delays and (t < 30 ms → A, 30 ms < t < 50 ms → B, 50 ms < t < 80 ms → C, t > 80 ms → D) for sessions 4&5 to reach the neurophysiologic limit. Every fifth trial for sessions 2-5 was performed without visual feedback, which served as control trials to ensure strictly ballistic action.

Experiment 2: Force Re-Directing

Likewise to the force scaling, session 1 was performed only with feedback on accuracy, while sessions 2-5 were also graded on speed with two different schemes for sessions 2&3 (transition time from -20 radians to 20 radians or vice versa; school grades; t < 100 ms → A, 100 ms < t < 150 ms → B, 150 ms < t < 250 ms → C, t > 250 ms → D) and sessions 4&5 (t < 50 ms → A, 50 ms < t < 100 ms → B, 100 ms < t < 150 ms → C, t > 150 ms → D). Every fifth trial for sessions 2-5 without visual feedback.

1.1.3 Data Analysis

Data was processed off-line and manually for each trial using custom written programs in Matlab (version R2011b, The Mathworks). A low pass filter with a cut-off frequency of 30 Hz was applied to all three force axes. While we processed all sessions, regression- and statistical analysis was only performed on the speed-graded sessions 2-5. We calculated a parameter for speed and accuracy for both experiments. Trials were excluded from the analysis when they grossly missed the target profile, e.g. due to slipping, distraction, etc. (98.6 % inclusion for scaling and 94 % for tilting). We used the sum of the normalized parameters $z = \frac{T}{\bar{T}} + \frac{E}{\bar{E}}$ as a combined performance indicator, where T represents the respective speed and E the accuracy parameter and the overlined parameters denote the mean across all trials. The normalization scales both paramaters to comparables magnitudes. Since we wanted to minimize speed as well as accuracy, a low value of the parameter z indicates a good performance. High values for T or E increase z and report bad performance. To apply appropriate speed-accuracy tradeoff models to the data we used Matlab's function nlinfit and reported Spearman's rank correlation coefficient $\rho = \frac{\sum_i (E_i - \bar{E})(T_i - \bar{T})}{\sum_i (E_i - \bar{T})^2 \sum_i (T_i - \bar{T})^2}$ to measure the statistical dependence between the parameters. For all the parameters indicating speed, accuracy and performance of the two experiments we used repeated measures ANOVA to test for differences across means of each session (4 levels). We verified that the residuals were normally and identically distributed, used Mauchly's test to rule out violations of the sphericity assumption and the means of successive sessions were contrasted to perform a trend analysis. We processed the blind trials in exactly the same way as the sighted trials. For both experiments we selected the best three repetitions in the first fifteen and the last fifteen repetitions for the parameters speed, accuracy and performance. We calculated the means across all subjects for the first and the last repetitions and compared them to make sure that the results are comparable to the sighted trials, even without immediate visual feedback. Additionally we looked at the evolution of rise time and rock time during the practice to

verify that the subjects are conditioned to perform the tasks in a time scale compromising closed-loop control and ensuring ballistic action.

Chapter 2

Additional Materials

The manuscript of the paper presented in the last section is addressed to the Journal of Neuroscience and presumes that the reader has specific previous knowledge. This chapter presents additional material to the manuscript and its sections have distinct purposes. The theory-block in section 2.1 gives an extended overview about the neuroscientific knowledge neccesary to understand the presented study. Section 2.2 describes the detailed experimental setup of the study. The section should facilitate the reader with the possibility not only to reproduce the experiment, but to use and adapt the hardware and the software written for the study. In a last section, we will present additional experimental results, which we weren't using in our publication.

2.1 Theory

Sherrington called the α-motoneuron the final common path [47]. All the signals converging from several descending tracts as well as afferent sensory information are somehow integrated in the motoneuron, which conducts the appropriate signal to the muscle. This chapter gives an introduction to the most important paths to control fingers and to spinal circuitry, which modulates descending signals or produces rhythmic outputs. Finally, different control strategies will be discussed. If not specifically indicated, the information in this section originates from [48, 49] and [15].

2.1.1 Descending Motor Paths

The motor system is organized hierarchically. Even on the lowest level of the motor hierarchy, spinal reflexes demonstrate sophisticated neural processing. However, these reflexes can be modulated by higher levels in the hierarchy and most motor commands originate from the cerebral motor cortex, the cerebellum or various brain stem nuclei. The brain stem modulates the action of spinal motor circuits, while the motor cortex acts on both, brain stem and directly on the spinal cord (see figure 2.1). The motor system is not a simple chain from higher to lower level areas. Many pathways allow the different levels to communicate with each other. Consequently, damage to one part does not cause paralysis, since there are always multiple paths between the parts of the system. Damage

to higher levels results in deficiencies in motor planning, initiation or coordination, but movement is still possible.

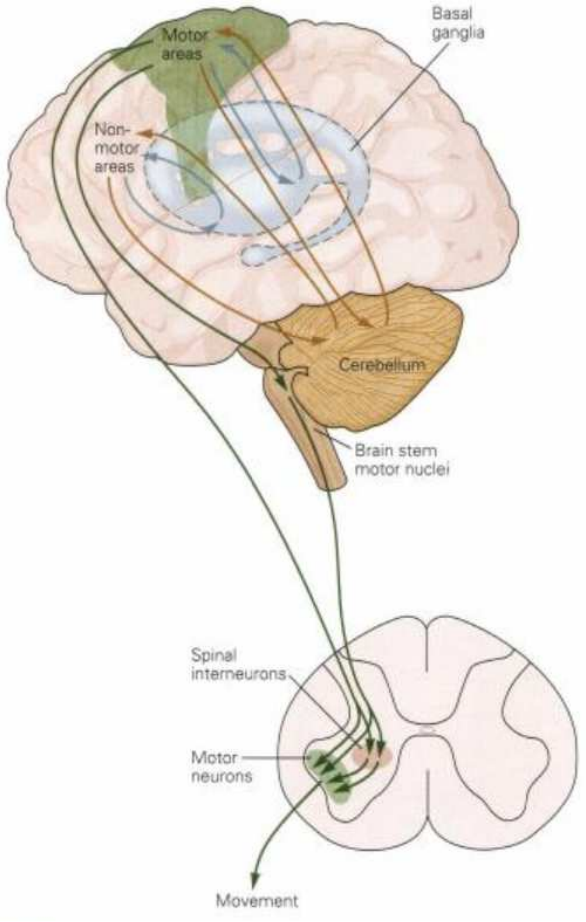

Figure 2.1: The motor systems have three levels of control: the spinal cord, the brain step and the cortex. They are organized both serially and in parallel. Figure adapted from [48]

Descending motor paths are defined as those, which originate in the brain and initiate or modify motion. They end in the spinal cord and innervate α-motoneurons, γ-motoneurons or interneurons. The motoneurons innervating the limb muscles are located in the anterior horn of the spinal cord and arranged according to two rules (see figure 2.2). Motoneurons that innervate flexor muscles are located posteriorly to the ones innervating extensor muscles and motoneurons that innervate distal muscles (e.g hand and finger) are located lateral to the neurons innervating proximal muscles.

Several paths bundle fibers originating in the brain and guide them to the spinal cord. The most important ones for arm and hand movement are the rubrospinal and corticospinal or pyramidal tracts. The corticospinal tract originates in the motor cortex and is the most important mediator for voluntary

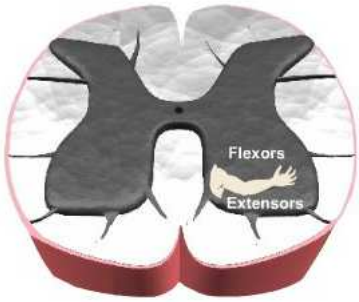

Figure 2.2: Flexor-extensor rule and proximal-distal rule the position of the motoneurons in anterior horn.

movements in humans. Approximately 90% of the axons in the tract cross over to the contralateral side at the pyramidal decussation, forming the lateral corticospinal tract. These axons run through the lateral funiculus of the spinal cord, before they synapse either directly with α-motoneurons or with interneurons in the anterior horn. The remaining 10% of the axons that do not cross at the decussation constitute the anterior corticospinal tract, which finally crosses over to the contralateral anterior horn in the spinal segment at which they terminate. While the latter is responsible for the control of the proximal muscles, the lateral corticospinal tract innervates the distal musculature with the important function of the fine control of hand and fingers. The corticospinal tract is the only descending pathway in which some axons make synaptic contacts directly onto α-motoneurons.
The rubrospinal tract originates in the red nucleus of the mid brain and innervates spinal neurons at all levels of the spinal cord. The rubrospinal tract is an alternative, by which voluntary motor commands can be sent to the spinal cord. The rubrospinal tract plays a minor role in humans than in other mammals but is amongst other things responsible for fine motor control of the upper limb. The red nuclei receive most of their input from the cerebellum and are therefore important for trained movements. They also receive some input from the motor cortex and can function as an alternative for a damaged pyramidal tract, which gives them higher significance for humans.
As mentioned before, corticospinal neurons make direct connections with α-motoneurons in the spinal cord. This connection is one of the mechanisms that permit individual movements of the fingers. Although there are redundant paths to actuate the fingers, the monosynaptic connection is salient for precise control of the finger muscles. Monkeys with sectioned pyramidal tracts are still able to climb, jump and execute normal movements. Though, they loose the ability to individually control muscles [14].
Another important descending path is the reticulospinal tract. Two tracts descend from the reticular formation to supply the trunk and proximal limb muscles. The reticular formation coordinates automatic movements from locomotion and posture. In addiation, it faciliates or inhibits voluntary movements and influences muscle tone. The reticular tract is phylogenetically older than the corticospinal tract. Compared to the direct and simple projections of the latter,

the neurons descending from the reticular formation form much more divergent projections, which compete with the corticospinal projections in individuated control uf upper limb muscles.
Like redundant descending paths, the corticospinal neurons also form indirect connections through several interneurons with motoneurons in the spinal cord. These connections regulate a larger number of muscles and may contribute to multi jointed movements such as reaching or scratching. These spinal circuits will be discussed in the next section 2.1.2.
The individual movement of fingers is controlled by patterns of activity in a population of cortical neurons. Though monosynaptic projections from the motor cortex allow precise control of the fingers, it is difficult to identify the contribution of single neurons. On the one hand, a single corticospinal neuron can control multiple muscles and on the other hand a single muscle is actuated by a crowd of neurons. Not single neurons, but the relative contribution of a population of neurons codes for the direction of a movement [50]. Even more surprising, Lemon and Muir have shown that different cells innervating the same muscle can cause different activation [51]. While specific cells excite high forces, there are other neurons that precisely control the force.
Conclusively, the descending path from the motor cortex through the corticospinal tract, over α-motoneurons to the finger musculature is surprisingly direct and enables immediate control. However, the contribution of cortical neurons to force magnitude and direction is inscrutably complex. And since the corticospinal neurons also synapse on spinal interneurons, circuitry in the spinal cord makes the control of finger musculature even more intricate.

2.1.2 Central Pattern Generators

Though the path described in the previous section 2.1.1 is astonishingly simple, voluntary movements are governed by complex motor programs. Supposedly a movement is represented in the brain in some abstract form rather than as a series of joint motions or muscle contractions. Our handwriting e.g., appears similar, regardless of the size of letters or the body part used to produce the letters [52].
It is still very little known about the highly integrated neural circuits in the brain that coordinate movements. Yet new experiences seem to show that highly coordinated and very complex systems of interneurons regulate the precise timing and sequencing of muscle activity on the level of the spinal cord. The central theory is that these interneurons form pattern generators within the central nervous system which produce the basic motor program. There are neuronal circuits within these pattern generators which coordinate muscular synergies and generate timing signals. Command neurons activate these pattern generators when a particular coordinated movement is required. The pattern generators mainly initiate rhythmic and cyclic behavior such as chewing, swallowing, scratching and of course locomotion. There is strong evidence that such spinal circuits also exist for finger control. Pattern generators in turtles that are shared by three different kinds of scratching as well as by locomotion have been identified [53]. Even more significant for our research is the use of biomimetic pattern generators to model rhythmic motions such as finger tapping in humans [54]. The model proved to be applicable for the evaluation of abnormal movements in humans with motor function impairments such as Parkinson's disease.

2.1.3 Feedback and Anticipatory Control

Voluntary movements are influenced by environmental obstacles and perturbations. They improve with practice, as one learns to anticipate and correct the movement for them. The nervous system monitors sensory inputs and uses this information to react with a moment-to-moment control. The so-called feedback control compares the sensory inputs with a reference signal and immediately corrects the output signal in order to follow the reference signal (figure 2.3 (A)). Obviously, such a mechanism further complicates the neural circuits and causes delays in reaction. The inclusion of visual feedback into a control-loop produces delays of more than 180 ms [55].

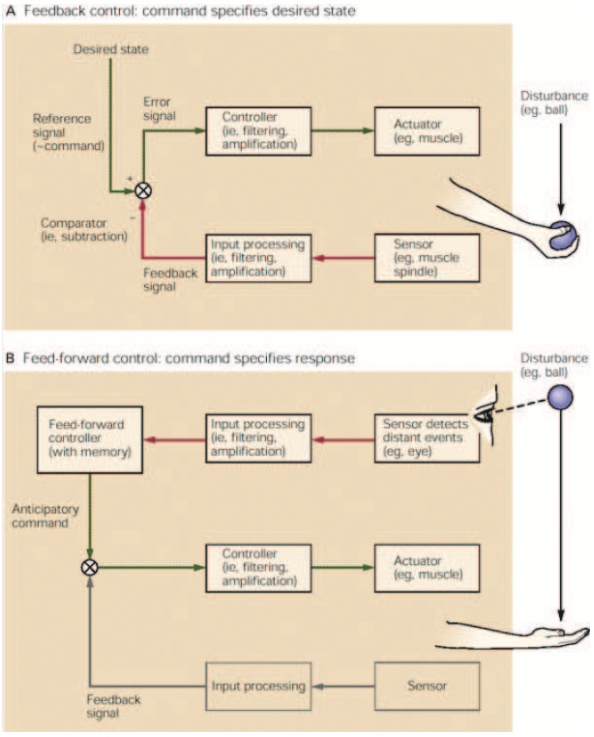

Figure 2.3: (A) In a feedback control system, the sensory input is constantly compared with a desired output state and the system instantly corrects for errors. Feed-forward control e.g. is used when catching a ball as soon as the ball approaches the catching hand. Visual feedback and information about muscle stretch is used to maintain and hold the hand in a given angle. This method of control is very precise but causes substantial sensory delays. (B) In contrast, feed-forward control integrates sensory information before the execution of certain movements. Since the control is anticipatory and relies on experience, the actual execution can be very fast. Feed-forward control is applied when catching a ball to estimate the time and location, where the ball hits the hand. Figure adapted from [48].

Unlike feedback systems, feed-forward control acts in advance of certain perturbations (figure 2.3 B). Experience helps us to adjust to environmental obstacles before we execute a movement. It is widely used by the motor system

to control posture and movement. Two key principles of feed-forward control are that it is essential for rapid action and depends on the ability of the nervous system to predict the consequences of certain events. The latency for monosynaptic feed-forward-control can be less than 40 ms.
Very fast feed-forward controlled movements with high acceleration and deceleration phases are referred to as ballistic movements. EMG recordings of such movements demonstrate that the triphasic (agonist/antagonist/agonist) muscle activation begins with a brief agonist motor unit activation signal with firing rates of 60 to 120 Hz that may last for 100ms and occurs 50 to 100ms before the movement begins [56, 21]. The function of the succeeding antagonist muscle contraction is believed to control the amplitude and timing of ballistic movements. The second agonist muscle activation is suggested to terminate the negative acceleration of the antagonist muscle contraction and thus the ballistic movement. In humans, ballistic movements involve spontaneous propulsion of the limbs, observed e.g. in martial arts [57] but also fast finger movements, required to type a text or play the piano.

Inspite the thorough introduction to our hypotheses in the paper, we would like to justify our research here again under a different point of view. We were using and exercising our fingers since our childhood, be it playing with a mobile, learning to knot our shoes, practicing a first instrument such as the piano or the guitar or even the handling of a computer with mouse and keyboard. Each of these activities is a complex interaction of scaling and redirecting the fingertip forces. As explained in the last sections, the possibilities to control our fingers range from simple monosynaptic connections to specifically change the activation of a few muscles to complex spinal circuitry to control and coordinate multi-jointed movements. Despite the fact that we were extensively exercising our fingers over the years, we represent the opinion that our ability of manipulation is far away from optimal. We would like to know what kind of circuitry is involved when sending motor commands to the fingers, if the task difficulty is reduced to a minimum and the computational load due to sensory feedback is minimized. Hence, we wanted to test the following two hypotheses.

