Low-Latency Metallic Interference Rejection in Electromagnetic Tracking

Alex Tarng

CMU-CS-22-130

August 2022

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Cameron Riviere, Chair
Nancy Pollard

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

This work is partially supported by the National Institutes of Health under grant no. R01EB024564.

Keywords: Performance Optimization, Microsurgery, Active Stabilization, Pose Estimation, Electromagnetic Tracking, Metallic Interference, Eddy Current, Unscented Kalman Filter

Abstract

The Micron is an active handheld micromanipulator designed for membrane-peeling procedures for use in retinal surgery. The tool performs hand tremor compensation by deflecting the tip of the tool to reduce damage to the retina during microsurgery. In order to accomplish this, the control system must have a precise, low-latency measurement of the 6 degree-of-freedom pose of the tool's body. This is done via electromagnetic tracking; however, this approach is susceptible to metallic interference, disturbing the pose measurement when using a high sample rate. By using multiple carrier frequencies, the tracker can acquire both low latency data with a high frequency carrier and data less susceptible to metal interference with a low frequency carrier.

This research formulates and implements a novel dual-rate compensation method which combines high and low frequency data using a pair of unscented Kalman filters to reject metallic interference. We also provide some preliminary evaluation of the system by showing metallic interference rejection in simple test cases and discuss the viability of using this scheme in a real-time control system.

Acknowledgments

I would like to thank senior research engineer Rob MacLachlan for being my primary point of contact with the research team, guiding me through the project, teaching me the basics of Kalman filtering, providing ideas for solutions, and always being flexible. I would also like to thank my advisor, Professor Cameron Riviere, for overseeing the project, always being responsive, being accommodating when my research interests changed, and for setting up a research stipend for me. In addition, I would like to thank my secondary reader, Professor Nancy Pollard, for being responsive and flexible when I was scrambling to find a new secondary reader. Finally, I would like to thank the 5th Year Master of Science Program Administrator Tracy Farbacher, for providing guidance on completing the program and being accommodating when I ran into scheduling issues.

Contents

List of Figures

Chapter 1

Introduction

The Micron is a handheld micromanipulator designed for use in retinal membrane peeling operations. The instrument aims to reduce damage to the patient's eye caused by the surgeon's hand tremors. It does this by deflecting the end-effector in the opposite direction of hand movement to cancel out unwanted tremors, keeping the force applied to the surface of the eye below a certain threshold and stabilizing the tip of the tool [3].

In order to perform active disturbance cancellation, the instrument must know the position and orientation of the handle. This pose estimate must have high precision and resolution, as movements during microsurgery rarely exceed 1 millimeter per second. It must also be fast—the latency of the pose estimation has a direct effect on how well the tool can perform tremor cancellation. The pose estimation method must meet these requirements for the instrument to properly perform its function [4].

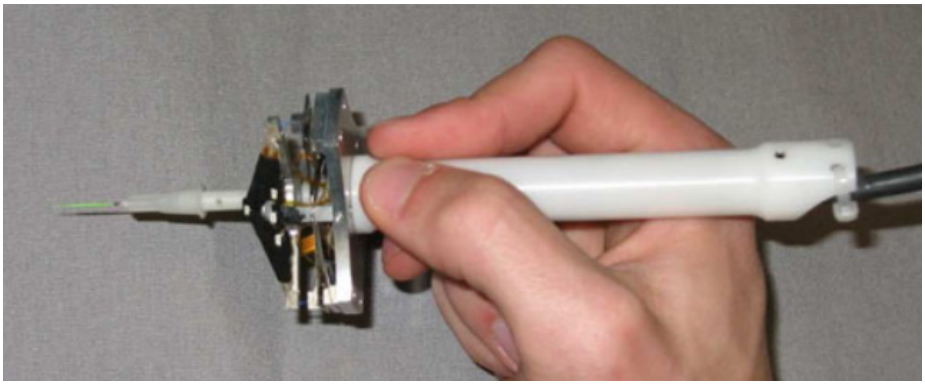

Figure 1.1: The Micron micromanipulator without its housing [1]. Piezoelectric linear actuators are used to move the tip in 6 degrees of freedom within a 4mmx4mm cylindrical workspace.

1.1 In-Loop Electromagnetic Tracker (ILEMT)

The ILEMT is our open design electromagnetic tracker for the Micron, designed to track the movement of the instrument with the speed, resolution, and precision required for hand tremor cancellation [2]. The sensor, embedded in the manipulator handle, is tracked within a 6 degree-of-freedom workspace in front of the source setup. The ILEMT is the successor of our previously developed optical tracker, ASAP, which had acceptable speed and resolution, but was deemed unsuitable for use in ophthalmic surgery due to the line-of-sight requirement, which was frequently obstructed by the surgeon and their assistants [4][5][6][7].

The setup consists of three source coils and a smaller sensor containing three sensor coils. Each sensor coil measures a characteristic magnetic coupling from each source coil, giving us a total of nine coupling measurements. We represent these measurements as a single 3x3 coupling matrix $\boldsymbol{C}$, where $\boldsymbol{C_{jk}}$ is the signed coupling measurement from source coil $\boldsymbol{j}$ into sensor coil $\boldsymbol{k}$. The sign captures the direction of the magnetic field vector with respect to the sensor coil [2].

