Delft University of Technology

The Mixed Ornstein-Uhlenbeck Model in Target Tracking for Mean-Reverting Motion and Weaving Motion

A 2nd Order Continuous Auto-Regressive Model

by

Koen Bavelaar

to obtain the Master's degree in Electrical Engineering at Delft University of Technology,
to be defended on Friday 1 May 2026.

Supervisors		**Thesis Committee**	
Dr. ir. Hans Driessen	TU Delft	Dr. Francesco Fioranelli	TU Delft
Roberto Arnaiz Burgueño	Thales	Dr. ir. Hans Driessen	TU Delft
Hannah Garcia Doherty	Thales	Dr. ir. Manon Kok	TU Delft
		Roberto Arnaiz Burgueño	Thales

Abstract

Targets tracked by radar may exhibit mean-reverting motion, such as motion along a nominal trajectory or weaving behavior. Conventional target tracking models typically do not account for this characteristic, which limits estimation and prediction performance. This study investigates the mixed Ornstein–Uhlenbeck model, also known as the second-order continuous auto-regressive model, as a framework for representing mean-reverting motion to model targets that follow a nominal trajectory or to model targets that display oscillatory motion. The model contains two unknown transition parameters per dimension, which are estimated offline by maximum likelihood estimation and online by joint estimation and a particle filter. Results show that joint estimation provides the best performance in both state and parameter estimation. Joint estimation is applied to real-world weaving motion from jet skis, water scooters, and RHIBs. The proposed approach improves position estimates and, in particular, velocity estimates, while also providing accurate trajectory predictions. These findings demonstrate that the mixed Ornstein–Uhlenbeck model can significantly enhance target-state estimation and prediction for mean-reverting motion.

Contents

Nomenclature

ARMA	Auto-Regressive Moving-Average
AR	Auto-Regressive
CARMA	Continuous Auto-Regressive Moving-Average
CI	Confidence Interval
CRLB	Cramer-Rao Lower Bound
EKF	Extended Kalman Filter
EM	Expectation-Maximization
i.i.d.	Independent identically distributed
IMM	Interacting Multiple-Model
iOU	Integrated Ornstein-Uhlenbeck
JE	Joint Estimation
KF	Kalman Filter
MA	Moving-Average
MLE	Maximum Likelihood Estimation
MM	Multiple-Model
MOU	Mixed Ornstein-Uhlenbeck
MSE	Mean Square Error
NCV	Nearly Constant Velocity
OU	Ornstein-Uhlenbeck
RHIB	Rigid hull inflatable boat
RMSE	Root Mean Square Error
std	Standard deviation
UKF	Unscented Kalman Filter

Chapter 1

Introduction

A surveillance radar looks at a large volume around it to track targets. Targets are objects of interest, like airplanes or ships, and target tracking consists of monitoring a target's position and velocity in real time. Sometimes tracking also includes the analysis or prediction of the target's trajectory. The main purpose of a surveillance radar is creating situational awareness. When a track on a new target starts, the target's trajectory and class can be estimated based on its kinematic behavior, which can be used for intention prediction and threat identification.

A radar works by transmitting electromagnetic waves. The position and velocity of an object are calculated based on the received reflections caused by the target. The raw radar detections are not enough to track a target. Firstly, the radar also receives reflections by clutter, in which we are not interested. For example, if a radar is used for drone detection it may also detect clutter from birds or raindrops. Secondly, the measurements of the position and velocity are not exact but contain measurement noise. Lastly, measurements are not continuous, instead they have intervals of a few seconds, and sometimes the radar misses detections which makes this interval even longer. To translate these raw detections into target trajectories, a tracking algorithm is needed.

A tracking algorithm consists of multiple steps. First, data association needs to be done. The algorithm decides to which target a measurement belongs or if a measurement is clutter. Secondly, the algorithm uses subsequent measurements to estimate the state of a target, which is also called filtering. The state is often defined by the position and velocity of a target, and sometimes also the acceleration. If necessary, the tracking algorithm can also make predictions or analyze the trajectory. For data association, filtering, and prediction a tracking model which describes the behavior of a target is used. A better tracking model can improve the state estimation and prediction, which in turn improves classification and trajectory analysis. It can also lead to more efficient sensor resource management as fewer measurements may be needed for the same tracking accuracy. For these reasons it is desirable to improve on current tracking models, which is the objective of this research.