- Scaling the force magnitude only requires little change in muscle activation. Therefore, humans should be able to perform this task very accurately and fast. Overtrained subjects would eventually reach an optimal tradeoff between speed and accuracy.

- In contrary, redirecting the force vector requires a more complex change in muscle activation. Even extensive training can't enable precise and fast reorientation of the force and will always be connected with erroneous force magnitude.

To pick up Isaac Asimov's quotation "*Writing, to me, is simply thinking* ***through*** *my fingers*", writing requires not only a lot of thinking to compose a stimulating text. Additionally, the simple action of writing letters is thinking **for** the fingers.

2.2 Experimental Setup

2.2.1 Detailed Hardware Setup

In the paper, we described the hardware setup only very schematically. Here, we show and explain construction drawings of the different parts. In addition, we present a short manual to mold the custom thimbles.
The arm of the subjects was supported by the armrest shown on the CAD-drawing in figure 2.4 (A). The supports were attached to the optical plate with four magnetic bases. The height of the actual supporting plate was adjustable. The horizontal dowel was attached between the two supports in front. A foam pad on top of the supporting plate made the construction more comfortable.
The force plate is shown in figure 2.4 (B) and manufactured by 3D FDM printing. The large area simplified the positioning of the setup and as we will show in section 2.2.3 the recording was consistent all over the plate. The lower part of the plate was screwed to the JR3 load-cell, which was screwed on its part to another magnetic base. The sensor was connected to the computer as described in the paper and operated in single-end mode.

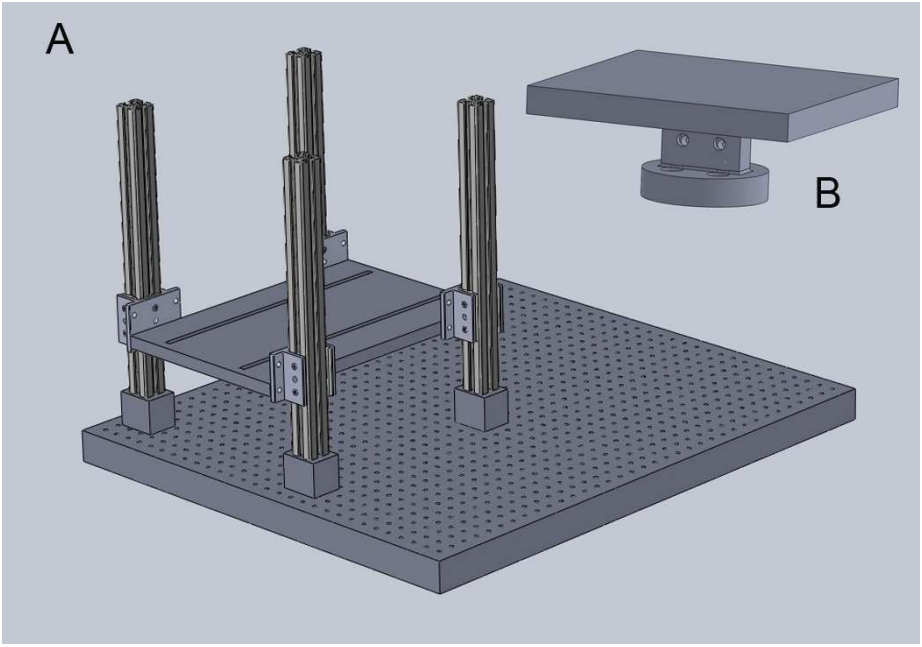

Figure 2.4: CAD drawings for the setup used in the tapping study. The armrest (A) served to position and fix the subject's forearm. The force plate (B) was mounted on top of a load-cell and was used as force transmitting component.

We used thermoplastic sheets to form the custom thimbles. Cruciform pieces were cut out of the sheets and were boiled in water to heat them to their glass temperature, where they become easily deformable. The center part of the cross was placed on the subject's fingertip and they were asked to wrap the four wings around the index finger. During the cool down process, they pressed the thimble against a small Teflon ball to prepare a dip. After the thimble had completely

cooled down, we used super glue to attach the ball to the thimble. Picture 2.5 shows the assembled parts and a subject waearing its thimble.

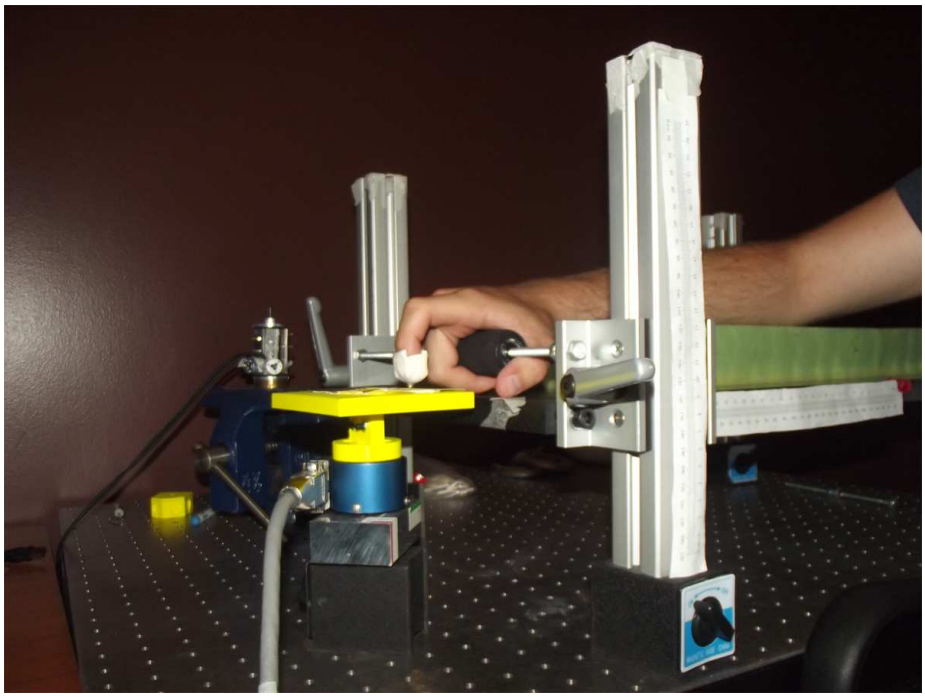

Figure 2.5: Experimental setup for the finger tapping study.

2.2.2 Software Implementation

Using Google-Spreadsheets as a Portable Database

Code of practice in the Brain-body dynamics lab is to use a 32-bit version of Matlab and the corresponding legacy-interface of its data acquisition toolbox. However, 64-bit computers are industry standard for several years and it's favorable to make use of their computational power. For this study and for the calibration project in chapter 3 we used the 64-bit version of Matlab and the newer session-based interface to acquire data. The big drawback of the newer interface is that it is only available for National Instruments DAQ-devices. In return, the usability improved a lot and the interface offers a variety of new features such as counter functions or accelerometer and temperature sensing.
Based on the interface, we were implementing a generic Matlab-class that allows to conveniently address arbitrary load-cells, whose data is digitalized with a NI-DAQ. The functionality of the class offers time-limited data acquisition, which blocks the code execution, and continuous background acquisition that allows Matlab to do intermediate execution of different commands. The class also accepts user-written handling routines, which either plot the data in real-time or process the data already while the measurement is running. Appendix A.1 lists the code for the whole class, while the following listing shows an example of

how to use it.

```
measurement = loadCell('2563',true,'Dev1',[1:6]); %create object
measurement.startBackgroundScan(0.1,@plot); % start a background ...
    scan
pause(5); %system delay does not interrupt data acquisition
[data,time] = measurement.stopBackgroundScan; %stop acquisition ...
    and get data
fancyDataAnalysis(time,data);
```

The constructor accepts four arguments: One specifying the load-cell, then a boolean variable deciding whether the output should be raw voltages or force data, third the data acquisition device to use and the fourth parameter specifies the input channels (six in this case). The second line starts a background acquirement, which uses the basic plot command to display the data every 100 ms. A system delay does not affect the measurement and after the break the acquirement is stopped and the data is returned.
For the rest of this section, I'd like to turn the reader's attention to the first parameter of line one. In this little example we used a JR3 load-cell with the serial number 2563. Google offers a variety of practical online services, amongst others an office suite and in particular a spreadsheet editor. We are maintaining a database with all load-cells available in the lab, which contains links to datasheets and the current application. More importantly, the database contains a separate sheet for every load-cell, showing only its calibration matrix (see section 3.1.4 for details). The load-cell class automatically downloads the calibration data from the database. It serves as a central storage and makes a local copy of the data superfluous.
We used the same principle to maintain the database of subjects. A Google-spreadsheed contained the subject number as well as all relevant parameters like age, handedness or experimental progress and allowed us to flexibly access and edit the data on different workstations. The necessary subject data could easily be read out and updated by Matlab while performing the actual experiment.
The only drawback of the method is that the communication framework between Matlab and Google-Docs is considerably slow. Nevertheless, since the download usually only happens during the start-up, there is no need for realtime access and the use of online spreadsheets is a good approach to store data.

Providing Visual Feedback

As described in the previous section, the used framework to acquire data from a load-cell offers a simple way to visualize the data during the measurement. The visual feedbacks for the two experiments have already been briefly described in the paper in chapter 1. This section describes the implementory and mathematical details.

Figure 2.6 shows a snapshot of the continuous visual feedback for the force scaling (A) and the re-directing (B). The displayed vector for both experiments represents the same data. While the length of the arrow maps the force magnitude, the angle to the vertical axis reflects the angle between z-axis of the load-cell and the deflection of the vector in the proximo-distal axis of the subject. At the beginning of the figure update, the mean of all individual components of the force vector is calculated. The length of the vector is determined by calculating

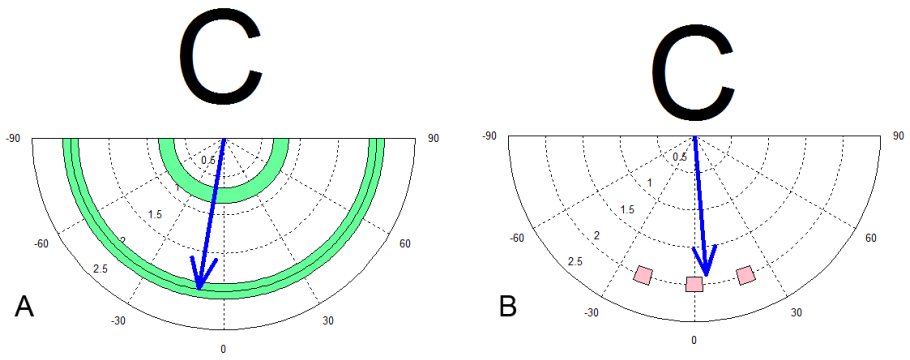

Figure 2.6: Visual feedback used for the two experiments in our study. (A) Feedback for force scaling, the shaded regions are green as long as the force magnitude is held inside the outer shaded region. (B) Feedback for force re-directing, the shaded areas are green as long as the force vector is held inside one of the three boxes, orange whenever the force magnitude is constant and red otherwise.

the euclidean length of the force components (equation 2.1) and the angle θ is calculated with simple trigonometry. These two parameters are mapped to the figure according to equations 2.3 and 2.4.

$$F = \sqrt{F_x^2 + F_y^2 + F_z^2} \tag{2.1}$$

$$\theta = arcsin(\frac{F_x}{F_z}) \tag{2.2}$$

$$L_x = sin(\theta) \cdot abs(F) \tag{2.3}$$

$$L_y = cos(\theta) \cdot abs(F) \tag{2.4}$$

After calculating the components of the average status vector, the software determines whether the subject maintains the force inside the desired area or not. The confidence interval for the force magnitude is $\pm 0.1N$ and $\pm 3°$ for the angular deflection. The deflection is also displayed for the ramping, but not reflected in the color coding. All target areas were green as long as the force vector was hold inside one of the areas and red otherwise, additionally the areas became orange for the re-directing as long as the magnitude is $2N \pm 0.1N$ but outside the desired angles.
The update interval was empirically chosen and set to 150 ms. The computation of the visual feedback was quite time consuming and didn't allow for a higher frame rate in order to avoid lags. However, since we looked at feed-forward control, there was no need for a totally smooth feedback. The update rate of about 7 frames per second is enough to provide sufficient feedback between the trials and to maintain a constant force vector.
The subjects were graded on their performance in real-time during the experiment. The grades ranged from A to D and reflected the time to either ramp up the force from 1.1 N to 1.9 N or re-direct the force vector from $-12°$ to $12°$. The most obvious approach to trigger the time measurement would have been derivative based, since it hadn't included only a part of the movement but determined the time from the beginning until the end of the movement. However

the time derivative of force magnitude and deflection turned out to be too rough and noisy to make reliable measurements.

Performing Experiments Using the GUI

This paragraph serves as a short manual for future investigators to use the software written for this study, though it is expected that the reader has already familiarized with the code. Another prerequisite for the software to work is to have the load-cell database and the subject database available in the personal Google-docs folder.

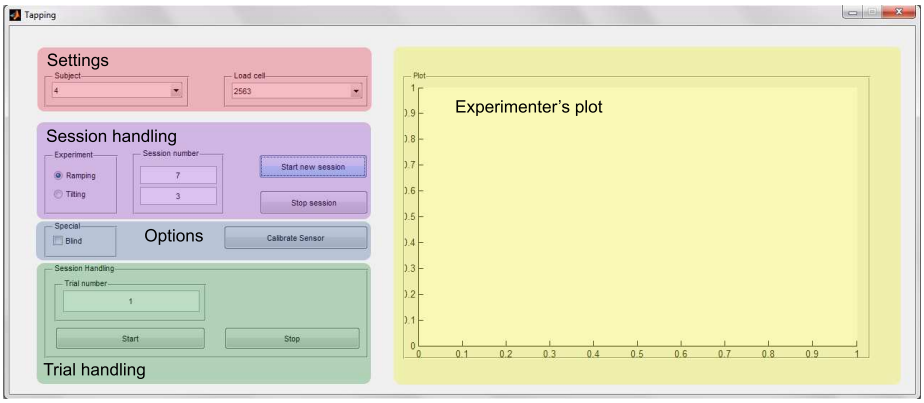

Figure 2.7: A screenshot of the Matlab GUI to control the scaling and re-directing experiments. The order of commands is from top to down and the different shaded regions indicate control groups.

Enter *Tapping* in Matlab's console to start the GUI. Before the actual GUI will open, you will be asked to enter your Google-credentials. After that it will take around 30 seconds to download all relevant information about the available load-cells. The operation of the software is very straightforward and an example is shown in figure 2.7. The experiment is set up from top to down. First off all you will have to choose the subject and the load-cell (we used the JR3 2563 for our study). Both, calibration and subject data are downloaded from the Google repository. The session counter in the middle of the purple section displays how many sessions of each experiment the subject has already performed. The radio buttons let you choose between the experiments. The next action will be to start a training session with the corresponding button. It is up to the user how often the load-cell baseline is measured (in our study, we did it after every 10th trial), but the action should be done at least whenever starting a new session. There is also an option to perform a trial without visual feedback. A click on the button *Start* triggers the measurement and opens the visual feedback for the subject on a secondary monitor. The trial counter will be incremented and a trial has to be terminated manually by a click on the corresponding button. There is possibility to add an additional plot for the experimenter. Yet we didn't make use of this feature in our study since the subject's visual feedback was sufficient to control the experiment. The experimental data is automatically saved after finishing a trial, but a session should always be terminated by a mouse click on

the button *Stop session* in order to update the subject's database.
We used the form in appendix B.1 to assess the subject's physical and mental state on every day of the experiment. It can help to survey the subject's motivation concentration and could help to explain the reason for abnormal trials.

2.2.3 Analysis of Noise and Measurement Consistency

Since the JR3 load-cell used for the experimental setup can measure a force up to 125 N and has a digital resolution of 0.013 N, the forces and their fluctuations that we measured during this experiment were considerably small compared to the nominal values. This chapter proves that the accuracy was by far good enough for our purposes and explains how the effect of noise was minimized.
Figure 2.8 shows the Fourier analysis of a recorded constant force. There were prominent noise frequency components originating from the power supply at 60 Hz and 180 Hz as well as unknown components between 20 Hz and 40 Hz. Matlab offers an assisting function to design appropriate low-pass filters. A butterworth filter of order 13 with a cut-off frequency of 30 Hz significantly reduces most noise components (figure 2.8).