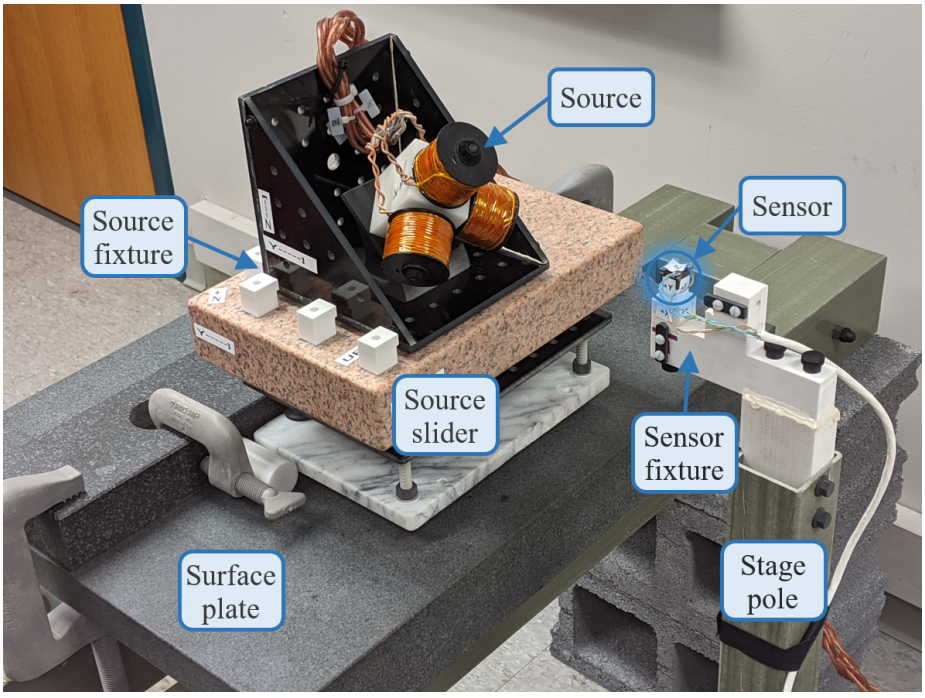

Figure 1.2: The calibration setup for the ILEMT [2]. The source coils and the sensor are mounted to fixtures, allowing for precise measurement of the source and sensor poses for calibration.

There are two different source designs for the ILEMT: *concentric* and *dipole-approximating*. The concentric source consists of a cube with three orthogonal coils wrapped around it, while the dipole-approximating source has three distinct orthogonal coils. Traditionally, in electromagnetic tracking, a concentric source is used due to the compactness of the source and simplification of the pose solution problem. However, we found that the dipole-approximating source gives a sig-

nificantly better precision than the concentric source (about half the error) [2]. In addition, there are some other considerations for using a non-concentric source specific to the Micron application, such as workspace shape and size. Throughout this study, we use the dipole-approximating source; however, the trade-off between precision and pose solution efficiency in the source design selection is an important consideration and will be discussed later.

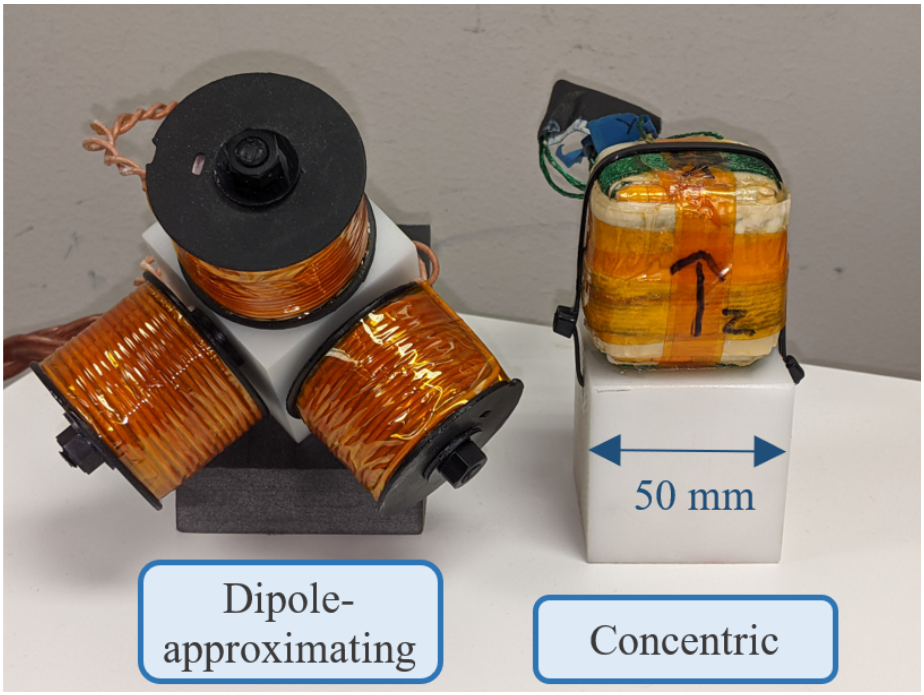

Figure 1.3: [2] Left: dipole-approximating source consisting of three distinct air-core coils mounted on a cube. Right: concentric source consisting of three orthogonal windings around a ferrite cube core.

1.2 Pose solution

The pose solution problem for the dipole-approximating source is more complex than that of the concentric source. Unlike the concentric pose solution, there is no closed form solution to finding the pose of the sensor from the coupling measurements—thus we must use nonlinear optimization. Using the dipole model, we will construct a forward kinematics function predicting the coupling output $\boldsymbol{C}$ measured from the sensor at pose $\boldsymbol{P}$, which we can then use in the nonlinear optimization solver of our choice by minimizing the error between the predicted and measured couplings.

Our tracker kinematics consists of two coordinate systems: $\boldsymbol{so}$ (source frame) and $\boldsymbol{se}$ (sensor frame), centered around the source and sensor origins respectively. For the following section, we will introduce the following notation to keep track of our coordinate systems. An arbitrary point

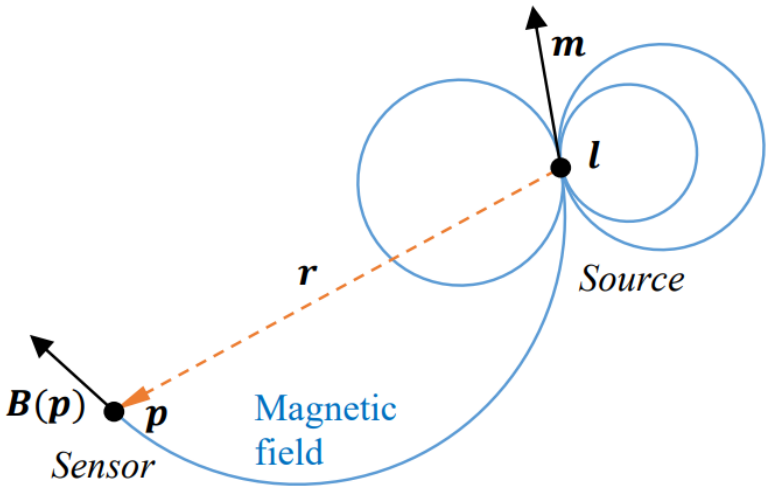

Figure 1.4: The dipole model [2]. A source coil is located at position $\boldsymbol{l}$ with moment $\boldsymbol{m}$. The sensor located at position $\boldsymbol{p}$ measures the source magnetic field vector $\boldsymbol{B}(\boldsymbol{p})$ along several axes.