1.1 State of the Art

The workhorse of the tracking models is the nearly constant velocity (NCV) model, which can be used for straight-ahead movement as it assumes the acceleration to be white noise. The model is described by

$$a(t) = w(t). \tag{1.1}$$

Here, a is the acceleration, and $w \sim N(0, q)$ is Gaussian process noise. A state-space approach can be used in combination with a Kalman filter to estimate the state. The state is described by $\mathbf{s} = [p, v]^\top$ where p is the position and v the velocity. The derivatives of the state are given by $\dot{p} = v$ and $\dot{v} = a = w$. The NCV model has many advantages: it is robust, computationally efficient and easy to tune and implement. However, it is a very general model as it does not take into account patterns in the acceleration. Because of this it is bad at predicting the future states of a target. For a target following the NCV model the process noise variances are given by

$$\sigma_p^2 = \frac{qt^3}{3}, \tag{1.2}$$

$$\sigma_v^2 = qt. \tag{1.3}$$

So the uncertainty for the position grows with t^3 and the uncertainty for velocity with t, leading to a high prediction uncertainty for large t. A high prediction error can be problematic if the intention of the target needs to be identified. Also, when there are missed detections, coverage gaps or targets fly close to each other the radar may "lose" the target. The problem with the long-term prediction uncertainty in coverage gaps is illustrated in Figure 1.1. Two airplanes are detected by radar 1, then fly out of the coverage region. Later two airplanes are detected by radar 2. For us, it is clear that the top airplane detected by radar 2 is likely the orange/top airplane detected by radar 1. But the NCV model cannot draw such conclusions due to the high prediction uncertainty. An alternative model with a lower prediction error is the Ornstein-Uhlenbeck model, this model will be discussed next.

1.2 Ornstein-Uhlenbeck Model

The Ornstein-Uhlenbeck (OU) model is a stochastic process that wanders randomly but is pulled back to a long-term average. In target tracking, this pull toward a mean

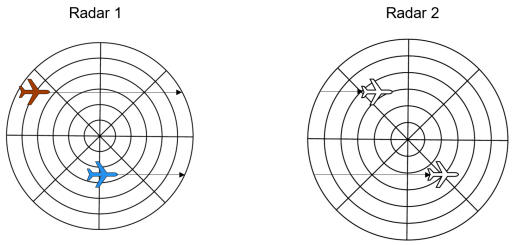
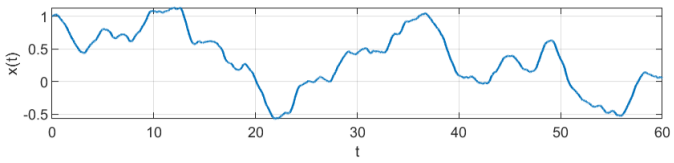
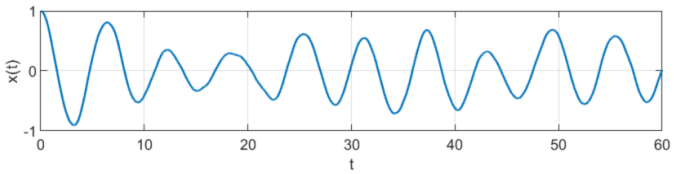


Figure 1.1: Two radars with a coverage gap in between, the airplanes detected by radar 2 are the same airplanes detected by radar 1, but the NCV prediction is not accurate enough to draw this conclusion.

may be observed in position or velocity. For example, the velocity of a target has an upper limit and may tend to stay around a certain mean value. Millefiori et al. [33] show that the velocity of maritime traffic is described well by an OU model. Forti et al. [18] show how to exploit this behavior to improve target tracking and anomaly detection, which works especially well under lower detection probabilities or data gaps. With data gaps, the velocity prediction uncertainty reaches a steady state, instead of increasing indefinitely with time. However, the position prediction uncertainty does not reach a steady state. The



(a) Mean-reverting motion.



(b) Weaving motion.

Figure 1.2: Simulated MOU trajectories.

OU model described by Millefiori et al. has mean reversion in velocity. Coraluppi et al. extended the OU model to also have mean reversion in position. This model is called the mixed Ornstein-Uhlenbeck model (MOU). The mean reversion in position allows for a nominal trajectory. If the target deviates from its nominal trajectory it will eventually drift back. The accurate state predictions improve data association under data gaps or coverage gaps [12]. Moreover, on the short term, some motion patterns are well described by a MOU model. One of these motion patterns is oscillatory motion, also called a weave. Mean reversion in position allows for such motion, which can improve state estimation of targets following a sinusoidal pattern [13]. Two examples of MOU process trajectories are shown in Figure 1.2. These patterns resemble some motion patterns observed in targets. This observation naturally leads to the research question of this thesis.

1.3 Research Question

The research question is formulated as follows:

> "Can the mixed Ornstein-Uhlenbeck model improve target tracking?"

The corresponding sub-questions are:

- How does the mixed Ornstein-Uhlenbeck model compare to other tracking models?
- How can the mixed Ornstein-Uhlenbeck model be implemented in a tracking filter?
- How can the parameters of the mixed Ornstein-Uhlenbeck model be estimated?
- Does the mixed Ornstein-Uhlenbeck model improve state estimation and prediction?