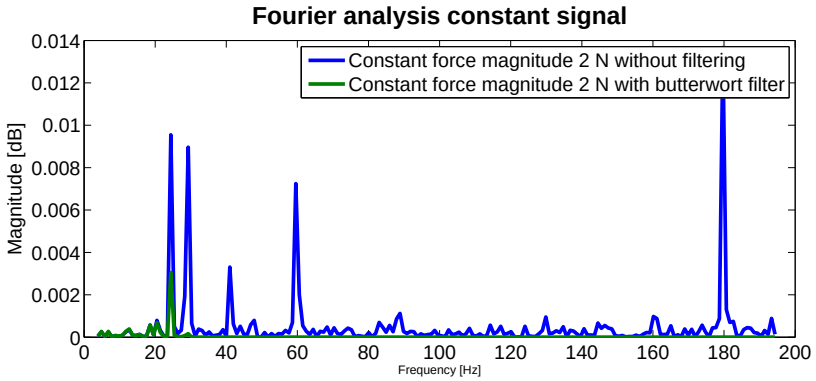

Figure 2.8: The force plate was loaded with a calibrated dead-weight of 2 N. The plots show the Fourier transform of the recorded force magnitude once unfiltered and once with applied butterworth filter. Notice that the filter successfully suppressed the noise frequencies above 60 Hz and strongly attenuated the frequencies between 30 Hz and 60 Hz.

The plot 2.9 shows the effect of the same filter, applied to an experimental trial. Again, the filter suppressed the noise frequencies, whereas the frequencies of physiological hand tremor around 8 Hz [58] were not affected. The standard deviation of the force magnitude due to noise of an unfiltered, constant signal is 0.018 N and in the range of the sensor resolution (compare figure 2.10). The butterworth filter significantly reduced the fluctuations of the measurement and increases the SNR from 101:1 to 404:1. The SD of 0.0049 N is well below the sensor resolution, which meas that the measurement accuracy is limited by the digitalization rather than by noise.

A noiseless signal was utterly important for a meaningful interpretation of the data. But it was also important for the subjects to get smooth visual feed-

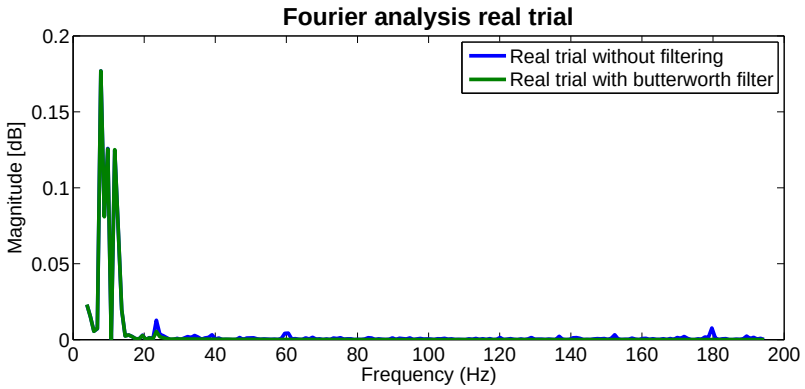

Figure 2.9: The plot shows the Fourier analysis of a real experimental trial. Notice that the filter successfully suppressed the noise components, while it left the tremor frequencies below 15 Hz unaffected.

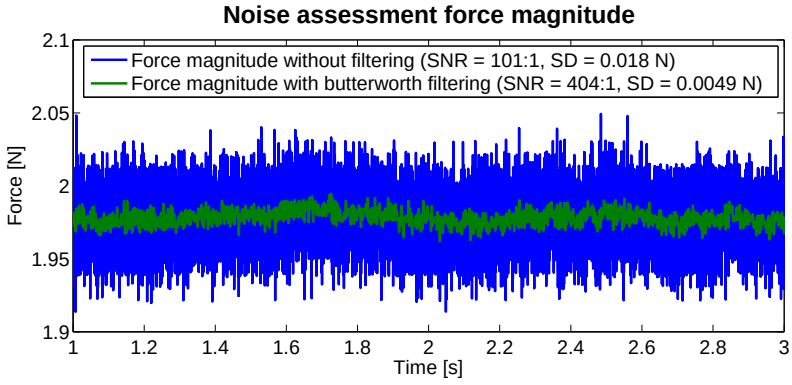

Figure 2.10: Force magnitude for a constant force with and without applied low-pass filter. Notice that the filter considerably increases the SNR and that the SD of the filtered signal is well below the sensor resolution of 0.016 N.

back that unaffectedly represented their performance. Since the visual feedback was updated less often than every 100 ms and the signal was sampled at 1000 Hz, roughly 100 data points were averaged to build up a single frame. The moving average operation already served as a low-pass filter and the effect of an additional filter would be negligible (figure 2.11). For the visual feedback as well, the SD of the signal is significantly lower than the sensor resolution. The SNR of the unfiltered frame sequence was even higher than for the real signal with applied filter, because the moving average operation filters out frequencies which belong to the natural tremor. In any case were these frequencies are only important for the actual data analysis and aren't needed for the visual feedback. The subjects were free to apply the force on the plate wherever they wanted. To make sure that the contact point didn't affect the measurements, we applied dead weights to nine uniformly distributed positions on the plate. The position of the calibration weight did not affect the measurement (figure 2.12).

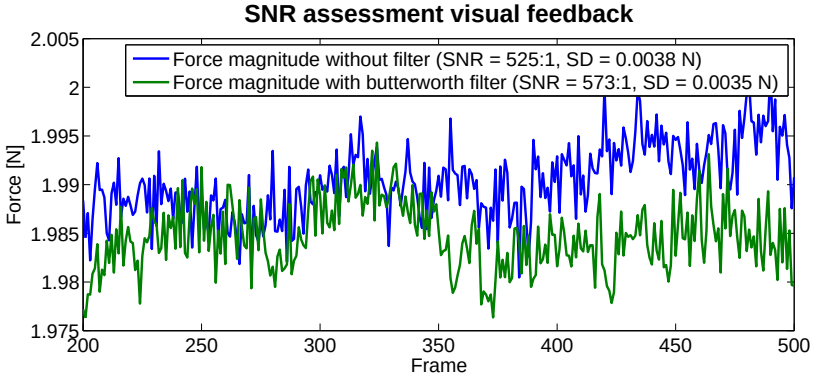

Figure 2.11: Effect of the butterworth-filter on the visual feedback. The filter hardly affected the signal, since the averaged force magnitude used by the visual feedback represented an already filtered signal.

The most salient limiting factor regarding system performance was the sensor resolution. Since we did not assess the amplitude of the actual tremor and our measurements for accuracy have values well above the resolution, the sensor was satisfactory for this experiment. However, if we plan to use the experimental protocol with PD patients to investigate a neurodegenerative disease (compare 3.4, an assessment of tremor frequency and magnitude might become neccesary.

2.2.4 Processing and Analyzing the Data

Processing

The software described in section 2.2.2 stores each trial consisting of three repetitions in a separate file. At the beginning we tried to automatically extract parameters for speed and accuracy with computer vision based methods. The most promising algorithm calculated the moving variance for a specific time window. The calculation of the variance mapped the rise phase or the rock phase to a bell-shaped peak, where the two bases represented onset and offset

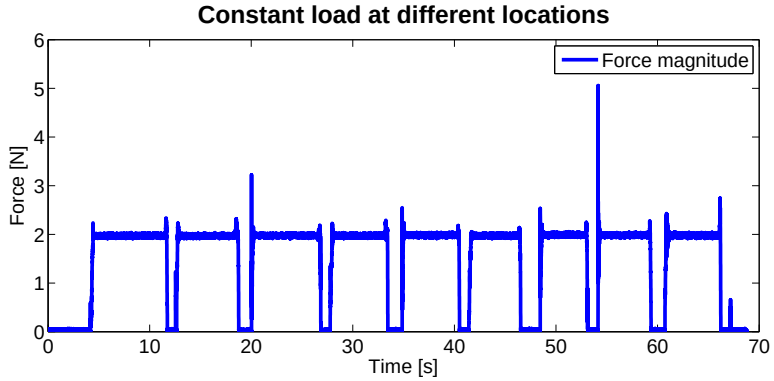

Figure 2.12: A calibrated weight of 2 N was applied to 9 different positions on the force plate. Visual inspection of the plot shows that the weight position didn't affect the measurement.

of the transition. The method worked very well for most trials, but didn't work for abnormal trials. Since it would have been very difficult to sort out these trials post-hoc, we decided to perform the data processing by manual inspection. Matlab offers functions to display data and to use a crosshair to select specific data points. For the analysis of the scaling, we were plotting the force magnitude against the time and for the re-directing the angular deflection against the time. In a first step, we splitted up one trial into the three repetitions and processed each separately. The plots to process the data are shown in figure 2.13. For each trial, the user selected the beginning and the end of the rise phase or rock phase respectively, as well as the end of the hold phase. The vertical black line splitted the screen in half and a fourth click inside the dedicated areas decided whether a trial was kept or dismissed due to abnormalities.

Based on the vertices set above, we extracted different parameters representing speed and accuracy. We looked at the time from the beginning of the transition phase until its end, but also stored the time between two the thresholds that were used for the real-time grading of the trials. For the scaling, we extracted the overshoot of the first peak, the standard deviation of the hold phase and the ratio of the standard deviation during the first and the last third of the hold phase as accuracy parameters. Similarly, the RMSE for the force during the rock phase and the angular overshoot over the target deflection were in consideration as accuracy parameters for the re-directing.

As described in chapter 1, we used Spearman's coefficient to express the correlation between speed and accuracy and we fitted the tradeoff between the parameters to a Fitts's law like distribution, which accounts for learning effects. Matlabs function NLINFIT is a powerful tool for non-linear regression. While most combinations between the proposed speed and accuracy parameters followed the behavior presented in the paper, the choice depicted in figure 2.14 produced the most articulated results. Outcomes with other parameters will be presented in section 2.3.2. As already mentioned in the paper, the normalized sum of speed and accuracy served as an indicator for overall performance.

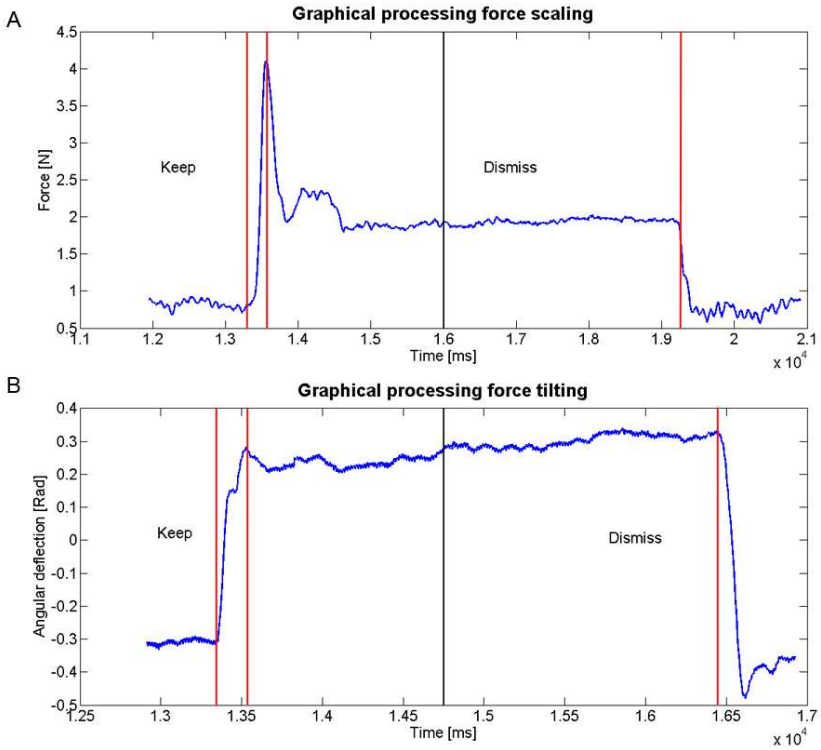

Figure 2.13: The figure shows a representative trial for force scaling (A) and re-directing (B). The first two red lines indicate the onset and offset of the transition phase and the third line denotes the end of the hold phase. A Matlab function allows to manually set these points and accordingly extract parameters of interest.

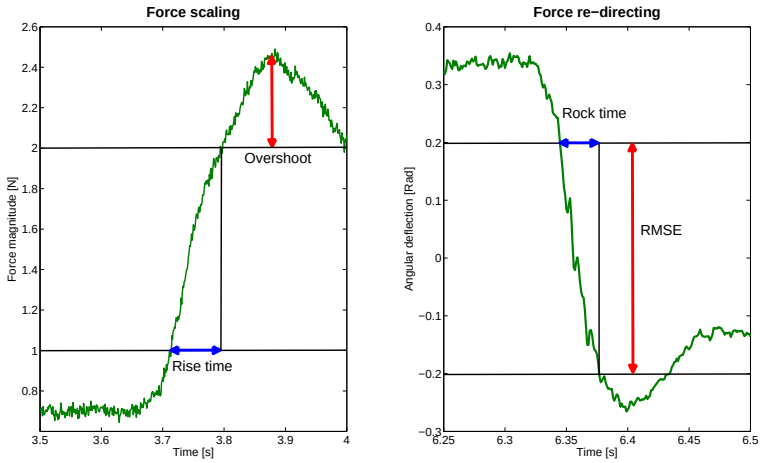

Figure 2.14: Definition of the parameters for speed and accuracy for the force scaling (left) and re-directing (right).

Statistical Analysis with SPSS

We used repeated measures analysis of variance (RM-ANOVA) to test for significant changes of the parameter values between the training sessions. Since the parameters fluctuate a lot during a single session, we calculated the means of each parameter for each session and each subject before performing the statistical analysis. Matlab offers a variety of statistical tests, however no methods for RM-ANOVA. Instead, we used the statistics package SPSS (IBM SPSS Statistics, Version 20). The most convenient way in SPSS to perform the test is defining a RM-General Linear Model (RM-GLR). The analysis through the RM-GLR includes Mauchly's test to prove sphericity assumptions and the actual ANOVA. Since we were only interested in significant differences between successive session, SPSS only checked for significant means between successive levels instead of using the Tukey-Kramer test for multiple comparisons. Finally the software performs a regression and reports linear, quadratic and cubic trends across the time series.
The RM-ANOVA was performed for accuracy and speed for both experiments. We conducted the same procedure to test for significant differences between the overall performance. The difference in Spearman's coefficient between scaling and re-directing was investigated using common ANOVA.

2.3 Additional Results

2.3.1 Detailed Statistical Results

In the paper (chapter 1), the statistical methods were explained very roughly and results of the analysis were given by reporting only the p-value. Here, we will present a more detailed analysis and further justify the significance of the presented results.
Recall that for the force scaling both, the changes in rise time and overshoot across the sessions were significant but there was no significant change in performance. The corresponding significance values of Mauchly's test were $p = 0.221$, $p = 0.162$ and $p = 0.082$. While the first two parameters didn't violate the sphericity assumption, it didn't matter in the performance, since there was no significant change at all. The pairwise comparison of successive sessions unveiled, that the only significant change is between session 2 and 3 ($p = 0.018$ for speed and $p = 0.007$ for accuracy. This could have been expected, since the overal performance didn't change but the experimental conditions were changed between these sessions. None of the pairwise comparisons between sessions regarding performance were significant.

The estimated marginal means allow to perform a regression analysis across all subjects. The trends are shown in figure 2.15. The rise time had a clear downward trend and the overshoot was increasing, especially between sessions 2 and 3. However, the indicator for overall performance only fluctuated very slightly. According to these observations, the regression analysis proved significant linear trends for speed ($p = 0.0224$) and accuracy ($p = 0.044$) but neither a linear nor a quadratic or cubic trend for performance.
The regression analysis for the re-directing produced very oppositional results (figure 2.16). Even if there is a visible downward trend in rock time, the linear regression was not significant ($p = 0.211$). The same holds true for the

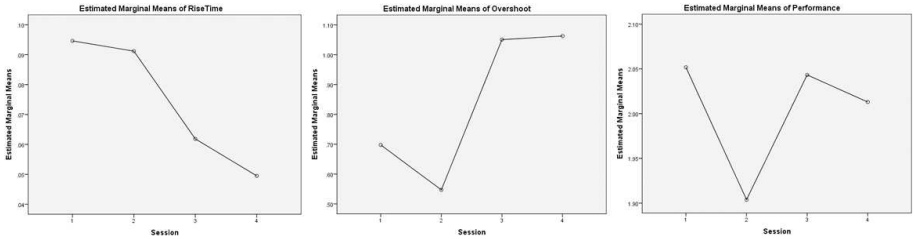

Figure 2.15: Estimated marginal means for the different measurements extracted for the force scaling. While speed and accuracy show an increasing and decreasing trend, respectively, there is no trend for the performance.