$\boldsymbol{p}$ can be written as:

$$^{part}\boldsymbol{p}^{coordinates}_{index}$$

where $\boldsymbol{part}, \boldsymbol{coordinates} \in \{\boldsymbol{so}, \boldsymbol{se}\}$, and $\boldsymbol{index}$ optionally designates a sub-part. For example, the moment of sensor coil $\boldsymbol{k}$ in the source coordinate frame is represented as $^{se}\boldsymbol{M}^{so}_{k}$.

In order to calculate the predicted coupling $\boldsymbol{C}_{jk}$ from source coil $\boldsymbol{j}$ into sensor coil $\boldsymbol{k}$ at sensor pose $\boldsymbol{P}$, we must know the locations and moments of the source and sensor coils. Thorough calibration [2] provides four 3x3 matrices representing the parameters of the dipole system: $^{se}\boldsymbol{L}^{se}$, $^{se}\boldsymbol{M}^{se}$, $^{so}\boldsymbol{L}^{so}$, and $^{so}\boldsymbol{M}^{so}$. These give the locations and moments of each source and sensor coil in their respective coordinate frames.

First, we want to establish a common frame by transforming the sensor parameters into the source coordinate frame. This can be done by converting the sensor pose $\boldsymbol{P}$ into a homogeneous transform and applying it to the sensor parameters with matrix multiplication:

$$^{se}\boldsymbol{L}^{so}_{k}(\boldsymbol{P}) = \boldsymbol{P}\ ^{se}\boldsymbol{L}^{se}_{k} \tag{1.1}$$

$$^{se}\boldsymbol{M}^{so}_{k}(\boldsymbol{P}) = \boldsymbol{P}\ ^{se}\boldsymbol{M}^{se}_{k} \tag{1.2}$$

Next, we can solve for the magnetic field $\boldsymbol{B}_{jk}(\boldsymbol{P})$ experienced by sensor coil $\boldsymbol{k}$ from source coil $\boldsymbol{j}$ as a function of $\boldsymbol{P}$. According to the dipole model:

$$\boldsymbol{B}_{jk}(\boldsymbol{P}) = \mu\frac{1}{|\boldsymbol{r}|^3}[3(^{so}\boldsymbol{M}^{so}_{j} \cdot \hat{\boldsymbol{r}})\hat{\boldsymbol{r}} - \ ^{so}\boldsymbol{M}^{so}_{j}] \tag{1.3}$$

where $\boldsymbol{r} = {}^{se}\boldsymbol{L}_{\boldsymbol{k}}^{\boldsymbol{so}}(\boldsymbol{P}) - {}^{so}\boldsymbol{L}_{\boldsymbol{j}}^{\boldsymbol{so}}$, $\hat{\boldsymbol{r}}$ is the unit vector $\frac{\boldsymbol{r}}{|\boldsymbol{r}|}$, and $\boldsymbol{\mu}$ is the permeability of the medium.

Finally, each individual sensor coil will measure this magnetic field vector as a scalar voltage along the sensor moment ${}^{se}\boldsymbol{M}_{\boldsymbol{k}}^{\boldsymbol{so}}(\boldsymbol{P})$. This is modeled as the scalar coupling $\boldsymbol{C}_{\boldsymbol{jk}}$, which is given with:

$$\boldsymbol{C}_{\boldsymbol{jk}} = \boldsymbol{B}_{\boldsymbol{jk}}(\boldsymbol{P}) \cdot {}^{se}\boldsymbol{M}_{\boldsymbol{k}}^{\boldsymbol{so}}(\boldsymbol{P}) \tag{1.4}$$

Using the steps outlined in equations 1.1-1.4, we can construct the forward kinematics function to be used in the nonlinear optimization problem to solve for the pose of the sensor frame $\boldsymbol{P}$ when given the coupling matrix $\boldsymbol{C}$ measured by the sensor coils.

1.3 Metallic Interference

A major flaw of EM trackers is the intolerance of metal objects in the workspace, which disturb the coupling measurements, causing pose estimation nonlinearity [8]. This is an important issue to solve for microsurgery, as any metallic bodies such as instruments or furniture near or on the patient, surgeon, or assistants will cause the manipulator to move its tip unpredictably in trying to correct this error, potentially causing damage to the patient. While metallic interference rejection is an unsolved problem for ferromagnetic metals, such as steel, it is possible to correct the eddy-current interference caused by non-magnetic metals, such as aluminum and some stainless steels [4].

This is possible due to the fact that eddy-current is roughly proportional to carrier frequency. However, low frequency necessarily means higher latency, since the bandwidth decreases as well, thus a single low frequency carrier alone is not sufficient for tremor compensation. In order to solve this issue, we will combine the data from both high and low rate carriers to create a pose estimation that is both fast and resistant to metal interference. The remainder of the paper discusses the formulation, implementation and preliminary assessment of this dual rate scheme.