Chapter 2

Dynamic Models for Target Tracking

In target tracking a state-space model is used to model the target dynamics. Because the time intervals between measurements may be irregular, it is important to use a continuous state-space model. For simplicity the models are analyzed in 2D (x, y) instead of 3D (x, y, z), but a 2D model can easily be extended to 3D. The state that is chosen depends on which model is used. If the state contains the position and velocity the 2D state is described by the state-vector

$$\mathbf{s}(t) = [x(t),\, y(t),\, \dot{x}(t),\, \dot{y}(t)]^T. \tag{2.1}$$

Here x and y denote positions, and $\dot{x}$ and $\dot{y}$ denote velocities. Some models, like the Singer model, also contain acceleration in the state. The general continuous model of a target can be described by

$$d\mathbf{s}(t) = \mathbf{F}\,\mathbf{s}(t) + d\mathbf{w}(t). \tag{2.2}$$

Here $\mathbf{F}$ is the transition matrix and $\mathbf{w}$ the process noise. Radar measurements are discrete so the continuous model has to be transformed to a discrete model, which is described by

$$\mathbf{s}_{k+1} = \mathbf{A}_k \mathbf{s}_k + \mathbf{w}_k. \tag{2.3}$$

Where the transition matrix $\mathbf{A}_k$ and the process noise $\mathbf{w}_k$ depend on $\tau_k = t_{k+1} - t_k$.

Kalman filter If the discrete model is known, the state can be filtered by a Kalman filter (KF) [5][p. 158-168]. A KF is unbiased and has the lowest possible estimation variance under the assumption that the process noise, measurement noise and the prior of the state are Gaussian distributed, and the transitions and measurements are linear. However, there exist a range of nonlinear KF variations [9]. Namely the extended KF (EKF), unscented KF (UKF), cubature KF and ensemble KF. The EKF is simple and efficient and performs well under nearly-linear, smooth functions. The UKF works better for systems that have more nonlinearity. The cubature KF is designed for high-dimensional spaces and improves the numerical stability of the UKF. Lastly, the ensemble KF can be implemented for highly nonlinear models. This KF uses a Monte Carlo ensemble to estimate the posterior.

Multiple model filter A single model is not sufficient for target tracking. For example, a model that performs well at filtering straight-ahead motion is not good at filtering turns. To address this, multiple models can be run in parallel in a multiple model (MM) filter [5][p. 221-233]. This filter selects (or weighs) the models according to their likelihood. In an interacting multiple model (IMM) filter, the state estimates from the parallel filters are probabilistically combined to approximate the posterior and give the highest likelihood state estimation. A MM for target tracking often includes a NCV, constant acceleration and coordinated turn model. In addition to these models, the Singer model [37], which assumes acceleration is correlated, is also popular in target tracking. Because of the MM framework, a single model does not need to provide a universal description of target motion. It may still improve performance if it captures the dynamics of a specific subset of motion regimes.

In the next section, an overview of common tracking models is given and the properties of these models are discussed. Also the OU model and variations thereof, namely the integrated OU and MOU, are discussed.

2.1 NCV Model

The NCV model is suitable for straight-ahead motion and assumes white-noise acceleration. The dynamics are described by Blackman and Popoli [5][p. 203-205]. The acceleration is given by

$$a(t) = w(t). \tag{2.4}$$

Here, a denotes the acceleration, and $w \sim N(0, q)$ the process noise. The continuous-time state-space model of a single dimension is given by

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} x(t) \\ v(t) \end{bmatrix}}_{\mathbf{s}} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t). \tag{2.5}$$

Here, x denotes position and v velocity. To obtain a discrete-time version of this continuous-time tracking model, we use the solution of $\mathbf{s}$ in

$$\dot{\mathbf{s}}(t) = \boldsymbol{F}\boldsymbol{s}(t), \tag{2.6}$$

which is given by

$$\boldsymbol{s}(t) = e^{\boldsymbol{F}t}\boldsymbol{s}_0. \tag{2.7}$$

For the NCV model, the corresponding discrete-time form is [31]

$$\begin{bmatrix} x \\ v \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \tau_k \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}_k + \mathbf{w}_k. \tag{2.8}$$

$$\tag{2.9}$$

Here, $\tau_k = t_{k+1} - t_k$ denotes the time interval, and $\mathbf{w}_k$ the process noise, whose covariance matrix is given by

$$\mathbf{Q}_w = q \begin{bmatrix} \frac{\tau^3}{3} & \frac{\tau^2}{2} \\ \frac{\tau^2}{2} & \tau \end{bmatrix}. \tag{2.10}$$

2.2 Coordinated Turn Model

The coordinated turn model describes horizontal motion with an approximately constant velocity and turn rate. The dynamics are given by Blackman and Popoli [5][p. 205-207]

$$\dot{v}_x(t) = -\omega v_y(t), \quad \dot{v}_y(t) = \omega v_x(t). \tag{2.11}$$

The target follows a circular trajectory described by

$$v = \sqrt{v_x^2 + v_y^2}, \qquad R = \frac{v}{|\omega|}. \tag{2.12}$$

Where R is the radius of the circle. In practice, a target does not follow a perfect circle. To account for this process noise can be added to the turn rate and velocity.