RMSE ($p = 0.911$), while there is a significant negative linear trend for the combined means (increasing performance) with a p-value of 0.041. The only statistically significant change across the four sessions was for the performance ($p = 0.013$), where Mauchly's test was not significant ($p = 0.913$) and the sphericity assumption was not violated. The successive changes in means between the session were all non-significant. In addition, the main improvement and only significant change for the performance happened between session 3 and 4 ($p = 0.021$).

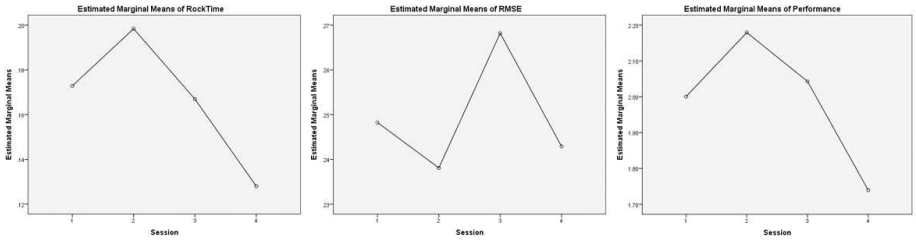

Figure 2.16: Estimated marginal means for the different measurements extracted for the force re-directing. While speed and accuracy didn't have significant trends, there is a negative trend for the normalized sum of seed and accuracy, which indicates improving performance.

2.3.2 Speed-Accuracy Tradeoffs

As mentioned in section 2.2.4, we extracted more parameters from the data than we were finally using in the paper. This section shows the same facts using different parameters for speed and accuracy. Here, we used the total rise time (rock time for re-directing) instead of the threshold-based measurements as a speed indicator. The SD ratio between the first third and the third part of the hold phase replaced the accuracy parameter for the scaling and the angular overshoot was used for the re-directing.

We could still observe the semilogarithmic tradeoff between speed and accuracy for the force scaling (2.17). The higher SD ratio between the different parts of the hold phase indicated less precision with increasing speed. Though,

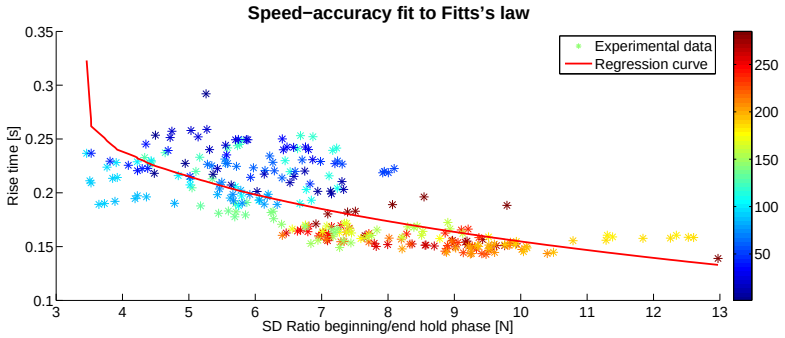

Figure 2.17: Speed-accuracy tradeoff for scaling the index fingertip force vector. The colored asterisks mark measurement data and the red curve is a fit to a semi-log function. It is obvious that the data follows the regression curve. The color codes for increasing trial number. The color gradient runs from left to right, indicating that the rock time decreased and the overshoot increased with increasing trial number.

the shape is less explicit and the $R^2 = 0.4344$ considerably lower. For the re-directing in contrast, there was still no correlation between rock time and angular overshoot. The datapoints were randomly distributed and also the training didn't lead to either higher speed or lower overshoot.

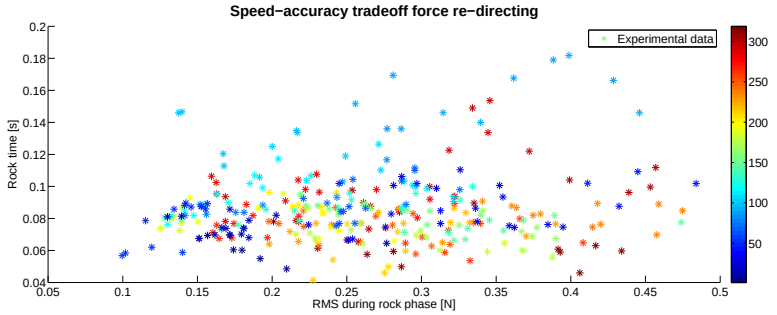

Figure 2.18: Speed-accuracy tradeoff for re-directing the index fingertip force vector and maintaining a constant force magnitude. The colored asterisks mark measurement data and the color codes for the trial number. Apparently there is no correlation between rock time and angular overshoot. Also there is no connection between trial number and speed or accuracy.

The plot of speed and accuracy against the trial number (figure 2.20) shows similar characteristics. There is no effect of the training on the angular overshoot but a slight decrease in rock time is noticeable. However, the curves are uncorrelated and the decreasing rock time didn't cause an increase in overshoot. On the other hand, there are clear trends for speed and accuracy when inspecting the same plot for the force scaling (figure 2.19). Both parameters clearly change between session 2 and 3 around trial 142. But also smaller changes in rise time cause antithetic fluctuations in the SD ratio.

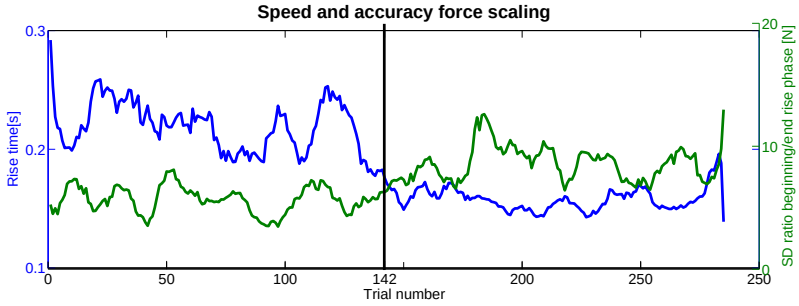

Figure 2.19: Rise time and SD ratio between first third and last part of hold phase plotted against trial number for scaling the force magnitude. The two curves fluctuate antithetically and noticeably change their values between session 2 and 3.

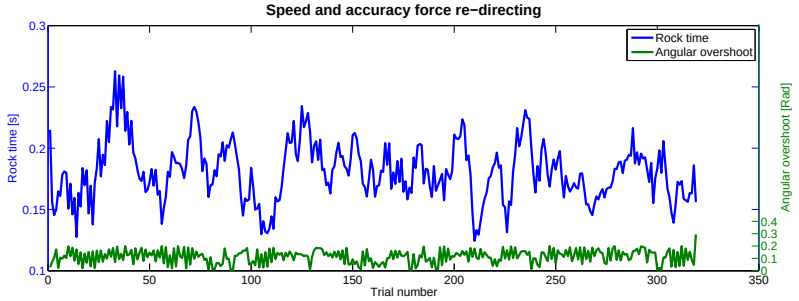

Figure 2.20: Rock time and angular overshoot plotted against trial number for re-directing a constant force vector. There is a slight negative trend in the rock time but no noticeable effect on the overshoot. Additionally, the two traces are not related.

The statistical evaluation could confirm the observations. The decrease in rise time was statistically significant ($p = 0.001$) and caused a significant increase of the SD ratio ($p = 0.002$). There was a significant linear regression for both parameters. In this case, also the change in performance was significant with a p-value of 0.036. However a pairwise comparison between the sessions revealed that there is no change between session 1 and 4, which means that the overall performance couldn't be improved.
As mentioned above, there was a slight negative trend in the rock time for the force re-directing. Since the data violated the sphericity assumption ($p = 0.05$ in Mauchly's test), we had to apply a Greenhouse-Geisser correction and the change across the sessions was non-significant ($p = 0.099$). Also the borderline-significant linear regression ($p = 0.091$) indicates a decrease in rock time, even if it's not significant under a 95% confidence interval. In any case, there was no significant variation in accuracy ($p = 0.396$). As presented in the paper however, the overall performance increased during the training ($p = 0.009$) and we could find a positive linear trend ($p = 0.016$).

2.3.3 Blind Trials

In addition to the regular trials with visual feedback, we suppressed the feedback screen for every 5th trials. Since the analysis of the blind trials didn't reveal interpretable results, we didn't include this part into the paper. These findings will be presented in this section together with a proposition about how to change the protocol for future experiments.
We analyzed the blind trials in exactly the same way as we analyzed the regular trials. Most subjects improved a lot in speed (figure 2.21 shows the plot of a representative subject). However, many trials could not be properly analyzed and had to be skipped or produced erroneous values. None of the three parameter's means (rise time, overshoot and normalized sum) changed significantly throughout the sessions.

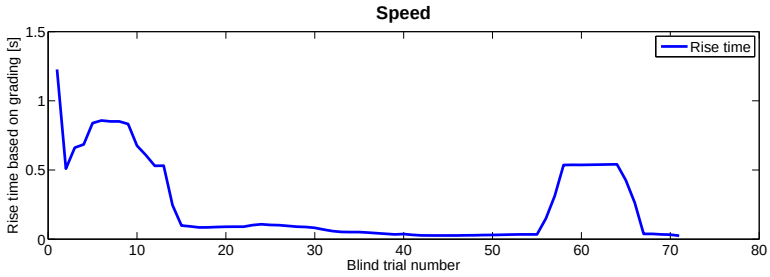

Figure 2.21: Threshold-based rise time measurement for trials without visual feedback. The rise time fell very quickly below a value that would enable the use of visual feedback during the trials, and a lot of trials were abnormal and couldn't be properly processed (e.g. trials 55 to 65).

We found similar results for the re-directing task. All three parameters didn't change significantly during the training. The performance actually had a positive trend identical to the trials with visual feedback, but the regression wasn't significant.

The blind trials showed the very important fact, that the subjects performed the tasks gradually faster (compare figure 2.22), even without visual feedback. This is evidence for the assumption that we had only feed-forward control in our experiment. However, there were too many abnormal trials to get statistically significant results. A main problem with the blind trials was that the subjects drifted away from the starting position during a trial. After the first repetition, the subjects couldn't adjust to the reference force or position anymore without visual feedback. To overcome this issue, blind trials should only consist of one repetition. The subjects also dismissed the blind trials. Supposedly, if we introduced the grading also for the blind trials, subjects would be more motivated to perform well and improve.

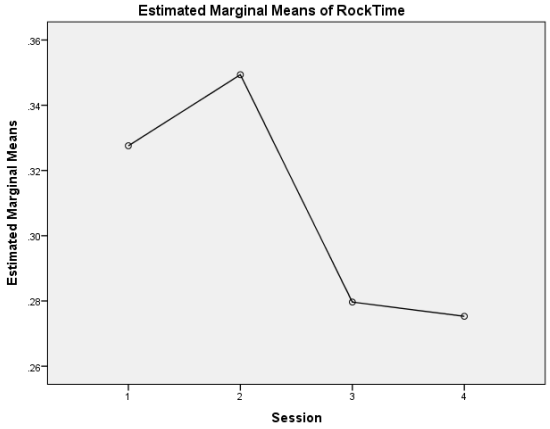

Figure 2.22: Estimated marginal means when performing an RM-ANOVA on the rock time for force re-directing without visual feedback. There is a visible negative trend, but the change is not significant ($p = 0.404$)

Everything is vague to a degree you do not realize
till you have tried to make it precise.
Bertrand Russell

Chapter 3

Using an Industrial Robot to Calibrate Load-cells

3.1 Theory

There are innumerable manufacturers of force and torque transducers, which offer a countless variety of sensors for applications ranging from biomedical devices through the automotive industry to aerospace design. These sensors measure very small loadings up to heavy-duty loads in one axis, multiple axis or include even movement and acceleration measurements [59]. While every company has their own designs for the sensors, the very basic principle of using a strain gauge to measure the deflection of a metallic component is preserved. The following section gives an overview over common designs, highlights measurement uncertainties and lists common calibration procedures and devices.

3.1.1 The Strain Gauge

Force and moment exerted on an object can be measured through the deflection, respectively the strain ϵ caused by the load. From several possibilities to measure strain, the foil-bonded metallic strain gauge is the most commonly used to produce load-cells. A major part of the information and figures presented in this section originate from an application note by National Instruments [60]. A thin metallic wire experiencing a strain will be deformed and reacts with a linear change in electrical resistance.

Figure 3.1 shows the basic design of the deflection-transducing element. The parallel thin wires are aligned with the direction of the applied strain. The zigzag structure of the wires causes a multiplicative larger strain effect than if using just one wire. The wiring of the strain gauge sits on a foil, which is bonded to the surface of a specimen experiencing strain. The most salient parameter of the strain gauge is its sensitivity to strain, namely the gauge factor GF in equation 3.1.

$$GF = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} = \frac{\frac{\Delta R}{R}}{\epsilon} \tag{3.1}$$

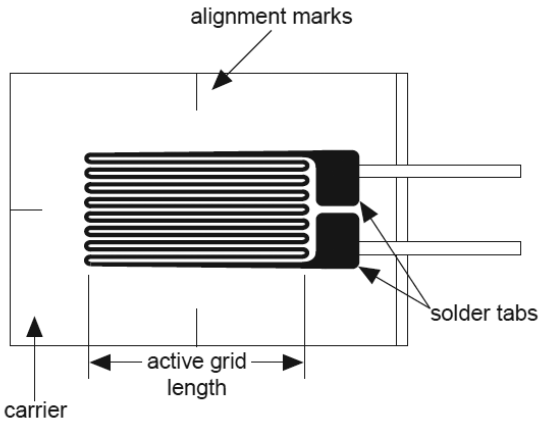

Figure 3.1: Bonded metallic strain gauge.

The gauge factor is defined as the ratio of relative change in electrical resistance and strain. Unfortunately both, gauge material and specimen material, respond to changes in temperature. Manufacturers try to minimize sensitivity to temperature changing by processing the material to compensate for the thermal expansion that the specimen material is experiencing. Compensated gauges reduce the sensitivity to temperature, but do not completely remove it.

$$V_0 = (\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2})V_{Ex} \tag{3.2}$$

A Wheatstone bridge is an electrical circuit used to measure an unknown electrical resistance by balancing two legs of a bridge circuit (Figure 3.2 A). The output voltage V_0 of the bridge follows equation 3.2 and thus for $\frac{R_1}{R_2} = \frac{R_3}{R_4}$ the output voltage is zero. Changing the value of one of the resistances results in a non zero output voltage. Therefore, if one resistance is replaced by an active strain gauge, the output voltage can be used to calculate the change in resistance. However, the function relating the output voltage with the strain for this so-called quarter-bridge is non linear. To overcome this issue, two active gauges attached in opposite directions can be included into the circuit (compare figure 3.2 B). The voltage output of this half-bridge is linear and approximately doubles the output of the quarter-bridge configuration. To further increase the sensitivity of the circuit, all four arms of the bridge can be made active strain gauges (figure 3.2 C). For the full-bridge circuit two gauges are mounted in compression and two in tension. The output voltage follows the very simple relation 3.3.

$$\frac{V_0}{V_E x} = -GF \cdot \epsilon \tag{3.3}$$

3.1.2 Uniaxial Load-Cells

A single-axis load-cell can be built surprisingly easy using a bending beam (compare figure 3.3 (B)). The right side shows a schematic drawing of the ready-

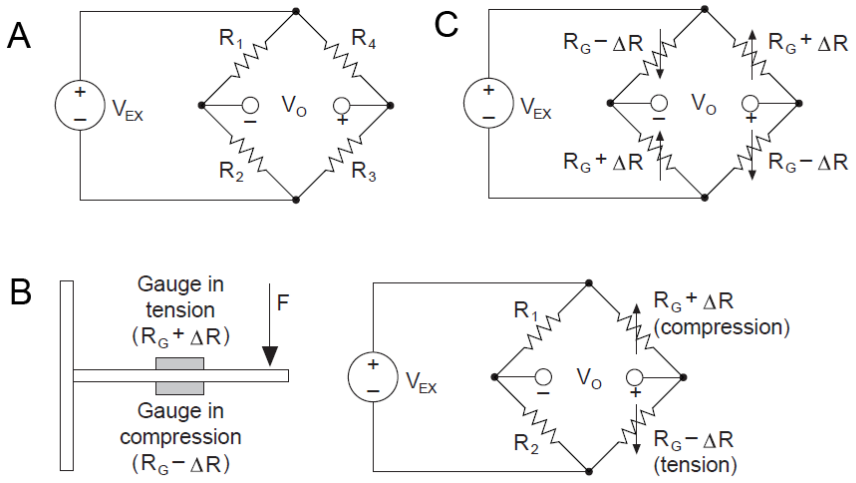

Figure 3.2: Wheatstone bridges and their use to measure strain gauges. (A) General Wheatstone bridge with four different, constant resistances. (B) Half-bridge solution with two strain gauges attached to the circuit, which results in linear strain response. (C) Full-bridge layout with four gauges to enable maximal sensitivity.

to-the-market sensor on the left. The large hole bored through the beam forms thin walls on top and bottom of the beam. This concentrates the compressive forces to a small area and causes high strains in these points. Usually two strain gauges are used for these sensors, one on top and one at the bottom of the hole.