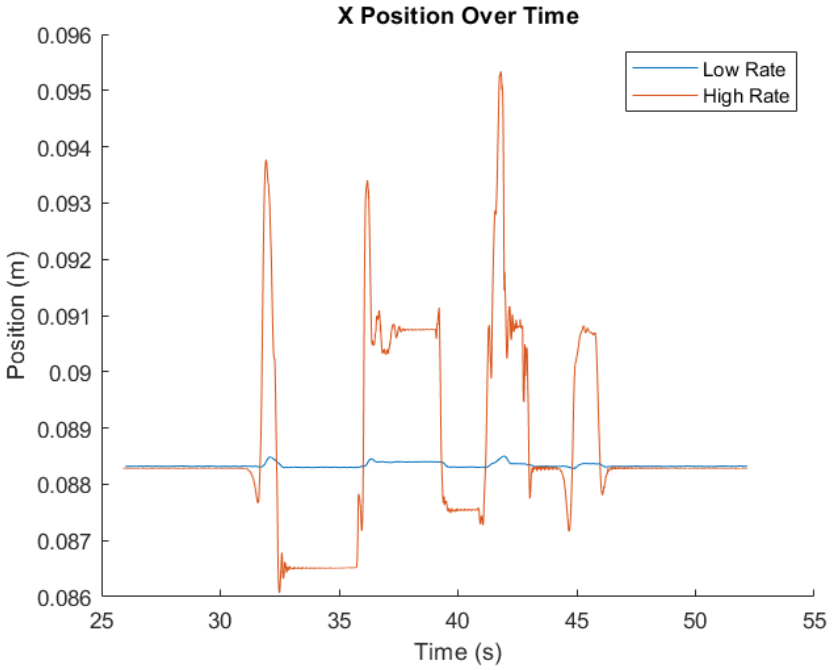

Figure 1.5: Position estimates for the X axis from high (≈10 kHz) and low (≈300 Hz) rate carriers. The data is taken with a stationary source and sensor, intermittently placing and removing an aluminum sheet between the source and sensor. The graphs should ideally be horizontal lines, but they deviate due to metal interference. Since eddy-current is roughly proportional to carrier frequency, the low rate suffers up to 33X less interference. Notice how the low rate estimate barely moves from the ideal horizontal line, but the high rate is unable to maintain an accurate pose estimate.

Chapter 2

Dual Rate Filter

2.1 Dual Rate Scheme

In order to combine the data from the high and low rate carriers, we will assume a constant bias between the high and low rate couplings. While metal interference in general does not cause a constant coupling bias, we can assume one as a local approximation, since it is rare to move more than 1mm per second during microsurgery. Therefore, this model will not be accurate in the general case, or when tracking large/fast movements.

However, we cannot directly use the high rate couplings with the low rate couplings to estimate the bias. The high rate measurements have a higher bandwidth and lower latency, causing the measurements to be out of sync with the low rate measurements. Thus, the signal processing step constructs a "high-at-low" coupling by processing the high rate measurements to have the same frequency response as the low rate measurements. The high-at-low measurement contains the high rate measurement, but with the same frequency response as the low rate measurement so that it will react to changes in the same way as the low rate. This gives us a total of three input sources: high rate couplings, low rate couplings, and high-at-low couplings.

Estimating the bias involves several steps. Every time a low rate measurement sample comes in, we can construct a corresponding high-at-low coupling for bias estimation. First, we must solve for the low rate pose estimate $\boldsymbol{P_{low}}$ from the low rate coupling such that:

$$\boldsymbol{fk}(\boldsymbol{P_{low}}) = \boldsymbol{C_{low}} \tag{2.1}$$

Then, we must solve for the 3x3 bias matrix $\boldsymbol{K}$ such that:

$$\boldsymbol{fk}(\boldsymbol{P_{low}}) + \boldsymbol{K} = \boldsymbol{C_{high-at-low}} \tag{2.2}$$

where $\boldsymbol{fk}$ is the forward kinematics function described in the introduction, $\boldsymbol{C_{low}}$ is the 3x3 low rate coupling matrix, and $\boldsymbol{C_{high-at-low}}$ is the 3x3 high-at-low coupling matrix.

We can then use the latest estimated bias $\boldsymbol{K}$ from the latest low rate measurement to make a low latency pose estimate $\boldsymbol{P_{high}}$ whenever a high rate measurement sample $\boldsymbol{C_{high}}$ comes in.

This is equivalent to solving for $\boldsymbol{P_{high}}$ such that:

$$\boldsymbol{fk}(\boldsymbol{P_{high}}) = \boldsymbol{C_{high}} - \boldsymbol{K} \tag{2.3}$$

$\boldsymbol{P_{high}}$ is our low latency, metal interference rejecting pose estimate.

2.2 Unscented Kalman Filter

There are several ways to solve equations 2.1-2.3 to complete the method. We chose to use the Unscented Kalman Filter (UKF). The UKF is an iterative linear estimator designed to filter noise like the normal Kalman filter, but is applicable to nonlinear systems unlike the Kalman filter. The latter part is important since our forward kinematics function is nonlinear, thus we cannot use the Kalman filter.

The UKF was chosen for several reasons. First of all, the iterative nature of the estimator fits with our scheme of finding the coupling bias in parallel with solving for the real pose. In addition, the Kalman filter and its variants are commonly used for applications where several inputs are combined by estimating a bias or error, such as GPS/INS integration [9]. Finally, this approach does not require the derivative of our forward kinematics function, which is a problem we haven't solved.

2.2.1 Alternatives

Nonlinear least-squares optimizer (optimize)

We originally used this approach to solve each coupling measurement for the corresponding pose, before running a Kalman filter over the resulting poses to combine high and low rate measurements and filter noise. However, this approach is incredibly slow, especially because we must use a derivative-free optimizer, since we do not have the forward kinematics derivative.

Concentric closed-form solution (kim18)

This is the closed-form solution for solving for the pose from the coupling measurement, only applicable when we are using the concentric source design. This would essentially allow us to invert the $\boldsymbol{fk}$ function in equations 2.1-2.3, and would be the fastest and simplest solution. However, as discussed before, the concentric source design is significantly less accurate than the dipole-approximating source design, as well as some other considerations for the Micron application.

Extended Kalman Filter (EKF)

The EKF is an alternate method for solving the linear estimation problem. However, using it requires the derivative of the forward kinematics function [10], which we do not have. We also

believed it would be easier to get the UKF working than the EKF, so the UKF was more suitable for the initial prototype implementation.

2.2.2 Application

The UKF uses a method called unscented transformation to pick a set of sample points, called sigma points, around the current state estimate. It then propagates each sigma point through two user-supplied functions: the state transition function and the measurement function. The outputs of these functions, as well as an input measurement, are used to create a new state estimate. This process is repeated for each incoming measurement to create an accurate state estimate that filters noise and conforms to the system dynamics defined by the user [11].