2.3 OU Model

The OU model for position is given by

$$\dot{x}(t) = -\gamma x(t) + w(t). \tag{2.13}$$

The OU process is mean reverting in the position. A positive position causes a negative velocity pulling the position back to 0. The transfer function from input w to output x is given by

$$H(s) = \frac{X(s)}{W(s)} = \frac{1}{s+\gamma}. \tag{2.14}$$

The corresponding impulse response, shown in Figure 2.1, is obtained by taking the inverse Laplace transform of $H(s)$ given by

$$h(t) = \mathcal{L}^{-1}\left\{\frac{1}{s+\gamma}\right\} = e^{-\gamma t}u(t). \tag{2.15}$$

This model is not well suited to target tracking, since the velocity is fully determined by the position. In practice, the velocity is strongly correlated with its past values and typically changes gradually between samples. Because this behavior is not captured by the model, it can produce large estimation errors and will not be practical.

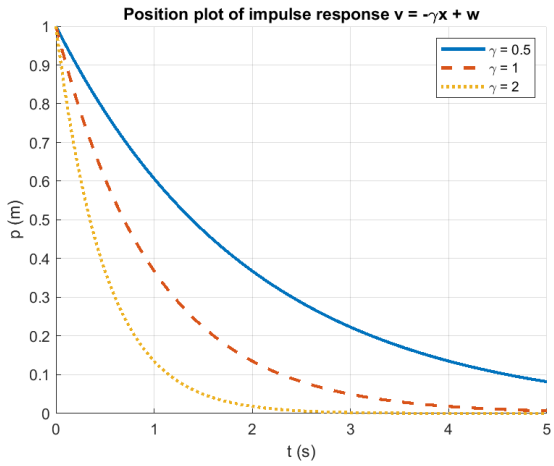


Figure 2.1: Position impulse response of the OU model.

2.4 Integrated OU Model

If velocity follows an OU model, then position follows an integrated OU (iOU) model, with acceleration given by [15]

$$\ddot{x}(t) = -\gamma \dot{x}(t) + w(t). \tag{2.16}$$

The iOU process is mean reverting in velocity. A positive velocity causes a negative acceleration pulling the velocity back to zero. The impulse response, shown in Figure 2.2, is the integral of the OU impulse response and is given by

$$h(t) = \int_0^t e^{-\gamma s}\, ds = \frac{1 - e^{-\gamma t}}{\gamma}. \tag{2.17}$$

Since the position does not revert to zero, this model can-

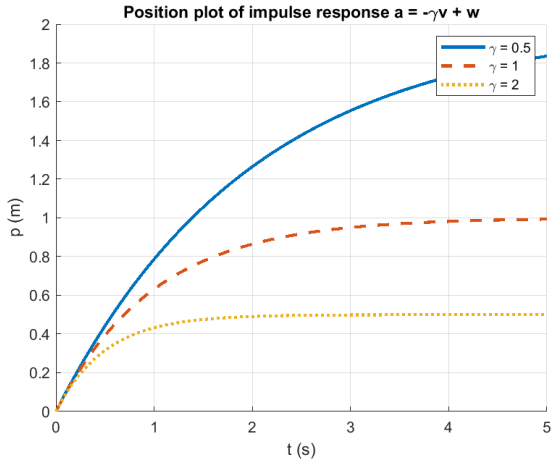


Figure 2.2: Position impulse response of the iOU model.

not follow a nominal trajectory, and the long-term position uncertainty does not reach a steady state. Nevertheless, it works well in modeling targets that follow a mean velocity and it provides a significant predictive improvement over the NCV model. The iOU model has been applied to maritime traffic in several studies, which are discussed next.

Applications

Millefiori et al. [33] apply the iOU model to cargo, tanker, and passenger vessels for which they analyzed

respectively 1370, 370 and 150 trajectories. In total, they collected about 200,000 automatic identification system messages, also known as AIS messages, for trajectories northeast of the Strait of Gibraltar. The velocity reversion parameter γ is estimated using offline maximum likelihood estimation. The estimation is not available in closed form but is done iteratively. The prediction error is then evaluated for different time horizons and they test what percentage of observations fall within the predicted 95% confidence interval. For shorter time horizons, the targets lie within the predicted 95% region for both the NCV and iOU model about 95% of the time. For longer time horizon this is only the case for the iOU model. An interesting observation is that mean reversion in the direction perpendicular to the trajectory is faster than in the parallel direction for all three classes of ships. Also, there seems to be a difference in parameters between the classes, especially the passenger vessel parameters seem to differ from the other two classes. Although it is not stated whether this difference is statistically significant, it hints at the possibility of target classification based on the model parameters.