An "S"-shaped bending beam as shown in figure 3.3 (A) is a slight improvement of the sensor described above. The load is applied straight through and the transducer is mainly designed for tension. However, also compressive loads can be measured since the principle with two strain gauges and the big through hole is kept.

3.1.3 Multi-Axis Load-Cells

While uniaxial load-cells account for 99.9% of the sensors on the market, multi-axis load-cells are the more interesting and technically more demanding versions of force transducers. Six-axis sensors measure force and torque components in all three spatial dimensions, while recent advances in sensor technologies even include angular and linear accelerations. The basic building block of these load-cells is usually a metal piece with an annular cross section, known as strain rings. While it is possible to build a six-axis sensor out of a single ring and cleverly place six strain gauges, most manufacturers use 3 to 4 rings for their sensors. Yet it is not uncommon that custom sensors with up to 16 rings are developed.
Besides a good layout for the strain rings, a second important factor to design a sensor is to reasonably choose the number of gauges and to place them intelligently.

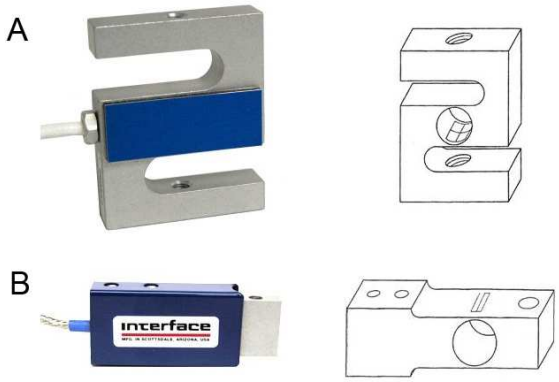

Figure 3.3: Two designs of single-axis load-cells with their schematic technical drawing. (A) S-beam, which enables to measure tension and compression on both ends of the beam. The loading axis corresponds to the vertical axis of the drawing. (B) Bending beam, which enables compression measurements in direction of the beam alignment. Figure adapted from [61].

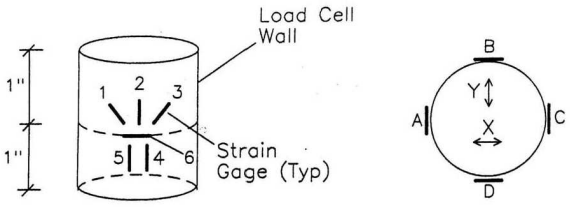

Figure 3.4: Placement of 24 strain gauges in a five-axis load-cell with two strain rings. Six gauges each are placed on four equidistant positions around the ring to measure the force in z-direction, shear in x- and y-direction and torque in x- and y-direction. Figure adapted from [62].

Figure 3.4 shows a design with 24 gauges to measure the axial force as well as shear in x- and y-direction and torque in x- and y-direction. The left side of the figure depicts the spatial orientation of the gauges. While the parts 1 (active element) and 6 (passive element to compensate for temperature, compare [60]) are used to measure the axial force, the gauges placed in 45° off axis measure shear and gauges 4 and 5 report the torques. 5 bridges, each combining 4 gauges at one of the positions indicated on the right side of the figure forward the output voltage for the 5 components. For detailed circuitry and more information compare [62].
Clever placement of the gauges will result in one distinct output voltage per axis [63]. However, usually its not possible to design the sensors that carefully and a strain in a specific direction also affects the gauges corresponding to different axes. Consequently all output voltages and their cross-talk need to be known in order to report the measurement of a single axis. This requires thorough calibration of the load-cell, which will be discussed in the next section.

3.1.4 Calibration Standards and Calibration Devices

Each manufacturer has own calibration protocols but also own, often custom-made calibration devices. An engineer from JR3 gave away unhesitatingly the information about the calibration procedure (G. Sakona, personal communication, August 10, 2012).
The principle for all their rigs is the same. They bolt the sensors in calibration to two thick and solid adapter plates, which are, on their part, bolted to the calibration rig. These plates idealize the boundary conditions in a sense that they stiffen the sensor's mounting surface and that they create a consistent path for loadings from the rig to the sensor's internal elements. The rigs allow to accurately apply single-axis loads to the sensor. They use calibrated dead weights on trays, which are suspended by small diameter cables and in some cases passed over pulleys, to allow all 6 axes of forces and moments to be loaded in both the positive and negative directions.
To collect the calibration data, they apply loads to each axis independently. One loading typically consists of 21 points, where they start at 0, increase the loading by 20% for each step until they reach full scale. Then they decrease the loading to the negative full scale in the same steps and finally increase the loading back to 0.
The total of six axis loadings, with 21 loadings each and the reading for all six axes per loading, results in 756 data values. To generate the 6x6 calibration matrix A (see equation 3.4 for its structure), they fit the data set to the linear best fit for every axis. Also, they are experimenting with new protocols that use combined loads in order to speed up the process.

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} A_{fx_1} & A_{fx_2} & A_{fx_3} & A_{fx_4} & A_{fx_5} & A_{fx_6} \\ A_{fy_1} & A_{fy_2} & A_{fy_3} & A_{fy_4} & A_{fy_5} & A_{fy_6} \\ A_{fz_1} & A_{fz_2} & A_{fz_3} & A_{fz_4} & A_{fz_5} & A_{fz_6} \\ A_{tx_1} & A_{tx_2} & A_{tx_3} & A_{tx_4} & A_{tx_5} & A_{tx_6} \\ A_{ty_1} & A_{ty_2} & A_{ty_3} & A_{ty_4} & A_{ty_5} & A_{ty_6} \\ A_{tz_1} & A_{tz_2} & A_{tz_3} & A_{tz_4} & A_{tz_5} & A_{tz_6} \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix} \tag{3.4}$$

Even though every company has its own calibration standards, the ASTM

norm E74-6 [64] regulates the calibration methods for force-measuring devices:

- Each axis of multi axis devices must be loaded independently
- At least 30 loadings per axis are required
- There should be at least 10 different loadings, while each loading should be applied twice
- They suggest an arbitrary polynomial regression, but recommend to use a second-order fit

$$\eta = 2.4\sqrt{\frac{d_1^2 + d_2^2 + \cdots + d_n^2}{n - m - 1}} \tag{3.5}$$

Also, they suggest the uncertainty (see equation 3.5) of a calibration as a measure of quality. d_i denotes the difference between actual loading and loading predicted by the calibration for the i_{th} loading, n is the number of calibration points and m the degree of the polynomial fit. The uncertainty is calculated for each axis seperatley.

Given this information, we can define the goals of this project: Since the calibration process of a load-cell is a costly and time-consuming process, we would like to build a calibration system to calibrate the sensors in the lab. In this project, we want to calibrate a cell of the type ATI Nano17 (see 3.2.2 for technical details) since it has the most complex calibration matrix. For this load-cell, we want two different calibrations: Calibrate compressive z-forces and x- and y-forces in a way, that the uncertainty is lower than the digital resolution of the sensor. Since the generation of torques is more difficult, we want a second calibration to include all axes in an as-accurate-as-possible manner.

3.2 Implementation

An Adept six-300 industrial robot with six independent axes has been used in the Brain-body dynamics lab for different purposes, such as positioning experimental setups or generating disturbing movements to investigate stable stance of chicks. The following section describes a novel application of the robot as a calibration device for six-axis load-cells. After a brief introduction to the robot, the hardware-side and the software-part of the calibration system will be described. The section closes with the exact calibration protocol and an explanation of the statistical methods used to calculate calibration matrices according to section 3.1.3.

3.2.1 Adept six-300 Industrial Robot and Its Control

The robot was released to the market in 2003, targeting assembly applications that require extremely accurate and fast movements. Figure 3.5 (A) shows the six different degrees of freedom. The robot tool tip has a horizontal reach of 677 mm, a vertical reach of 1019 mm and each axis can rotate with a speed

of at least $170°/s$ [65]. More importantly for the calibration of sensitive load-cells, the tool tip can be slowed down many times over and the precision of the movements is $\pm 0.02mm$.

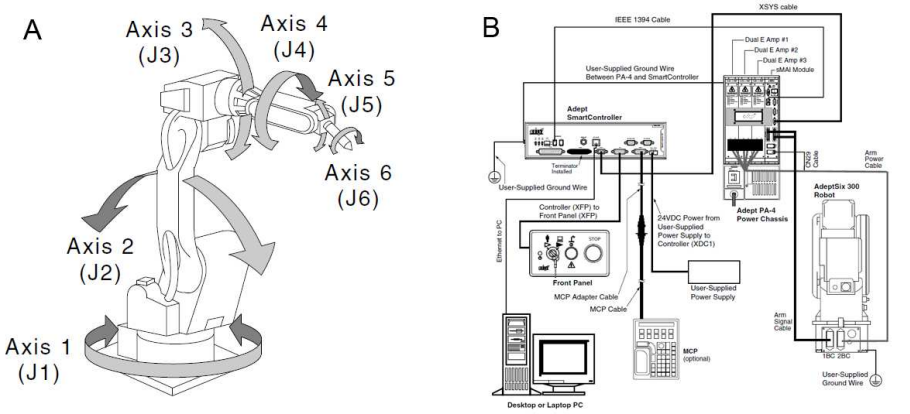

Figure 3.5: Adept six-300 industrial assembly robot. (A) Technical drawing of the robot with its six independently movable joints indicated. (B) Wiring plan of all system components. The robot has an external power supply, a manual control panel and a specialized smartController, which stores software and controls the robot in real time. Even though, programs and user commands are sent from a separate workstation.

The robot requires an external power supply (compare figure 3.5). A smart-Controller provided by Adept stores software and implements the real-time controller of the robot. The robot can be moved around the axis described before, but the tool tip can also move along and rotate around the spatial x-, y- and z-axes. Moreover, the robot can accomplish the movements either on a straight line or joint-interpolated to minimize the length of a planned trajectory. The controller provides all this functionality.
Albeit the device can be controlled through a manual panel, a user-supplied workstation usually serves to control the robot. Programs are written in an assembler-like, proprietary language V+, which enables huge flexibility in controlling the robot. The editing of software is done and commands to start and stop routines are sent from the workstation, however, software is executed on the controller.

3.2.2 Hardware Implementation of Calibration System

To recall the system requirements defined in section 3.1.4, the goal of the system is to calibrate an ATI Nano17 load-cell in a way that the calibration uncertainty for compressive z-forces as well as for positive and negative x- and y-forces is below the sensor resolution. A different calibration protocol shall implement the ASTM E 74 and calibrate forces as well as torques as accurate as possible. Table 3.1 shows the most salient design parameters of the Nano17. We needed to be particularly cautious not to exceed the torque ranges. On one hand they are considerably low (e.g. maximal axial load applied 10 mm away from the central axis) and on the other hand this is the most effective way to damage the

sensor.
A first approach tried to exploit forces caused by gravitation only and the calibration loads were theoretically calculated through mechanical and geometrical properties. As shown in figure 3.6 (A), weight balls of different size are placed inside spherical holes in the calibration cap (transparent part). The robot tip would have been attached to the dark blue plate and used to tilt the whole structure and exert forces against the sensor (between the orange and light blue plates) in all spatial directions. However, it turned out quickly that it wouldn't be possible to generate tension in z-direction (axial loading). More important, the height of the calibration cap causes a long moment arm and the allowed torques would be exceeded very quickly.

F_x,F_y	F_z	T_x,T_y	T_z	F_x,F_y	F_z	T_x,T_y	T_z
Sensing Ranges				Resolution			
25 N	35 N	250 Nmm	250 Nmm	1/80 N	1/80 N	1/16 Nmm	1/16 Nmm

Table 3.1: Measurement range and sensor resolution of ATI Nano17 SI-25-0.25 force transducer

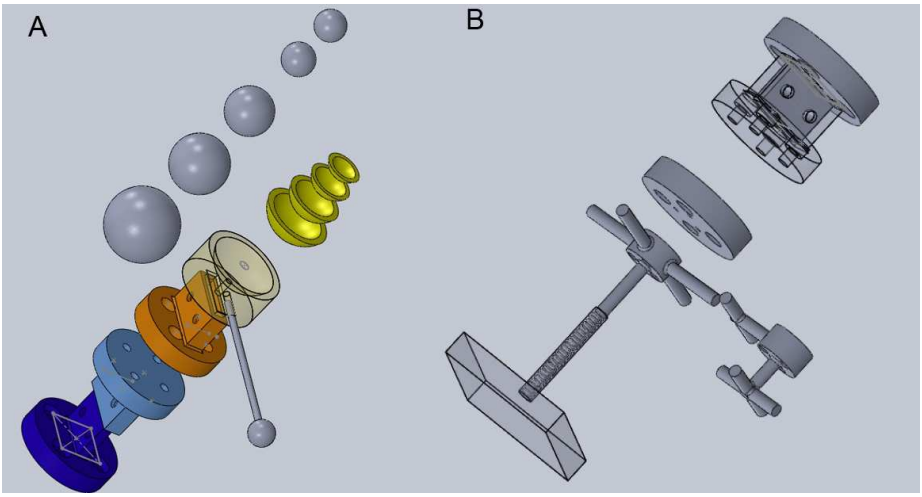

Figure 3.6: Two different approaches to calibrate a Nano17 load-cell using an industrial robot. (A) Precision weight balls of different size are placed in the spherical holes to generate different calibration forces. The whole device is tilted to generate forces in different directions. Weight gauges are used to generate torques. (B) An adapter with bars in different directions is attached to the load-cell in calibration. The bars are pressed against an appropriately chosen spring to generate forces. A trustworthy reference load-cell is used to measure the calibration loads.

Figure 3.6 (B) shows the redesign of the calibration device. Instead of just relying on theoretical values, the calibration loads are measured with a trustworthy and robust load-cell (model 20E12A-I25; JR3, Woodland, CA), attached between the small round and the transparent plates. The Nano17 is placed on the opposite side of the round plate. Different calibration caps can be attached to the sensor in calibration. A plate fixed to ground with a hollow tube holds

a compression spring. The robot is now used to compress the spring with the different bars and to generate specific forces and torques. An appropriate spring (LCM060B 13 M, Lee Springs) with a compression rate of 0.97 N/mm and a maximal load of 24 N was chosen, so that a single spring covers the whole measurement range of the sensor. Figure 3.7 shows the three different calibration caps that were used. Part A produces forces in uniaxial direction. The bars in part B to exert forces in x- and y-direction are pivoted around 45° to exert forces with both, x- and y-components, while the axial bar still exerts a compressive force in z-direction. The shorter, angled bars of part C are meant to specifically create uniaxial torques in all three dimensions. All parts were manufactured using an FDM 3-D printer by the company printo3D, Tunkhannock.

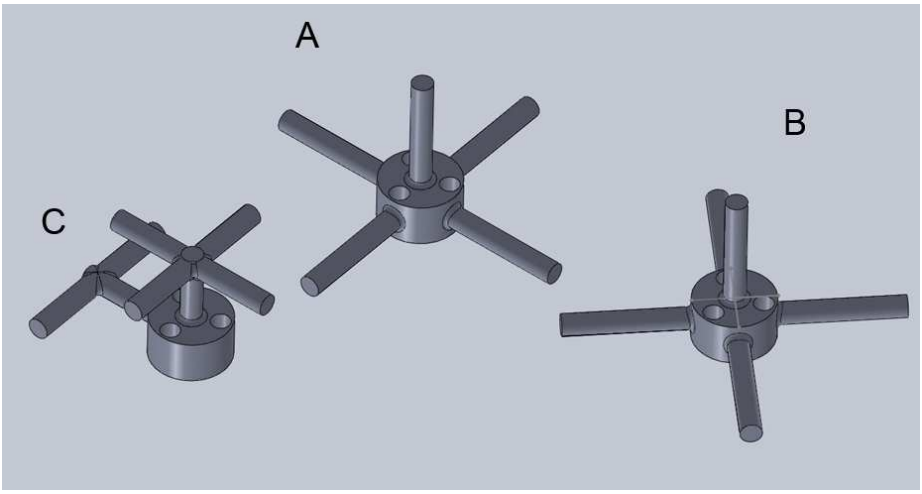

Figure 3.7: Calibration caps for the Nano17 calibration system. (A) Cap, which generates axial forces. (B) Cap generating forces, which are 45° off axis. (C) Angled bars to produce torques in axial direction.