The measurement function takes in a state estimate and outputs the predicted measurement input for it. Thus, to complete the measurement functions, we simply plug in equations 2.1-2.3 above. The state transition function takes in a state estimate and a time delta, and outputs the next state estimate. We have two choices for the state transition function. We can choose to represent system dynamics by keeping track of velocity and acceleration states for each component of the pose estimate; however, this means the UKF will have to keep track of more states, as well as calculate more sigma points, since the number of sigma points scales with the number of states. This has a big impact on performance, but provides a state estimate that more accurately reflects real-world physics. Our other option is to ignore system dynamics completely, allowing us to completely skip this step, which saves computation time, but provides a rougher state estimate. This trade-off will be discussed again later.

2.3 Dual Rate UKF

Our method uses two UKFs, a low rate filter and a high rate filter. The low rate filter takes the low rate and high-at-low rate measurements as input, and uses them to estimate the bias. We then use the latest bias estimate to adjust the high rate measurements, then feed them into the high rate filter to get our final low latency, metallic interference rejecting pose estimate.

For both filters, we can optionally add an additional 9 states for system dynamics: 3 for velocity, 3 for acceleration, and 3 for angular velocity. We would also add a state transition function that uses these derivative states to estimate the change in pose over the iteration timestep. However, this will result in more sigma points being calculated per iteration and increase the runtime of each iteration.

2.3.1 Low Rate Filter

- Inputs: Low rate coupling $\boldsymbol{C_{low}}$, high-at-low coupling $\boldsymbol{C_{high-at-low}}$
- States (15-24 states):
  - Low rate pose estimate $\boldsymbol{P_{low}}$ (6-vector)
    - Low rate position estimate $\boldsymbol{x_{low}}$ (3-vector)
    - Low rate orientation estimate $\boldsymbol{\theta_{low}}$ (3-vector)
  - Coupling bias estimate $\boldsymbol{K}$ (3x3 matrix)
  - (Optional) Velocity $\boldsymbol{v_{low}}$ (3-vector)
  - (Optional) Acceleration $\boldsymbol{a_{low}}$ (3-vector)
  - (Optional) Angular velocity $\boldsymbol{\omega_{low}}$ (3-vector)
- (Optional) State transition function:
  - $\boldsymbol{x'_{low} = x_{low} + v_{low} \cdot \Delta t + a_{low} \cdot \frac{1}{2} \Delta t^2}$
  - $\boldsymbol{\theta'_{low} = \theta_{low} + \omega_{low} \cdot \Delta t}$
  - $\boldsymbol{K' = K}$
  - $\boldsymbol{v'_{low} = v_{low} + a_{low} \cdot \Delta t}$
  - $\boldsymbol{a'_{low} = a_{low}}$
  - $\boldsymbol{\omega'_{low} = \omega_{low}}$
- Measurement function:
  - $\boldsymbol{C_{low} = fk(P_{low})}$
  - $\boldsymbol{C_{high-at-low} = fk(P_{low}) + K}$

The basic low rate filter maintains 15 states: 6 for the pose estimate and 9 for the coupling bias estimate. This filter simultaneously estimates both the low rate pose and bias such that the estimated low rate and high-at-low couplings match the actual measurement inputs. The measurement function uses equations 2.1 and 2.2 to provide estimated couplings. This filter only runs one iteration per low rate measurement, which come in at a much slower frequency than high rate. Thus, the runtime of an iteration of this filter is less important than that of the high rate filter.

2.3.2 High Rate Filter

- Inputs: Unbiased high rate coupling $\boldsymbol{C_{high} - K}$
- States (6-15 states):
  - High rate pose estimate $\boldsymbol{P_{high}}$ (6-vector)
    - High rate position estimate $\boldsymbol{x_{high}}$ (3-vector)
    - High rate orientation estimate $\boldsymbol{\theta_{high}}$ (3-vector)
  - (Optional) Velocity $\boldsymbol{v_{high}}$ (3-vector)

    - (Optional) Acceleration $\boldsymbol{a_{high}}$ (3-vector)
    - (Optional) Angular velocity $\boldsymbol{\omega_{high}}$ (3-vector)
- (Optional) State transition function:
    - $\boldsymbol{x'_{high} = x_{high} + v_{high} \cdot \Delta t + a_{high} \cdot \frac{1}{2}\Delta t^2}$
    - $\boldsymbol{\theta'_{high} = \theta_{high} + \omega_{high} \cdot \Delta t}$
    - $\boldsymbol{v'_{high} = v_{high} + a_{high} \cdot \Delta t}$
    - $\boldsymbol{a'_{high} = a_{high}}$
    - $\boldsymbol{\omega'_{high} = \omega_{high}}$
- Measurement function:
    - $\boldsymbol{C_{high} - K = fk(P_{high})}$

The basic high rate filter maintains 6 states for the pose estimate. For each high rate coupling measurement, we first subtract the current bias estimate before using it as input to this filter. The measurement function uses equation 2.3 to provide the estimate unbiased high rate coupling $\boldsymbol{C_{high} - K}$. Since the high rate sample rate is significantly higher than the low rate sample rate, the single-iteration runtime for this filter is very important.