Coscia et al. [15] use the iOU model for detecting waypoints in maritime traffic. Waypoints are regions where changes in velocity or heading are observed. Ships often navigate in straight lines between waypoints so the majority of maritime traffic is very regular, and the iOU model describes this behavior well as it pulls velocity back to a mean value. Based on the iOU model, a likelihood ratio is computed for the hypotheses of a mean change and no mean change. If the likelihood ratio crosses the threshold, a waypoint is detected.

Forti et al. [18] applied the OU model to both tracking and anomaly detection in maritime traffic. They conclude that their proposed model outperforms the classical approach, in which models such as the NCV model and the constant turn-rate model are run in parallel, in both state estimation and anomaly detection. The improvement is particularly pronounced when the detection probability is low or when data gaps are present. This study demonstrates that the iOU model can improve state estimation and anomaly detection in maritime traffic compared with the classical multiple model approach.

2.5 Singer Model

The Singer model can be used for smoothly varying, correlated acceleration. The dynamics are described by Blackman and Popoli [5][p. 200-203]. The model is given by

$$\dot{a}(t) = -\frac{1}{\tau}a(t) + w(t). \tag{2.18}$$

The parameter τ is the time constant: after τ seconds, the acceleration has decreased to $e^{-1} \approx 0.37$ of its initial value. The position impulse response of the Singer model, denoted by $h(t)$, is obtained by integrating the impulse response of the iOU model. It is shown in Figure 2.3 and is given by

$$h(t) = \int_0^t \frac{1 - e^{-\gamma\sigma}}{\gamma}\, d\sigma = \frac{t}{\gamma} - \frac{1 - e^{-\gamma t}}{\gamma^2}. \tag{2.19}$$

γ in this formula is the reciprocal of τ, so $\gamma = \frac{1}{\tau}$. Clearly,

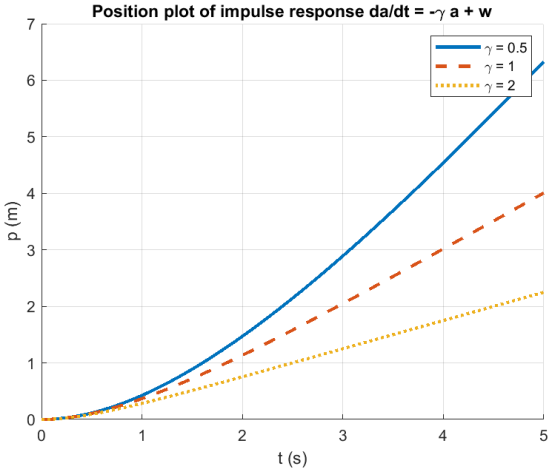


Figure 2.3: Position impulse response of the Singer model.

this process is not mean reverting in position and therefore bad at long-term predictions. However, in short-term state estimation the Singer model can outperform other models. This is because in practice, acceleration is often temporally correlated over short time intervals.

2.6 MOU Model

The mixed Ornstein-Uhlenbeck (MOU) process is mean reverting in both the position and velocity for $\gamma_1, \gamma_2 > 0$, but oscillatory motion may occur. The model is given by [13]

$$\ddot{x}(t) = -\gamma_1\, x(t) - \gamma_2 \dot{x}(t) + w(t). \tag{2.20}$$

The impulse response is shown in Figure 2.4, the meaning of the damping ratio ζ and natural frequency w_n will be discussed later. The MOU model is not commonly men-

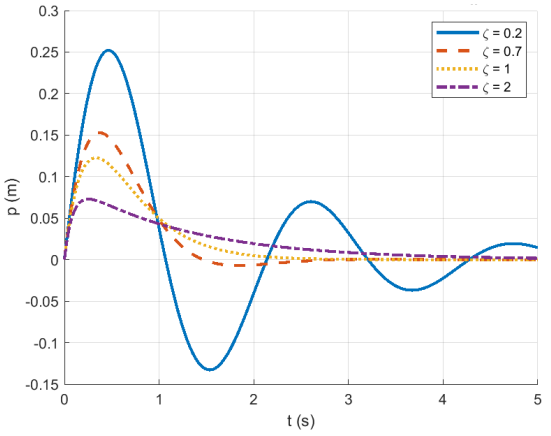


Figure 2.4: Position impulse response of MOU model.

tioned in the literature, as it often goes by other names

like the continuous auto-regressive model. Coraluppi published multiple papers on the MOU model for target tracking [11][12][13][14]. Two papers contain applications of the MOU model that align with our research question, which will be discussed next.