Both sensors, the reference load-cell and the one in calibration, provide 6 output voltages. These are acquired using a built-in data acquisition card (PCI-6225, National Instruments Corporation, Austin, Texas) and transferred to a computer.

3.2.3 Calibration Protocol

We have used two different but very similar loading protocols to generate calibration data, whose implementation will be described in this section. The first protocol sticks very accurately to the ASTM norm [64] and implements uniaxial loading in all directions for force and torque. The second protocol passes on the torque calibration and includes off-axis force loading instead.
For every axis, the positive and negative directions are loaded seperately. The procedure for each partial axis is always the same: In order to condition the load-cell in a specific direction, the maximum load is applied twice before acquiring the actual data. Thereafter, ten different loadings are applied and the corresponding data is measured. The applied force or torque respectively, is se-

quentially increased by one tenth of the maximum loading and randomly modulated with an error of maximal 5%. Then the sensor is unloaded and another 10 data points are aquired in the same manner. The following two lists describe the sequence of loading the different axes.

Protocol 1

1. Negative z-force
2. Positive y-force
3. Negative y-force
4. Positive x-force
5. Negative x-force
6. Positive x-torque
7. Negative x-torque
8. Positive y-torque
9. Negative y-torque
10. Positive z-torque
11. Negative z-torque

Protocol 2

1. Negative z-force
2. Positive y-force
3. Negative y-force
4. Positive x-force
5. Negative x-force
6. Positive x-force and positive y-force
7. Negative x-force and negative y-force
8. Positive x-force and negative y-force
9. Negative x-force and positive y-force

3.2.4 Software Structure

As depicted in section 3.2.1, software for the robot is written in a proprietary language and the commands are sent through a special program. On the other hand the best way to read out and process data from a data acquisition card is Matlab with its toolboxes. Therefore we had to find a solution to communicate between the two programs. Figure 3.8 explains the whole calibration procedure and the collaboration of sensors, Matlab and robot controller.

The communication between the robot's smartController and the user workstation happens via Ethernet. Luckily not only the proprietary software, but also any other program can send TCP-packets through the Ethernet connection to the robot controller. Therefore we permanently run a tcp-server on the robot controller during the whole calibration process. The TCP-server accepts number codes, which represent specific movement sequences. Whenever the server recognizes a code, it executes the corresponding sequence, which can be either preloading an axis, performing a single loading step, unloading an axis or repositioning the whole calibration hardware to load a different axis.
However, the core of the software is implemented in Matlab. A frugal graphical user interface simplifies the use of the system. Once the calibration process is started, Matlab sequentially loops through the loading protocol described in the previous section 3.2.3. A TCP-connection is established with the server on the controller and either a command to load the sensor or to switch to a new axis is sent. Each loading step is followed by measuring the raw-voltage data from

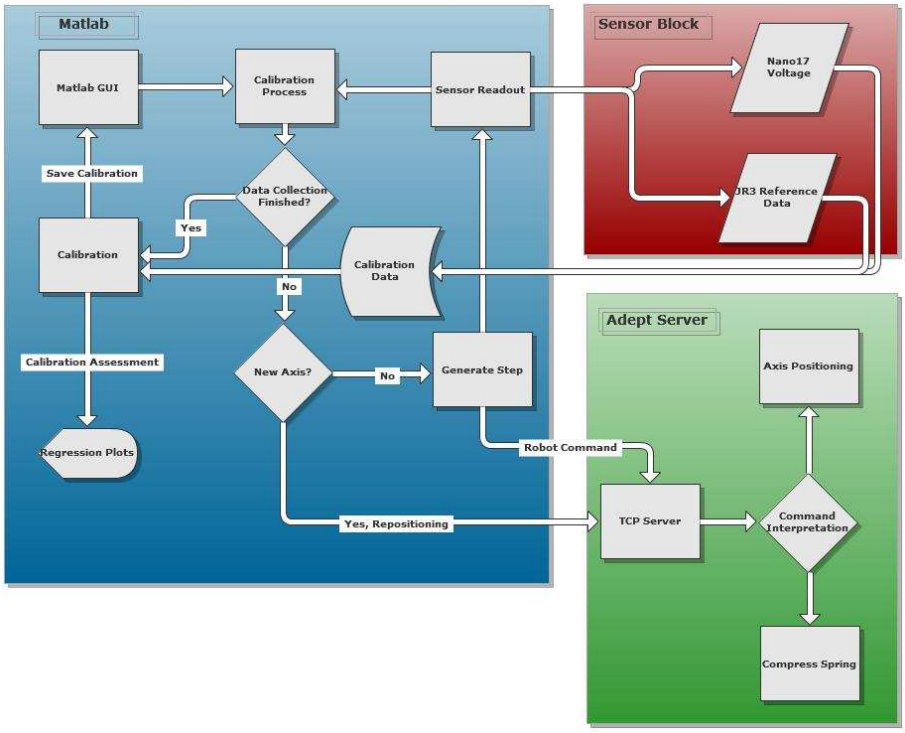

Figure 3.8: The diagram explains the workflow and the interaction of the different software components for the calibration system. The adept controller runs a TCP-server and accepts different commands to reposition the tool tip. A GUI in Matlab controls the whole calibration process, acquires data from the sensors and sends positioning commands to the TCP-server.

the Nano17 and storing the wrench vector from the reference cell. Once the protocol has terminated, the TCP-connection is closed and the acquired data will be statistically analyzed.
An important factor for the proper operation of the system is the correct timing. Since Matlab and the robot are independent components, they need to be thoroughly synchronized. The Matlab program execution has to be suspended until the robot has terminated a movement. We achieved this by implementing a custom acknowledgment protocol. Whenever Matlab sends a movement command to the robot, the software enters a waiting loop. After the successful termination of a movement, the robot sends an acknowledgment signal to the Matlab software, which enables to exit the waiting loop.
The statistical analysis of the calibration data provides different plots to evaluate the quality of the calibration. The user can decide whether he wants to accept the calibration matrix and store online, in order to be accessible for other users.

3.2.5 Methods for Calibration and Evaluation

The output of the loading process are six voltages each from the reference cell and the sensor in calibration for all loading steps. Since the properties of the bolting between the components change whenever the robot repositions the system, a record of the baseline voltages for all axis positions is taken without the attached calibration cap before the actual loading process is started. This makes sure that the forces measured by the JR3 load-cell equals exactly what the Nano17 is supposed to measure. Because the lever is different for the load-cells when measuring torques, the calculated torques measured by the JR3 have to be transformed to the torques exerted on the Nano17.

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ T_x \\ T_y \\ T_z \end{bmatrix} = \left(A_{JR3} \cdot \begin{bmatrix} V_1 - V_{O_1} \\ V_2 - V_{O_2} \\ V_3 - V_{O_3} \\ V_4 - V_{O_4} \\ V_5 - V_{O_5} \\ V_6 - V_{O_6} \end{bmatrix} \right) \cdot \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & l_x & 0 \\ & 0 & 0 & l_y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

Equation 3.6 explains how the output voltages of the reference sensor are transformed to the forces and torques being exerted on the Nano17. Here, V_i are the reference voltages, V_{O_i} are the offset voltages measured for a specific axis and l_x, l_y are the lever ratios between JR3 and Nano17. The offset is subtracted from the output voltages and the result is multiplied with the calibration matrix of the JR3. This wrench vector measured by the JR3 is finally multiplied with the lever transform matrix to conclude on the Nano17's wrench vector. For convenient cable routing, the x-axis of the JR3 is aligned with the y-axis of the Nano17. The matrix also compensates for that and exchanges the vector components.
We were using and comparing three different methods to calculate the calibration matrix out of the measured data. According to section 3.1.4 we were using Matlab's built-in functions for linear and quadratic regression to regress the output voltages from the Nano17 to the measured force data for every axis separately. Additionally, we were using Partial Least Squares regression to perform

a simultaneous linear regression of all six axis. For the latter case we were using all collected force data instead of only the data for a specific axis (e.g. loading the sensor with a force in x-direction also produces a torque in y-direction). PLS regression searches for a set of components that performs a simultaneous decomposition of force data and voltages with the constraint that these components explain as much as possible of the covariance between the two. It is followed by a regression step, where the decomposition of the voltages is used to predict force and torques [66].
To assess the quality of a calibration, we report the R^2 values, RMSE of the regression and the uncertainty. Moreover the software provides several plots that support the assessment. Plotting the residuals against the applied load reveals information about the applicability of a model. Plotting the time series of the residuals gives information about hysteresis effects. Both plots shouldn't have a structure for an appropriate mode and the residuals should be randomly distributed. Accordingly a histogram of the residuals should be bell-shaped. Another method to test independence of succeeding measurements is to plot the residuals against the residuals of the preceding measurement. Again, randomly distributed data points indicate a good fit.

3.3 Results

The calibration process could be successfully finished for both protocols and we obtained reasonable and useful results. However, the calibration needed some additional assistance apart from attaching and changing the calibration caps. The rapid prototyping material of the caps was not stiff enough and the forces caused them to bend. This had no influence on the calibration but thorough observation of the axis repositioning was necessary to ensure a smooth process. Both, force and torque axes, were loaded close to their full range. Figure 3.9 shows a plot of the six axes loadings as well as all measured gauge voltages (bottom plot). Only the z-torque did not reach its full range (compare the sixth plot). The bottom plot shows the gauge voltages of the sensor in calibration. While the reference load-cell has six gauges, that almost uniquely respond to one axis, the Nano17 obviously has a much more complex design. There are at least two gauges contributing to the output for any arbitrary loading situation.

Image 3.10 shows the evaluating plots for calibrating the x-axis. Obviously, there is no recognizable pattern when plotting the residuals against the loading step or the predicted load, which means that the error was random and not correlated with either the sequence of loading or the magnitude of the applied force. The bell-shaped histogram and the near-liner normal probability plot are evidence for a random residual distribution. The chaotic lag plot proves that successive measurements were not dependent on each other. Finally, plotting the actual force measurements against the predicted values does not deviate from a straight line. The community of these plots indicates a successful calibration of the x-force axis.
The residual plots for the two other force coordinates look similar and satisfactory as well. Also the calibration of the x-component and the y-component of the torque was successful. The residuals were random and uncorrelated and bounded to roughly 0.3 Nmm. It is noteworthy that we didn't calibrate the sensor for forces in positive z-direction. However, the calibration of the nega-

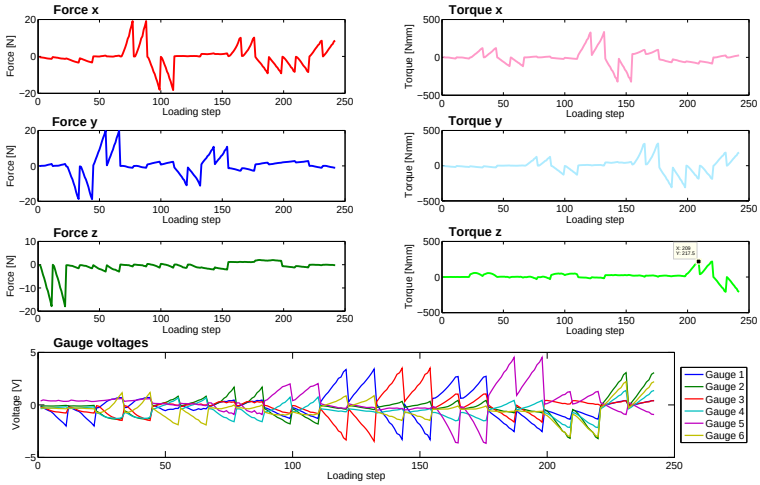

Figure 3.9: The figure shows the loading of the different axes during the calibration process. The displayed data are the loads measured by the reference load-cell, transformed to the point of view of the sensor in calibration. Notice that the three force and torque axes are loaded considerably independently, but that a force is always connected to a torque and vice versa. The bottom plot shows the recorded gauge voltages from the cell in calibration. Noteworthy is the complex design of the sensor, since always at least two gauges have non zero values.

tive direction is very well extrapolateable. We were verifying the calibration by applying three loadings of 100 g, 500 g and 1 kg to that axis. The average difference between reference value and calculated weight was 0.67%. The analytical plots for the z-torque are given in figure 3.11.

The errors were still randomly distributed, meaning that there was no connection between loading point (timewise and according to value) and error. However, the residuals were about ten times larger than for the other torque axis and the deviation from the straight line in plot six indicates that either a linear calibration is not appropriate or that the axis couldn't be calibrated properly.

The most salient plots for the x-force data look very similar when comparing a linear regression with the proposed quadratic regression (see figure 3.12). The more complex method approximately halves the error. Table 3.2 compares the different uncertainties and sets them in contrast with the sensor resolution. The linear regression performs well, the uncertainty is somewhat larger than the resolution of the reference cell for forces and even below it for x- and y-torques. As already stated before, the calibration of the torque's z-component was not satisfactory. The quadratic regression as proposed in the norm could decrease the uncertainty to a value below the sensor resolution and matches the design criteria. Row four in the table shows the uncertainty for using partial least squares regression with **all** measured data points for all axes. While the values for the force still lie in a reasonable range, the algorithm is completely inappropriate to calibrate torques.

Interestingly, the uncertainty increased and the quality of the calibration

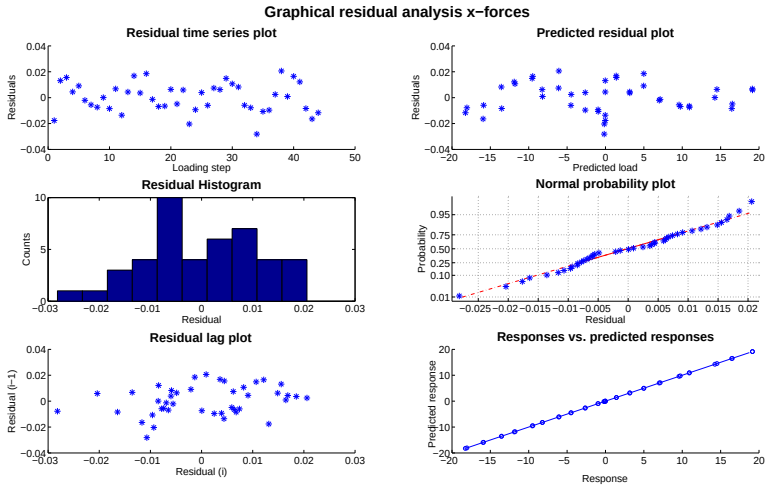

Figure 3.10: Graphical residual analysis when calibrating the x-force axis of a Nano17 load-cell. The community of these plots is an example for a satisfactory calibration.

Method	F_x [N]	F_y [N]	F_z [N]	T_x [Nmm]	T_y [Nmm]	T_z [Nmm]
Linear	0.0276	0.0378	0.0470	0.334	0.317	4.015
Quadratic	0.0122	0.0084	0	0.138	0.151	0.868
PLS	0.0779	0.0862	0.135	24.94	31.46	2.26
Force only	0.0597	0.0866	0.0468			
Resolution JR3	0.013	0.013	0.025	1	1	1

Table 3.2: Calculated uncertainties for different calibration methods in comparison with the sensor resolution.

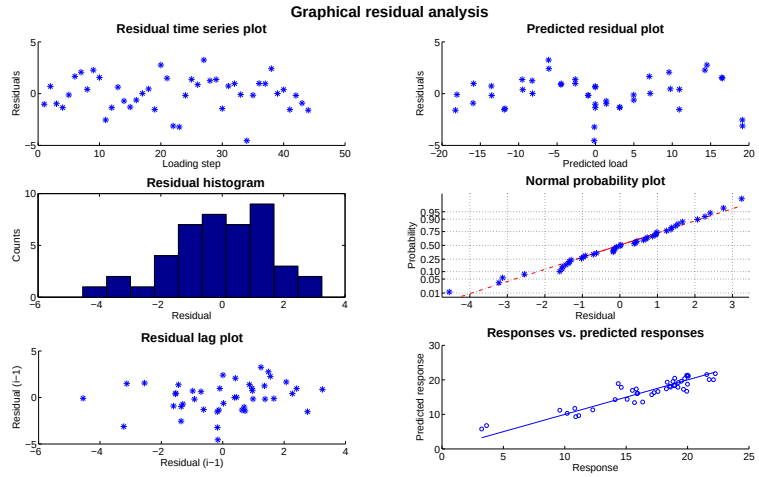

Figure 3.11: Graphical residual analysis when calibrating the z-torque axis of a Nano17 load-cell. Notice the big residuals and the significant deviation from the straight line in plot six.