Chapter 3

Results

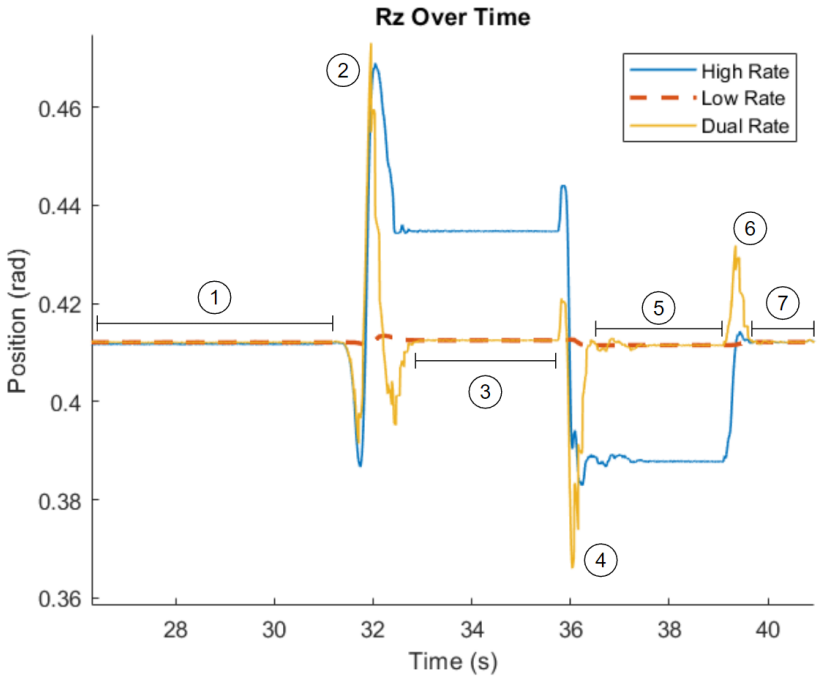

Figure 3.1: A pose estimation trace where the source and sensor are stationary and a sheet of aluminum is placed, moved, and removed between the source and sensor. This shows the Z component of the orientation part of the pose estimations.

Figure 3.1 shows the result of applying the filter to a trace with metal interference. Included are the pose estimations for the raw high rate and low rate measurements compared to the processed dual rate estimation. Sections 1 and 7 have no metal interference, sections 3 and 5 have metal interference but the metal is stationary, and 2, 4, and 6 are transients where the metal is being moved.

Since the low rate carrier is not very susceptible to metal interference, the pose estimation

barely changes when metal is introduced to the workspace (less than 2 millimeters range of output). In contrast, the high rate estimation follow the motion of the metal while it is being moved, such as during transients 2, 4, and 6 before settling at a value when the metal stops moving, such as during sections 3 and 5.

The dual rate estimation follows the high rate estimation until it learns the new bias, allowing it to settle at the low rate estimate. For example, during transient 2, the dual rate will initially follow the high rate as it goes down and then up, before gradually settling towards the baseline low rate estimate. This settling takes around 0.25-0.5 seconds in this test.

Due to the fact that it takes some time to learn the new bias, the dual rate estimate will sometimes seem to ”overshoot” when the direction of the movement of the high rate changes. For example, during transient 4, the high rate estimate dips from above the low rate to below it, and the dual rate estimate dips to a much lower value than the high rate. This is caused by the bias not changing yet, so the amount of change the dual rate experiences is the same as that of the high rate. After the initial change, the dual rate starts to learn the new bias and settles back to the low rate estimate. While, theoretically, this overshoot could pose a problem for patient safety, realistically, metal interference should not be changing so rapidly, and the surgeon should have time to react and adjust accordingly. However, it is definitely still a concern, and future testing with more realistic traces is needed to validate this conjecture. This overshoot can also been seen in transient 6 and at the end of transient 2.

3.1 Use in a Control Loop

While the instability, overshooting, and settling times in the transients are non-ideal for use in a real control loop, the fact that it settles at an accurate value is a huge improvement over the raw high rate pose estimate. In addition, the trace tested is a rather extreme case with rapid movements of metal in the workspace. In a more realistic scenario, metal interference would change much more slowly. Testing with more realistic traces remains as future work. The filter response could also be improved by further tuning the UKFs and modifying their behavior.

Chapter 4

Real-time Implementation

The biggest obstacle to getting a real-time pose estimate using this method is the runtime of a single iteration of the high rate UKF. As mentioned before, the low rate UKF runs much less frequently than the high rate UKF, and thus only the high rate runtime is a challenge. My original prototype implementation in Matlab averages at about 20 ms per high rate UKF iteration, which is much too slow for the current 1500 Hz sample rate, which requires each iteration to be less than 0.67 ms. Therefore, I spent the last few months of my research time implementing an optimized version of the dual rate UKF solution in C++.

4.1 Current Optimizations

The C++ implementation uses the Eigen matrix library and an open source UKF library developed by the SFWA UAV Challenge team [12]. The original C++ high rate filter implementation averaged about 10 ms per iteration, but by using several optimization techniques and design decisions, I was able to reduce this to 1.5 ms per iteration.

4.1.1 Removing System Dynamics

As mentioned in section 2.2.2, we have the option of completely removing the system dynamics from the filter, resulting in a rougher pose estimation, but resulting in massive performance savings. Using the `callgrind` function profiling tool revealed that the vast majority of the runtime of each iteration was taken up by various Eigen matrix operations. Around 40-50% of the total runtime was spent in the forward kinematics function, which takes a pose and several calibration matrices and predicts the coupling measurement. This is due to the fact that the forward kinematics function is called for every sigma point, causing this function to be run many times per iteration.

By ignoring system dynamics, we are able to reduce the number of states from 15 to 6. In addition to reducing the size of many of the matrices used internally in the UKF, this also reduced the number of sigma points calculated per iteration. As mentioned above, the call count of the expensive forward kinematics function is proportional to the number of sigma points calculated.

Thus, removing system dynamics provides us with a significant speedup, cutting around 5 ms from the original iteration runtime.

4.1.2 Reducing Redundant Calculation

Since the forward kinematics function comprises such a significant portion of the total runtime, I focused on optimizing the internals of this function. I found that the initial, most intuitive implementation of the forward kinematics function had many redundant calculations. For example, slicing calibration matrices into the components for each source/sensor is an expensive operation, and since the calibration matrices are static, these operations can be done during a single preprocessing step upon initialization of the UKF. In addition, the original equations included recalculation of operations such as norms and dot products to simplify the equations, but by reducing these to only be calculated once we can save a decent amount of time.

Lastly, the main body of the forward kinematics function consists of a double-nested for loop iterating over the sources and sensors. Intuitively, due to how the calibration matrices are stored, it makes sense to iterate in row-major order, meaning we iterate over sources, then sensors. However, by reversing the order of the loops, we reduce the number of times we have to calculate ${}^{se}\boldsymbol{L}_{\boldsymbol{k}}^{\boldsymbol{so}}(\boldsymbol{P})$ and ${}^{se}\boldsymbol{M}_{\boldsymbol{k}}^{\boldsymbol{so}}(\boldsymbol{P})$ by a factor of 3. Since both of these terms require matrix multiplication to calculate, we save a significant amount of time by doing this. This also offsets the cost of iterating in column-major order, which is negligible due to the small size of the calibration matrices (3x3).