Applications

Coraluppi et al. [13] propose the MOU model for tracking a simulated target that follows deterministic sinusoidal motion in the z-direction and a MOU model in the x-direction. The MOU model outperforms the NCV model by a large margin, it has a lower position and velocity RMSE in both dimensions. It is important to note that the model parameters are assumed to be known in this simulation.

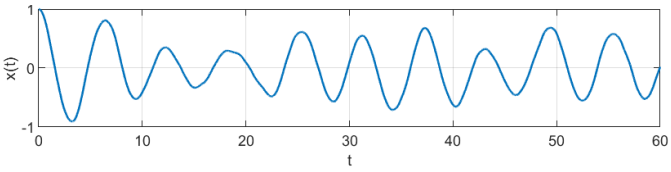
In a later paper, Coraluppi et al. [12] compare the NCV and MOU models in the presence of large coverage gaps. A simulation is created where a MOU filter exploits a weighted set of known trajectories for a coverage gap. Target reacquisition occurs with a much higher probability of correct track association. Another, more realistic, simulation is also created based on AIS historical data along the Australian coast. Also here the paper demonstrates the improvement of the multiple model MOU over the NCV model for a scenario where a target may follow multiple different nominal trajectories.

2.7 Dynamic Models Summary

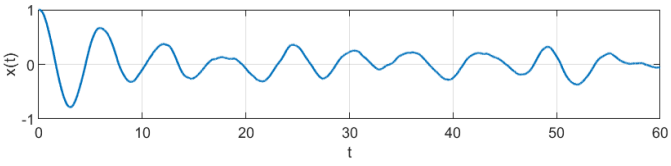
For state estimation, the NCV, Singer, iOU, and MOU models are all viable options. The most suitable model depends on the type of motion being considered. For processes with a nominal trajectory or weaving motion, the MOU model is the most appropriate because it is mean reverting in position. Moreover, the iOU model is a special case of the MOU model obtained by setting $\gamma_1 = 0$. For this reason, this thesis focuses on the MOU model. The Singer and NCV models are used as benchmarks for evaluating its performance. Before implementing the MOU model, its properties are examined in more detail.

2.8 MOU Model Properties

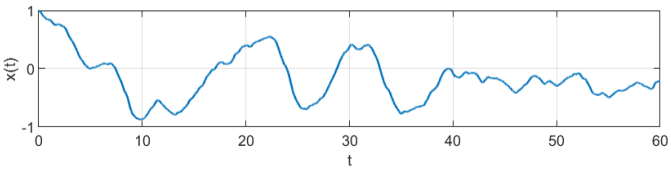
To illustrate the types of trajectories that can be described by an MOU model, Figure 2.5 shows three simulated trajectories generated with different parameter settings. Here, q denotes the process noise variance, and λ_1 and λ_2 are the eigenvalues, whose expressions are given later. The three simulations represent the following cases: a slowly decaying oscillatory case with $\lambda_{1,2} = -0.02 \pm j\,1$, a rapidly decaying oscillatory case with $\lambda_{1,2} = -0.1 \pm j\,1$, and a non-oscillatory case with $\lambda_1 = \lambda_2 = -0.5$. Damped oscillatory motion is not unique to the MOU model, but also appears in several related models. We therefore first discuss the connection between the MOU model and these models. Next, we present formulas for the eigenvalue decomposition of the MOU model, which are needed to compute the process noise covariance and to accelerate the evaluation of the transition function. Finally, we derive expressions for the autocovariance of the MOU model. These are required to initialize the prior distribution of the target state and are also needed when using the method of moments for parameter estimation.



(a) $\lambda_{1,2} = -0.02 \pm j\,1$



(b) $\lambda_{1,2} = -0.1 \pm j\,1$



(c) $\lambda_1 = \lambda_2 = -0.5$

Figure 2.5: Trajectories of the MOU model for different eigenvalues.

2.8.1 Equivalent models

The MOU model can be viewed as a second-order linear time-invariant system, or two-pole system, by rewriting Equation 2.20 to

$$X(s) = \frac{W(s)}{s^2 + \gamma_2 s + \gamma_1} = \frac{W(s)}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \tag{2.21}$$

With

$$w_n = \sqrt{\gamma_1}, \tag{2.22}$$

$$\zeta = \frac{\gamma_2}{2\sqrt{\gamma_1}}. \tag{2.23}$$

The damped frequency, so oscillation frequency observed when ζ is non-zero, is given by

$$w_d = \sqrt{\gamma_1 - \gamma_2^2/4}. \tag{2.24}$$

Without mean reversion in velocity, so $\gamma_2 = 0$, the system becomes undamped. The impulse response of this system is undamped sinusoidal motion. For the case where there is no reversion in position it becomes equivalent to the iOU model. For different damping ratios and $w_n = 3$ the impulse response is shown in Figure 2.4. w_n only affects the timescale.