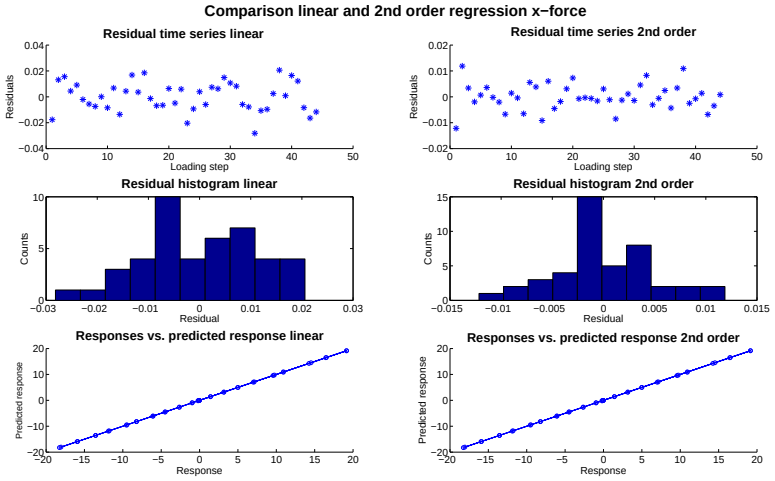

Figure 3.12: Comparing the linear and quadratic regression of the measured force data in x-direction. The shape of the plots does not change at all, however, the second-dorder fit can decrease the calibration error.

decreased when we were using combined loads for the force calibration. The results are still acceptable, but did not match the design criteria. A glimpse at the residual plots 3.13 reveals the same. While the plot loadings against predicted loadings forms a straight line, there is structure inside the time series plots. There are different linear and quadratic dependencies visible.

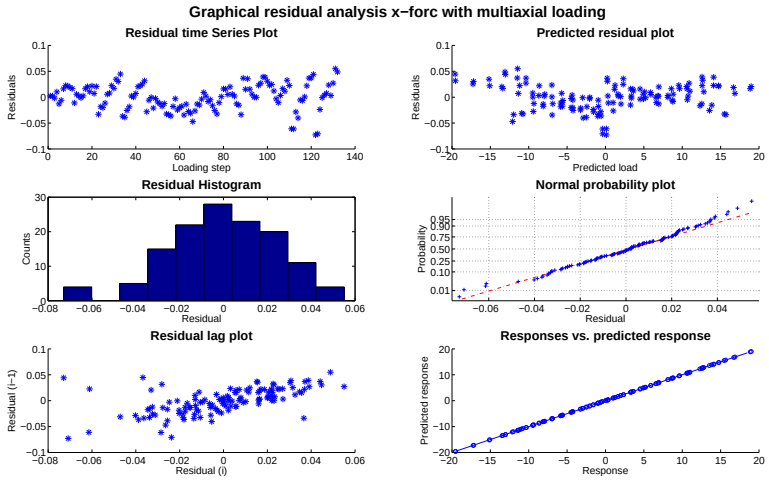

Figure 3.13: Graphical residual analysis for calibrating the force x-axis with multiaxial loadings. The protocol introduces a time dependence of the error (top left subplot).

3.3.1 Discussion and Conclusion

The best performing calibration method was using all axes to calibrate the sensor and applying a second-order regression to the data. In this way we fulfilled the design criteria for all axes except the axial torque. Obviously the resolution of the JR3 limited the performance. While the force resolutions of the two cells are comparable, the Nano17 has a much higher resolution for torques. A drawback of the quadratic regression is the complex calibration matrix. Though the linear regression is not as effective and did not completely match the criteria, we got very satisfactory results. The uncertainty is in the or of magnitude of the sensor resolution and is by far low enough for our measurements with the load-cell. Also, verification measurements with dead weights have proven practicability of the calibration.
Interestingly, the calibration quality decreased when we included combined loads in the protocol. Different factors such as temperature or hysteresis effects contribute to the non linearity of a load-cell. Creep is introduced as another very prominent factor when the sensor is loaded over a longer time span. While according to the all-axes protocol continuously different axes were loaded, the loads on the x- and y-axes were maintained during 15 minutes for the force-only protocol. Creep effects start playing a role for sustained forces after 5 minutes and cause an error of about $\pm 0.02\%$ after 15 minutes [67]. The calibration is

slightly worse with combined loads than for single-axis loading, but the procedure introduces new effects and non linearities into the data. Hence, arguably, this is the better calibration method in a sense that the result is more robust against errors.
The terribly bad calibration of torques, when using PLS regression, is clue on a big uncertainty of the system. The transformation of torques from the reference load-cell to the Nano17 is solely based on the theoretical dimensions of the system. However, bolting and imprecise manufacturing of the rapid prototyping parts caused deviations from the calculated values. Since the PLS-regression includes the data from all measurements and since all the force loadings also produce small torques as a side effect, the mentioned imprecision could have caused the high uncertainty values for torques.
Another drawback of the rapid prototyping material is the low stiffness of the calibration cap. When applying forces or especially torques, the bars started to bend and twist. This did not cause a loss in accuracy for calibrating most axes, since only the cap was deformable and the rest of the system rested considerably stiff. However, when applying an axial torque, not only the cap twisted but also the Nano17 did due to its mechanical construction. Accordingly the torque was not transmitted completely to the reference load-cell, which caused the unsatisfactory calibration of the z-torque.
Conclusively, the calibration system fulfilled the requirements and is applicable to calibrate the Nano17 for our needs. Especially since we only measure forces, the setup produced very good results. For future work, the reliability of the reference load-cell will have to be proven. Experience showed that the JR3 is a very precise load-cell, however the experiments should be repeated with another load-cell of the same type to show reproducibility. Also, to overcome the issue with the low torque resolution, we might use another Nano17 as reference for future setups. In any case, we should rethink the production process of the system parts. We chose 3-D printing because the manufacturing is cheap and quick. But for the next version we should consider stiffer materials and more precise production methods.

3.4 Manual

This manual serves as a short step-by-step introduction how to use the robot to calibrate load-cells. However, to explain the operation of the robot would be beyond the scope of this report. It is expected that the reader knows how to control the Adept six-300.

1. Attach the JR3 to the robot tool flange. Firstly bolt the load-cell in calibration to the attachment plate and then bolt this plate from the other side to the JR3. Make sure that the cables are routed in the same direction.

2. Start the robot and the corresponding software as well as Matlab and navigate to the folder which contains the calibration software.

3. Type *cd Richard* in the robot console to switch to the correct directory, type *load calib* to load the necessary methods and finally type *execute*

tcpserver to start the communication between Matlab and robot. The method tcpserver moves the robot to its initial position, hence make sure that it can move unobstructedly.

4. Type *LUCA* in Matlab to start the control GUI (see figure 3.14). The first three dropdown menus let you choose the load-cell in calibration, the reference load-cell and the data acquisition device. Even if the options are quite flexible, the reference load-cell should be the JR3 with serial number 2563 and the DAQ device is usually Dev1. You can choose between uniaxial calibration for all six axes and combined load calibration only for the force axes. The next field specifies the input channels for the data acquisition. The first six channels represent the sensor in calibration, the second six channels the reference load-cell.

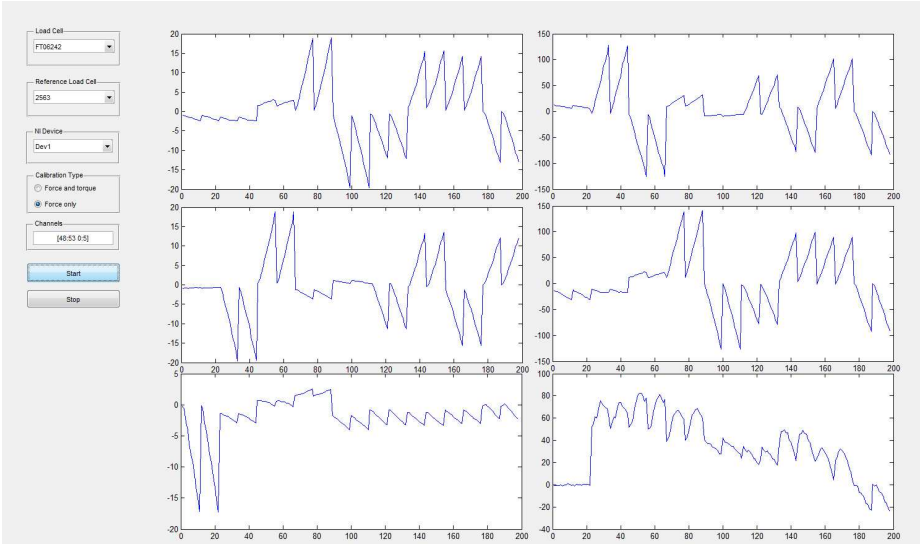

Figure 3.14: Matlab GUI to control the robot calibration process.

5. A click on the button *Start* will initiate the calibration process. At the beginning the software will take a baseline scan in different robot positions. Make sure that the robot can move unobstructedly.

6. After completion of the baseline scans, you will be asked to attach the calibration cap. Screw the force calibration cap (see figure 3.7 and picture 3.15) to the Nano17. Attach the ground plate with inserted spring to a tripod (e.g. using double-sided tape) and position the tripod in a way, that the vertical bar of the cap slightly touches the spring. As soon as everything is properly set up, press *Return* in Matlab's command line to start the actual calibration.

7. Even though the calibration and the reposition of the axes is automated, the process can dislocate the tripod. It is recommendable to monitor the correct axes repositioning. Wait until the loading of the uniaxial force axes has finished.

8. You will be asked to detach the force calibration cap and press return. The software will perform a different set of baseline scans, where the robot should be able to move freely.

9. Attach either the torque calibration cap or the combined force calibration cap and reposition the tripod. Press *Return* to initiate the second part of the calibration process.

10. After completing the second set of loadings, the robot will stop its movements and Matlab displays a set of calibration results and plots (compare section 3.2.5). The obtained calibration matrix is automatically stored in the repository as described in section 2.2.2.

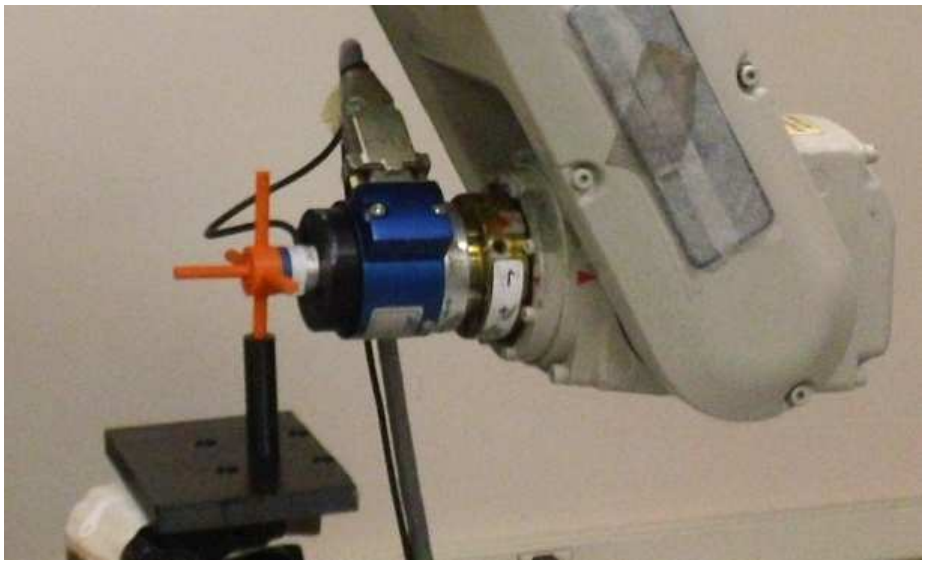

Figure 3.15: Calibration system during calibration of a force axis. The bar of the orange calibration cap interlocks with the tube of the ground plate. The cables of both load-cells, Nano17 between the calibration cap and the black adapter plate and the blue JR3 are routed in parallel.

Epilogue

This thesis is a requirement to obtain the title of MSc in Biomedical Engineering from ETH Zürich. The term clearly points out the broad range of areas and skills covered by this title at the multiple intersections of biology, medicine and engineering.
In our study, which was presented in chapter 1, we could successfully point out the complex neural circuitry involved in performing simple manipulation tasks with the fingers. We showed that humans generally are very skilled in scaling a fingertip force. The clear tradeoff between speed and accuracy and the inability to improve the performance is evidence for the maturity of the system. When the subjects tried to re-direct the fingertip force, complex circuitry, supposedly in the spinal cord, corrupted the controller and blurred the tradeoff between speed and accuracy. In return, the subjects could train the corresponding circuits and their performance increased with practice. This study proved that the control of finger musculature, although apparently versatile, operates on the verge of failure. Thus our manipulation owes as much to adequate, but incomplete and non-robust-neuromechanical evolutionary adaptations, as to socio-biological co-evolutionary design of objects and tools to make them forgiving when manipulated.
The study has not only biological impact. The analysis of neurodegenerative diseases, such as Huntington's disease, Alzheimer's disease and, most important, Parkinson's disease (PD), has a very high significance in todays medical research. There is still a lack of preindicators for PD. Since all subjects performed similarly well when scaling forces, abnormal results could indicate an early stadium of PD.
We developed with very simple and rudimentary components a calibration system for load-cells, which nearly meets industry standards and is accurate enough for the requirements in our lab. We used approaches that combine basic principles of software engineering (communication and control of the robot), electrical engineering (acquisition of the sensor measurements) and mechanical engineering (design of calibration hardware) to meet these requirements.
A major skill of an engineer is the ability to analyze a problem and to disentangle its different aspects. During my master's project, I could work on different sub-projects in different fields of biomedical engineering independently. However, the significance of these subjects is highly entangled and my work in a holistic view covered a broad range of my specialization.

Appendix A

Code Snippets

A.1 Load-Cell Interface

```
classdef loadCell < handle
    % This class provides access to the data of different load cells
    % (either raw voltage or claibrated force and torque) over a
    % session-based 64-bit interface to NI data acquirement classes
    properties
        cellName = '5761'; %laod cell type
        samplingRate = 400; %sampling rate in Hz
        samplingTime = 1; %sampling time for foreground measurements
        calibrate = true; %set true if force/torque, otherwise ...
            raw voltage data
        calibrationMatrix = eye(6);
        daqObject;
        baselineScan = zeros(1,6);
        backgroundData;
        backgroundTime;
        lh;
    end

    methods
        %constructor
        function c = loadCell(name,calibrate)
            c.cellName = name; %set name of load cell
            c.daqObject = daq.createSession('ni'); %create daq ...
                session
            c.calibrate = calibrate;
            %add six input channels (force/torque)
            for i=0:5
                c.daqObject.addAnalogInputChannel('Dev1', i, ...
                    'Voltage');
            end
            %if a calibration matrix is provided load it and take
            %baseline reference data
            if calibrate == true
                mat = load(strcat(c.cellName, '.mat'));
                c.calibrationMatrix = ...
                    eval(strcat('mat.','calibrationMatrix',name));
                [baselineData,time] = c.daqObject.startForeground;
                c.baselineScan = mean(baselineData);
            end
```

```
36          end
37          %set the sampling rate
38          function setSamplingRate(obj, rate)
39              obj.samplingRate = rate;
40          end
41          %set the sampling time
42          function setSamplingTime(obj, time)
43              obj.samplingTime = time;
44          end
45          %get a single scan on all channels
46          function data = getSingleScan(obj)
47              data = obj.calibrationMatrix*
48              ...(obj.daqObject.inputSingleScan()-obj.baselineScan)';
49          end
50          %get a foreground scan with specified length and smaple rate
51          %provide optional plot or data-handling functions
52          function [data, time] = getForegroundScan(obj,varargin)
53              %get raw data
54              [data_raw,time_raw] = obj.daqObject.startForeground();
55              %calculate effective measurement values
56              data = obj.calibrationMatrix*
57              ...(data_raw'-repmat(obj.baselineScan',1,size(data_raw,1)));
58              time = time_raw';
59              if(length(varargin) == 1)
60                      plotfunction = varargin{1};
61                      plotfunction(time,data);
62              end
63              if(length(varargin) == 2)
64                  %if a data handling function is provided
65                      datahandler = varargin(2);
66                      datahandler(time,data);
67              end
68          end
69          function startBackgroundScan(obj, updateInterval, varargin)
70              obj.daqObject.NotifyWhenDataAvailableExceeds = ...
                    obj.samplingRate*updateInterval;
71              obj.daqObject.IsContinuous = true;
72              if(length(varargin) == 1)
73                  obj.lh = ...
                        obj.daqObject.addlistener('DataAvailable', ...
                        @(src, ...
                        event)standardBackgroundDataHandler(obj, ...
                        src, event,varargin{1}));
74              elseif(length(varargin) == 2)
75                  obj.lh = ...
                        obj.daqObject.addlistener('DataAvailable', ...
                        @(src, ...
                        event)standardBackgroundDataHandler(obj, ...
                        src, event,varargin{1},varargin{2}));
76              else
77                  obj.lh = ...
                        obj.daqObject.addlistener('DataAvailable', ...
                        @(src, ...
                        event)standardBackgroundDataHandler(obj, ...
                        src, event));
78              end
79              obj.backgroundData = [];
80              obj.backgroundTime = [];
81              obj.daqObject.startBackground();
82          end
83          function [data, time] = getBackgroundData(obj)
84              data = obj.backgroundData;
```

```
85              time = obj.backgroundTime;
86          end
87          function [data,time] = stopBackgroundScan(obj)
88              [data,time] = getBackgroundData(obj);
89              obj.daqObject.stop();
90              delete(obj.lh);
91              obj.daqObject.IsContinuous = false;
92          end
93          function standardBackgroundDataHandler(obj, src, event, ...
                varargin)
94              %get raw data
95              data_raw = event.Data;
96              time_raw = event.TimeStamps;
97              %calculate effective measurement values
98              obj.backgroundData = ...
                    [obj.backgroundData,obj.calibrationMatrix*
99              ...(data_raw'-repmat(obj.baselineScan',1,size(data_raw,1)))];
100             obj.backgroundTime = [obj.backgroundTime,time_raw'];
101             if(length(varargin) == 1)
102                     plotfunction = varargin{1};
103                     plotfunction(obj.backgroundTime,
104                     ...obj.backgroundData);
105             end
106             if(length(varargin) == 2)
107                 %if a data handling function is provided
108                     datahandler = varargin(2);
109                     datahandler(obj.backgroundTime,
110                     ...obj.backgroundData);
111             end
112         end
113         %release the data acquisition channels. sould always be ...
                called when
114         %the measurements are finished
115         function releaseChannels(obj)
116                 obj.daqObject.release();
117         end
118         %automatic cleanup function
119         function delete(obj)
120             obj.releaseChannels();
121         end
122     end
123 end
```