4.1.3 UKF Type

Another variable affecting runtime is the type of UKF used. The SFWA library, in addition to the standard UKF, implements the Square-Root UKF (SR-UKF), which is an optimized version of the UKF meant to provide better performance and added numerical stability. However, switching to the Square-Root UKF actually had a negative impact on performance, taking 2.5 ms for the SR-UKF and 2.0 ms for the parameter estimation SR-UKF, as opposed to 1.5 ms for the standard UKF. This is likely due to the fact that the SR-UKF is meant for systems with many states/parameters, as the computational complexity of the standard UKF is $\mathcal{O}(\boldsymbol{L^3})$, and the SR-UKF has complexity $\mathcal{O}(\boldsymbol{L^2})$, where $\boldsymbol{L}$ is the number of states/parameters. While the SR-UKF can achieve 20% faster performance than the UKF on machine learning applications, where many parameters are being estimated simultaneously [13], our system has only 6-15 states, causing other overheads to dominate the computational cost.

4.2 Options for Future Speedup

While 1.5 ms is a good starting point and in the right order of magnitude of our goal of 0.67 ms, it is not yet fast enough to achieve true real-time pose estimation. Unfortunately, I ran out of time to be able to meet this goal, but there are several options to explore in the future to achieve

further speedup.

One of these is trying to further reduce the amount of time spent in the forward kinematics function, which is still the most expensive part of the filter. There are several techniques we could attempt. One of them is implementing the internal forward kinematics calculation without using high level libraries like Eigen. While Eigen claims to be fast [14], there is still a cost being paid for the matrix abstraction, and when the function is being run so many times, small costs will add up. Since the most complex operations are matrix multiplication and homogenous matrix construction, this would be fairly simple to implement. Another possible route is using parallelism to evaluate the sigma points allowing us to run many forward kinematics calls at the same time. This is the only part of the UKF that could be parallelized, since everything else has sequential dependencies.

In addition to this, the UKF library itself may not be fully optimized. The SFWA UKF library is designed with usability in mind rather than speed, which means the code makes sacrifices in runtime to provide a simple interface for the user. The wiki page of the Github repository mentions an alternate implementation written in plain C, which is optimized for their hardware [12]. In addition to trying this other implementation, it would also be possible to roll our own UKF implementation, which has the advantage of being tailored to our use case, but may be too difficult or time-consuming for an unknown amount of speedup.

Finally, it may be necessary to use a different optimization technique than the UKF for the high-rate solver. In this case, we would still use the UKF for the low-rate filter, but we would replace the high-rate UKF with another solver. One option is the kim18 solver. While this would require the use of the concentric source design, which is susceptible to the issues discussed in section 1.1, the closed-form solution might be much faster than a UKF iteration. Another option is using the Extended Kalman Filter (EKF). The EKF may be faster than the UKF due to not having to evaluate the forward kinematics function at so many sigma points. However, the biggest challenge to using the EKF is whether or not we can figure out the derivative of the forward kinematics function. We might need to reformulate the function to make it easier to differentiate, or use a non-symbolic way of differentiating.

4.3 Other Considerations

Another option for achieving a real-time pose estimate is to slow down the frequency of the high rate carrier. This would allow the filter to catch up with the measurement sample rate; however, this has the effect of adding latency to the pose estimate, which may affect the Micron's ability to cancel tremors. Ideally, we would want to stay above 1000 Hz so we do not overly impede the latency.

Another consideration is that even bringing down the filter's single iteration runtime to run at the high rate frequency may not be enough for good control. Pose estimation is not the only step required; other steps such as signal processing and controls calculations need to happen as

well. Unlike pose estimation, their inputs do not depend on the output of the previous iteration, so these steps can be pipelined. However, doing this will add extra latency, again affecting how well the system can operate.

4.4 Latency Comparison

How much better is the latency of the dual rate compared to the low rate? To answer this question, we will look at 3 relevant properties: computation time, sampling delay, and group delay.

For the computation time, we will assume 1.5 ms for both filters, since that is the iteration runtime of the most optimized high rate filter at this point in time, and the low rate filter should be similar if it doesn't have to estimate bias in parallel. The sampling delay is around half the time between samples. Since our high rate sample rate is 1500 Hz and our low rate is slower by a factor of 128, we get around 0.3 ms for dual rate and 43 ms for low rate. Lastly, we have the group delay, which we measure by giving the filter a digitally constructed ideal step input and measuring the 2% settling time of the filter response.

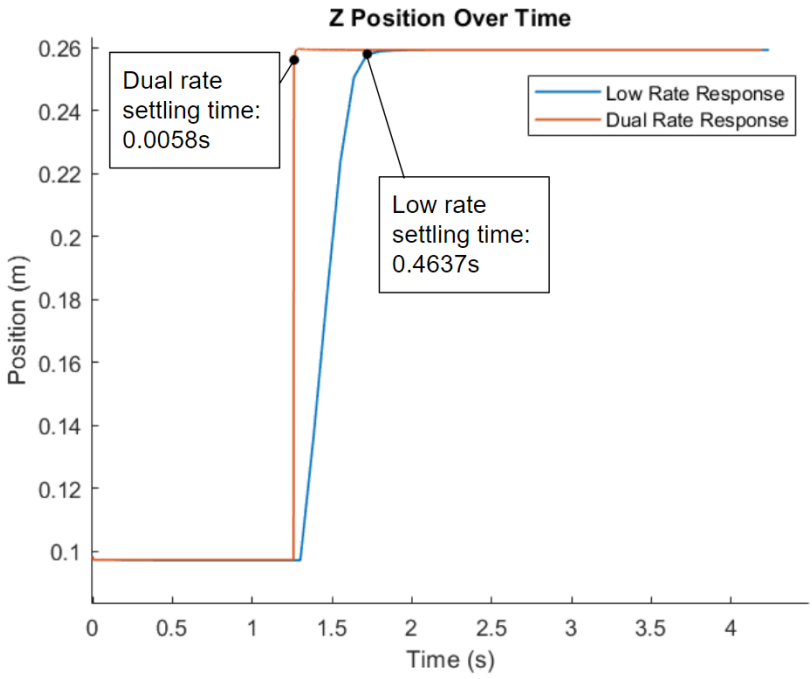

Figure 4.1: Step response for the low and dual rate filters. Computation time and sampling delay are added in, shifting the responses slightly towards the right. The time at which each filter settles within 2% of the end pose gives the total latency of the method.