CARMA model

The continuous-time counterpart of an autoregressive moving-average (ARMA) model is the CARMA model. Consider a CARMA(p, q) model of the form [6]

$$\begin{aligned}&(c_0 + c_1 D + c_2 D^2 + ... + c_p D^p)x \\ &= (b_0 + b_1 D + b_2 D^2 + ... + b_q D^q)w.\end{aligned} \tag{2.25}$$

Where D is the differential operator. The CARMA(2,0) model is given by

$$(c_0 + c_1 D + c_2 D^2)x = b_0 w \tag{2.26}$$

$$\Rightarrow c_2 \ddot{x} = -c_1 \dot{x} - c_0 x + b_0 w. \tag{2.27}$$

This model is equivalent to the MOU model up to a nonzero scaling of the coefficients. Dividing Equation 2.27 by c_2 gives

$$\ddot{x} = -\frac{c_1}{c_2}\dot{x} - \frac{c_0}{c_2}x + \frac{b_0}{c_2}w, \tag{2.28}$$

which has the same form as the MOU model given in Equation 2.20. Hence, the two models are equivalent after setting

$$\gamma_2 = \frac{c_1}{c_2}, \qquad \gamma_1 = \frac{c_0}{c_2},$$

with the noise term scaled accordingly. This means the MOU model is a CARMA(2,0) model, also known as a CAR(2) model as it does not have a moving-average component. This is important to note, because relatively little research exists on the MOU model, whereas much more research exists on CAR models.

For regularly spaced intervals a CAR(2) model equals an ARMA(2,1) model, but not every ARMA(2,1) model has a CAR(2) equivalent [3][8]. More information on the embeddability of a CARMA model in an ARMA model is given in Appendix A.6.

2.8.2 Eigenvalue decomposition

When the x- and y-dimensions are decoupled, MOU model for a single dimension can be written in state-space form as $\dot{\mathbf{s}} = \mathbf{F}\mathbf{s}$, with $\mathbf{s} = [x, v]^\top$ and

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ -\gamma_1 & -\gamma_2 \end{bmatrix}. \tag{2.29}$$

The eigenvalues of the state matrix $\mathbf{F}$ are obtained from the roots of the characteristic polynomial

$$\det(s\mathbf{I} - \mathbf{F}) = s^2 + \gamma_2 s + \gamma_1, \tag{2.30}$$

which are given by

$$\lambda_{1,2} \;=\; \frac{-\gamma_2 \pm \sqrt{\gamma_2^2 - 4\gamma_1}}{2}. \tag{2.31}$$

From this equation, it becomes apparent that for a stable process, i.e. one with only negative eigenvalues, both γ variables need to be positive.

Right eigenvectors

The right eigenvectors $\mathbf{v}$ are obtained by solving

$$(\mathbf{F} - s\mathbf{I})\mathbf{v} = 0.$$

Writing $\mathbf{v} = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^\top$ gives

$$\begin{bmatrix} -s & 1 \\ -\gamma_1 & -\gamma_2 - s \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{2.32}$$

which implies

$$-sv_1 + v_2 = 0, \qquad v_2 = sv_1. \tag{2.33}$$

Choosing $v_1 = 1$ yields the right eigenvector corresponding to the eigenvalue λ_i as

$$\mathbf{v}_i = \begin{bmatrix} 1 \\ \lambda_i \end{bmatrix}, \qquad i = 1, 2. \tag{2.34}$$

Collecting the right eigenvectors in a matrix gives

$$\mathbf{V} = \begin{bmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{bmatrix}. \tag{2.35}$$

Left eigenvectors

The left eigenvectors $\mathbf{w}^\top$ are obtained by solving

$$\mathbf{w}^\top(\mathbf{A} - s\mathbf{I}) = 0.$$

Writing $\mathbf{w}^\top = \begin{bmatrix} a & b \end{bmatrix}$ gives

$$\begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} -s & 1 \\ -\gamma_1 & -\gamma_2 - s \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \tag{2.36}$$

which implies

$$-as - b\gamma_1 = 0, \qquad a - b(\gamma_2 + s) = 0. \tag{2.37}$$

Choosing $b = 1$ gives $a = \gamma_2 + s$. Hence, for the eigenvalue λ_i, the corresponding left eigenvector is

$$\mathbf{w}_i^\top = \begin{bmatrix} \gamma_2 + \lambda_i & 1 \end{bmatrix}, \qquad i = 1, 2. \tag{2.38}$$