Appendix B

Documents

B.1 Forms

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

USC Viterbi
School of Engineering

Questionnaire Physical and Mental State

Information

Name | First Name

Date:

Please answer the questions truthfully. Your information will be kept confidential and will not be shared with outside persons.

Questions

1. How receptive and concentrative are you at the moment?

Not at all			Moderately			Extraordinarily

2. How fit are you today?

Not at all			Moderately			Extraordinarily

3. In average, how many hours sleep do you get every night?

4. How long did you sleep last night?

5. When did you go to bed yesterday and when did you get up today?

6. How drowsy are you today?

Not at all			Moderately			extraordinarily

7. How stressed do you feel today?

Not at all			Moderately			Extraordinarily

8. When did you last have a coffeinated drink?

9. How motivated are you to do the experiment

Not at all			Moderately			Extraordinarily

Acknowledgment

First of all and most importantly, I would like to thank Professor Francisco J. Valero-Cuevas. He helped my through the visa application process, cordially welcomed me in the Brain-body dynamics lab, provided me with the financial means to perform my research, coached me during the practical part of my project and finally assisted me in writing a summarizing paper. Moreover, he introduced me to Los Angeles through the participation in diverse social events.
Professor Robert Riener supervised my project on the side of my home university, ETH Zürich, and supported me with regular concise feedback.
I would like to thank all members of the Brain-body dynamics lab for upfrontly welcoming me in their lab, advising me for my presentations and participating in my experiments. Also, I'm equally thankful to all other subjects who volunteered in my projects and sacrificed a considerable amount of time.
The financial aid of my parents enabled me a memorable and educational stay at USC. Moreover, they assisted my editing and revising this thesis.
Finally, I'm grateful to Sophia and all my friends here in Los Angeles for the non-academic enchantments and solid moral support.

Bibliography

[1] S. E. Churchill. Hand morphology, manipulation, and tool use in neandertals and early modern humans of the near east. *Proceedings of the National Academy of Sciences of the United States of America*, 98:2953–2955, 2001.

[2] T. L. Kivell, J. M. Kibii, S. E. Churchill, P. Schmid, and L. R. Berger. Australopithecus sediba hand demonstrates mosaic evolution of locomotor and manipulative abilities. *Science*, 333:1411–1417, 2011.

[3] Frank R. Wilson. *The hand : how its use shapes the brain, language, and human culture*. Pantheon Books, New York, 1998.

[4] M. H. Schieber and M. Santello. Hand function: peripheral and central constraints on performance. *J Appl Physiol*, 96:2293–2300, 2004.

[5] H. van Duinen and S. C. Gandevia. Constraints for control of the human hand. *J Physiol*, 589:5583–5593, 2011.

[6] M.L. Latash. *Neurophysiological basis of movement*. Human Kinetics Publishers, 2008.

[7] M.C. Nussbaum. *Aristotle's De motu animalium*. Princeton Univ Pr, 1985.

[8] S. L. Wolf, C. J. Winstein, J. P. Miller, E. Taub, G. Uswatte, D. Morris, C. Giuliani, K. E. Light, and D. Nichols-Larsen. Effect of constraint-induced movement therapy on upper extremity function 3 to 9 months after stroke: the excite randomized clinical trial. *JAMA*, 296:2095–2104, 2006.

[9] T. D. Sanger, M. R. Delgado, D. Gaebler-Spira, M. Hallett, and J. W. Mink. Classification and definition of disorders causing hypertonia in childhood. *Pediatrics*, 111:89–97, 2003.

[10] H. Forssberg, A. C. Eliasson, H. Kinoshita, R. S. Johansson, and G. Westling. Development of human precision grip. i: Basic coordination of force. *Exp Brain Research*, 85:451–457, 1991.

[11] K. J. Cole, D. L. Rotella, and J. G. Harper. Tactile impairments cannot explain the effect of age on a grasp and lift task. *Experimental Brain Research*, 121:263–269, 1998.

[12] F. J. Valero-Cuevas, F. E. Zajac, and C. G. Burgar. Large index-fingertip forces are produced by subject-independent patterns of muscle excitation. *J Biomech*, 31:693–703, 1998.

[13] R. M. Murray, Z. Li, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC, 1994.

[14] Donald G. Lawrence and Henricus G. J. M. Kuypers. The functional organization of the motor system in the monkey. *Brain*, 91:15–36, 1968.

[15] Roger N. Lemon. Descending pathways in motor control. *Annual Review of Neuroscience*, 31:195–218, 2008.

[16] C. N. Riddle, S. A. Edgley, and S. N. Baker. Direct and indirect connections with upper limb motoneurons from the primate reticulospinal tract. *J Neursci*, 29:4993–4999, 2009.

[17] S. N. Baker. The primate reticulospinal tract, hand function and functional recovery. *J Physiol*, 589:5603–5612, 2011.

[18] K. Racz, D. Brown, and F. J. Valero-Cuevas. An involuntary stereotypical grasp tendency pervades voluntary dynamic multifinger manipulation. *J Neurophysiol*, 297:E–Pub, 2012.

[19] S.W. Ranson. *The Anatomy of the nervous system*. W.B. Saunders Company, 1920.

[20] Francsico J. Valero-Cuevas. Predictive modulation of muscle coordination pattern magnitude scales fingertip force magnitude over the voluntary range. *Journal of Neurophysiology*, 83:1469–1479, 2000.

[21] Madhusudhan Venkadesan and Francsico J. Valero-Cuevas. Neural control of motion-to-force transitions with the fingertip. *The Journal of Neuroscience*, 28:1366–1373, 2008.

[22] Kevin G. Keenan, Veronica J. Santos, Madhusudhan Venkadesan, and Francsico J. Valero-Cuevas. Maximal voluntary fingertip force production is not limited by movement speed in combined motion and force tasks. *The Journal of Neuroscience*, 29:8784–8789, 2009.

[23] Opher Donchin, Joseph T. Francis, and Reza Shadmehr. Quantifying generalization from trial-by-trial behavior of adaptive systems that learn with basis functions: Theory and experiments in human motor control. *The Journal of Neuroscience*, 23:9032–9045, 2003.

[24] John W. Krakauer, Zachary M. Pine, Maria-Felice Ghilardi, and Claude Ghez. Learning of visuomotor transformations for vectorial planning of reaching trajectories. *The Journal of Neuroscience*, 20:8916–8924, 2000.

[25] Maurice A. Smith, Ali Ghazizadeh, and Reza Shadmehr. Interacting adaptive processes with different timescales underlie short-term motor learning. *PLoS Biology*, 4:1035–1043, 2006.

[26] Robert J. van Beers. Motor learning is optimally tuned to the properties of motor noise. *Neuron*, 63:406–417, 2009.

[27] Xiaolin Liu, Kristine M. Mosier, Ferdinando A. Mussa-Ivaldi, Maura Casadio, and Robert A. Scheidt. Reorganization of finger coordination patterns during adaptation to rotation and scaling of a newly learned sensorimotor transformation. *JN Physiol*, 105:454–473, 2011.

[28] J. R. Flanagan, M. C. Bowman, and R. S. Johansson. Control strategies in object manipulation tasks. *Curr Opin Neurobiol*, 16:650–659, 2006.

[29] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

[30] S. W. Wu, M. F. Dal Martello, and L. T. Maloney. Sub-optimal allocation of time in sequential movements. *PLos One*, 4:8228–8337, 2009.

[31] B. C. Smits-Engelsman, D. Sugden, and J. Duysens. Developmental trends in speed accuracy trade-off in 6-10-year-old children performing rapid reciprocal and discrete aiming movements. *Hum Mov Sci*, 25:37–49, 2006.

[32] Francsico J. Valero-Cuevas, Madhusudhan Venkadesan, and Emanuel Todorov. Structured variability of muscle activations supports the minimal intervention principle of motor control. *Journal of Neurophysiology*, 102:59–68, 2009.

[33] N. Smaby, M. E. Johanson, B. Baker, D. E. Kenney, W. M. Murray, and V. R. Hentz. Identification of key pinch forces required to complete functional tasks. *J Rehabil Res Dev*, 41:215–224, 2004.

[34] J. J. Kutch, A. D. Kuo, and W. Z. Rymer. Extraction of individual muscle mechanical action from endpoint force. *J Neurophysiol*, 103:3535–3546, 2010.

[35] F E Yamaguchi, G T Zajac. A planar model of the knee joint to characterize the knee extensor mechanism. *J Biomech*, 22:1–10, 1989.

[36] C. M. Harris and D. M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, 1998.

[37] F. J. Valero-Cuevas. *Muscle coordination of the human index finger*. PhD thesis, Stanford University, 1997.

[38] Jason J. Kutch and Francsico J. Valero-Cuevas. Muscle redundancy does not imply robustness to muscle dysfunction. *Journal of Biomechanics*, 44:1264–1270, 2011.

[39] Madhusudhan Venkadesan and Francsico J. Valero-Cuevas. Effects of neuromuscular lags on controlling contact transitions. *Phylosophical Transactions of the Royal Society A*, 367:1163–1179, 2009.

[40] F. Mechsner, D. Kerzel, G. Knoblich, and W. Prinz. Perceptual basis of bimanual coordination. *Nature*, 414:69–73, 2001.

[41] J. van Kordelaar, E. E. van Wegen, R. H. Nijland, J. H. de Groot, C. G. Meskers, J. Harlaar, and G. Kwakkel. Assessing longitudinal change in coordination of the paretic upper limb using on-site 3-dimensional kinematic measurements. *Phys Ther*, 92:142–151, 2012.

[42] S. Dayanidhi. *BEHAVIORAL, MUSCULAR AND DYNAMICAL CHANGES IN LOW FORCE DEXTEROUS MANIPULATION DURING DEVELOPMENT AND AGING*. PhD thesis, University of Southern California, Los Angeles, CA, 2012.

[43] B. Vollmer, L. Holmstrom, L. Forsman, L. Krumlinde-Sundholm, F. J. Valero-Cuevas, H. Forssberg, and F. Ullen. Evidence of validity in a new method for measurement of dexterity in children and adolescents. *Dev Med Child Neurol*, 52:948–954, 2010.

[44] K. Mosier, C. Lau, Y. Wang, M. Venkadesan, and F. J. Valero-Cuevas. Controlling instabilities in manipulation requires specific cortical-striatal-cerebellar networks. *J Neurophysiol*, 105:1295–1305, 2011.

[45] L. Holmstrom, O. de Manzano, B. Vollmer, L. Forsman, F. J. Valero-Cuevas, F. Ullen, and H. Forssberg. Dissociation of brain areas associated with force production and stabilization during manipulation of unstable objects. *Exp Brain Research*, 215:359–367, 2011.

[46] J. B. Colby, J. D. Van Horn, and E. R. Sowell. Quantitative in vivo evidence for broad regional gradients in the timing of white matter maturation during adolescence. *NeuroImage*, 54:25–31, 2011.

[47] Robert E. Burke. Sir charles sherrington's the integrative action of the nervous system: a centenary appreciation. *Brain*, 130:887–894, 2007.

[48] Claude Ghez and John Krakauer. *The Organization of Movement*, chapter 33, pages 653–674. McGraw-Hill, 2000.

[49] John Krakauer and Claude Ghez. *Voluntary Movement*, chapter 38, pages 756–781. McGraw-Hill, 2000.

[50] AP Georgopoulos, JF Kalaska, R Caminiti, and JT Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, 2:1527–1537, 1982.

[51] R.B. Muir and R.N. Lemon. Corticospinalneurons with a specialr ole in precision grip. *Brain Research*, 261:312–316, 1983.

[52] Marc H. Raibert. Motor control and learning by the state space model. Technical report, Artificial Intelligence Laboratory, MIT, 1977.

[53] Ari Berkowitz and Zhao-Zhe Hao. Partly shared spinal cord networks for locomotion and scratching. *Integrative and Comparative Biology*, 51:890–902, 2011.

[54] K Shima, Y Tamura, Y Tsuji, A Kandori, and S Sakoda. A cpg synergy model for evaluation of human finger tapping movements. In *Conf Proc IEEE Eng Med Biol Soc*, 2011.

[55] MJ Nissen and P Bullemer. Attentional requirements of learning: evidence from performance measures. *Cogn. Psychol*, 19:1–32, 1987.

[56] E.P. Zehr and D.G. Sale. Ballistic movement: Muscle activation and neuromuscular adaptation. *Canadian Journal of Applied Physiology*, 19:363–378, 1994.

[57] J.B. Lee, T. Matsumoto, T. Othman, M. Yamauchi, A. Taimura, E. Kaneda, N. Ohwatari, and M. Kosaka. Coactivation of the flexor muscles as a synergist with the extensors during ballistic finger exetension movement in trained kendo and karate athletes. *International Journal of Sports Medecine*, 20:7–11, 1999.

[58] Martin Lakie, Carlijn A. Vernooij, Timothy M. Osborne, and Raymond F. Reynolds. The resonant component of human physiological hand tremor is altered by slow voluntary movements. *The Journal of Physiology*, 590:2471–2483, 2012.

[59] John Ramming, Pat Byrne, and George Sakona. Multi-axis load cells measure force and torque in six dimensions. *Machine Design*, Sep.:1–3, 2011.

[60] NI. Straing gauge measurement - a tutorial. Technical report, National Instruments Corporation, 1998.

[61] Interface Inc. Load cells: An introduction to the design and use of strain gage load cells. Technical report, Interface Inc., 2009.

[62] Andrei M. Reinhorn and Joseph Bracci. Multi axis load cell design and construction. Technical report, University at Buffalo, 1992.

[63] Barry Spletzer and Lisa Marron. Information package for the simplified six-axis load cell. Technical report, Sandia National Laboratories, 2000.

[64] Various. Standard practice of calibration of force-measureing instruments for verifying the force indication of testing machines, 2012.

[65] Adept. *Adeptsix 300 Robot Instruction Handbook*. Adpt Technology Inc, 2005.

[66] I.S. Helland. Pls regression and statistical methods. *Scandinavian Journal of Statistics*, 17:97–114, 1990.

[67] Thomas W. Bartel and Simone L. Yaniv. Creep and creep recovery response of load cells tested according to u.s. and international evaluation procedures. *Journal of Research of the National Institute of Standards and Technology*, 102:349–362, 1997.