Figure 4.1 gives us the total latency numbers: 5.8 ms for the dual rate, and 463.7 ms for the low rate. In this case, the dual rate filter gives us a faster response by a factor of 80, well-justifying the decision to use high rate data to lower the latency of the pose estimation.

Chapter 5

Conclusion

This research introduces and implements the dual rate filter for rejecting metallic interference when solving for pose in an electromagnetic tracking system. We showed the results of running the filter on a basic trace with a stationary source and sensor and rapidly changing metal interference. These results showed that, after the metal interference stabilizes, the filter is able to quickly settle back to the correct pose estimate.

The research also implements a second, optimized implementation of the filter with the goal of eventually using it in a real-time control loop. We discussed the tradeoffs between performance and other desirable traits of changing certain system parameters, such as source coil design (*concentric* vs. *dipole-approximating*), nonlinear solver type (*UKF* vs. *optimize* vs. *kim18* vs. *EKF*), UKF type (*standard* vs. *square-root*), and whether or not to simulate system dynamics.

While the current iteration of the optimized implementation is not yet sufficiently fast enough to achieve real-time control, we discussed the options we can explore in the future to achieve our performance goals. These include implementing an optimized version of library code ourselves, using parallelism to calculate sigma points, and using a different optimization technique than the UKF for the high-rate solver. We also discuss the possibility of lowering the high carrier frequency, allowing the filter to catch up and lowering our bar for filter performance.

There are still many additional steps remaining before the ILEMT is complete. The dual rate method needs to be tested using more realistic traces, allowing us to evaluate the tuning of the UKFs and the validity of our assumptions, including the constant bias assumption. The signal processing step, which reads voltages from the sensors and converts them into coupling measurements, also requires an optimized implementation. These steps then need to be integrated with the existing hybrid position/force control [15] and virtual fixtures [1] implementations to form a complete control system. The system must then be tested with and without metal interference to evaluate the filter for latency, accuracy, settling time, and other requirements in order to tune the filter for practical use.

Bibliography

[1] Brian C Becker, Robert A MacLachlan, Louis A Lobes, Gregory D Hager, and Cameron N Riviere. Vision-based control of a handheld surgical micromanipulator with virtual fixtures. *IEEE Transactions on Robotics*, 29(3):674–683, 2013. (document), 1.1, 5

[2] Robert A MacLachlan, Claudia Pelle, Ralph L Hollis, Elena De Momi, and Cameron N Riviere. Calibration and characterization of electromagnetic position and orientation trackers. *IEEE Transactions on Instrumentation and Measurement*, submitted. (document), 1.1, 1.2, 1.1, 1.3, 1.4, 1.2

[3] Sungwook Yang, Robert Maclachlan, and Cameron Riviere. Manipulator design and operation of a six-degree-of-freedom handheld tremor-canceling microsurgical instrument. *IEEE/ASME Transactions on Mechatronics*, 20:761–772, 04 2015. doi: 10.1109/TMECH.2014.2320858. 1

[4] Robert A. MacLachlan, Nicholas Parody, Shohin Mukherjee, Ralph L. Hollis, Cameron N. Riviere, Joseph N. Martel, and Louis A. Lobes. Electromagnetic tracker for active handheld robotic systems. In *2016 IEEE SENSORS*, pages 1–3, 2016. doi: 10.1109/ICSENS.2016.7808415. 1, 1.1, 1.3

[5] Robert A. MacLachlan and Cameron N. Riviere. High-speed microscale optical tracking using digital frequency-domain multiplexing. *IEEE Transactions on Instrumentation and Measurement*, 58(6):1991–2001, 2009. doi: 10.1109/TIM.2008.2006132. 1.1

[6] Robert A. MacLachlan, Brian C. Becker, Jaime Cuevas Tabares, Gregg W. Podnar, Louis A. Lobes, and Cameron N. Riviere. Micron: An actively stabilized handheld tool for microsurgery. *IEEE Transactions on Robotics*, 28(1):195–212, 2012. doi: 10.1109/TRO.2011.2169634. 1.1

[7] Sungwook Yang, Louis Lobes, Joseph Martel, and Cameron Riviere. Handheld automated microsurgical instrumentation for intraocular laser surgery. *Lasers in surgery and medicine*, 47, 08 2015. doi: 10.1002/lsm.22383. 1.1

[8] Mark A. Nixon, Bruce C. McCallum, W. Richard Fright, and N. Brent Price. The Effects of Metals and Interfering Fields on Electromagnetic Trackers. *Presence: Teleoperators and Virtual Environments*, 7(2):204–218, 04 1998. doi: 10.1162/105474698565587. URL `https://doi.org/10.1162/105474698565587`. 1.3

[9] J.L. Crassidis. Sigma-point kalman filtering for integrated gps and inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 42(2):750–756, 2006. doi: 10.1109/TAES.2006.1642588. 2.2

[10] Dan Simon. Optimal state estimation: Kalman, h, and nonlinear approaches. 2006. 2.2.1

[11] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. Spie, 1997. 2.2.2

[12] Daniel Dyer et al. Ukf. `https://https://github.com/sfwa/ukf`, 2022. 4.1, 4.2

[13] Rudolph Van Der Merwe and Eric A Wan. The square-root unscented kalman filter for state and parameter-estimation. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 6, pages 3461–3464. IEEE, 2001. 4.1.3

[14] Main page, Jan 2022. URL `https://eigen.tuxfamily.org/index.php?title=Main_Page`. 4.2

[15] Trent Wells, Sungwook Yang, Robert Maclachlan, Louis Lobes, Joseph Martel, and Cameron Riviere. Hybrid position/force control of an active handheld micromanipulator for membrane peeling: Position/force control of active handheld micromanipulator. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 12, 05 2015. doi: 10.1002/rcs.1659. 5