Collecting the left eigenvectors in a matrix yields

$$\mathbf{W} = \begin{bmatrix} \gamma_2 + \lambda_1 & 1 \\ \gamma_2 + \lambda_2 & 1 \end{bmatrix}. \tag{2.39}$$

2.8.3 Auto-covariance

Coraluppi et al. [12] give the steady-state covariance matrix of the state $[x, v]^\top$ as

$$\overline{\mathbf{Q}} = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}, \tag{2.40}$$

where the standard deviations of position and velocity are given by

$$\sigma_p = \frac{q}{2\gamma_1\gamma_2}, \qquad \sigma_v = \frac{q}{2\gamma_1}. \tag{2.41}$$

Here, q denotes the variance of the process noise. The autocorrelation of a general state-space model can be computed using the expression given by Brockwell [6]:

$$\mathbf{\Sigma}(\tau) \;=\; \mathbb{E}\big[\mathbf{s}(t+\tau)\,\mathbf{s}(t)^\top\big] \;=\; e^{\mathbf{F}|\tau|}\,\overline{\mathbf{Q}}. \tag{2.42}$$

Coraluppi et al. [13] provide an expression for $e^{\mathbf{F}\tau}$, which is verified by the derivations in Appendix A.3. For real-valued eigenvalues of $\mathbf{F}$, the matrix exponential $e^{\mathbf{F}\tau}$ is given by

$$e^{\mathbf{F}\tau} = \tfrac{1}{\lambda_1-\lambda_2}\begin{bmatrix} -\lambda_2 e^{-\lambda_1\tau} + \lambda_1 e^{-\lambda_2\tau} & -e^{-\lambda_1\tau} + e^{-\lambda_2\tau} \\ \lambda_1\lambda_2 e^{-\lambda_1\tau} - \lambda_1\lambda_2 e^{-\lambda_2\tau} & \lambda_1 e^{-\lambda_1\tau} - \lambda_2 e^{-\lambda_2\tau} \end{bmatrix}. \tag{2.43}$$

For the repeated eigenvalue case it is given by

$$e^{\mathbf{F}\tau} = \begin{bmatrix} (1+\lambda\tau)e^{1+\lambda\tau} & \tau e^{-\lambda\tau} \\ -\lambda^2 e^{-\lambda\tau} & (1-\lambda\tau)e^{-\lambda\tau} \end{bmatrix}. \tag{2.44}$$

For complex-valued eigenvalues ($\lambda_{1,2} = \alpha \pm i\beta$) it is given by

$$e^{\mathbf{F}\tau} = e^{-\alpha\tau}\begin{bmatrix} \cos(\beta\tau) + \dfrac{\alpha}{\beta}\sin(\beta\tau) & \dfrac{1}{\beta}\sin(\beta\tau) \\ -\dfrac{\alpha^2+\beta^2}{\beta}\sin(\beta\tau) & \cos(\beta\tau) - \dfrac{\alpha}{\beta}\sin(\beta\tau) \end{bmatrix}. \tag{2.45}$$

Chapter 3

2D MOU-Based Filter Implementation

This chapter describes how the MOU model is implemented for target tracking. In the 2D MOU model the state vector $\mathbf{s}$ is given by $\mathbf{s}(t) = [p_x(t),\, p_y(t),\, v_x(t),\, v_y(t)]^T$, with p the positions and v the velocities. The process noise is assumed to be i.i.d. with covariance matrix $\mathbf{Q}_c$. The state-space model is given by

$$\dot{\mathbf{s}}(t) \;=\; \mathbf{F}\mathbf{s}(t) \;+\; \mathbf{w}(t), \qquad \mathbb{E}[\mathbf{w}(t)\mathbf{w}(\tau)^\top] \;=\; \mathbf{Q}_c\,\delta(t-\tau). \tag{3.1}$$

For non-coupled dimensions, the matrices $\mathbf{F}$ and $\mathbf{Q}_c$ are given by

$$\mathbf{F} \;=\; \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\gamma_{1x} & 0 & -\gamma_{2x} & 0 \\ 0 & -\gamma_{1y} & 0 & -\gamma_{2y} \end{bmatrix}, \tag{3.2}$$

$$\mathbf{Q}_c \;=\; \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q_x & 0 \\ 0 & 0 & 0 & q_y \end{bmatrix}. \tag{3.3}$$

Up to this point, it has been assumed that both position and velocity revert to zero, which would correspond to a target remaining in a fixed location and is not realistic. A nominal trajectory may be learned from historical data but when such data are unavailable, it can also be estimated. To keep the dimensionality of the optimization problem small, the nominal trajectory may be approximated as a straight line. To represent this, as well as a potentially time-varying mean, the state vector is augmented by

$$\boldsymbol{\mu} = [\mu_x, \mu_y, \mu_{vx}, \mu_{vy}]^\top, \tag{3.4}$$

for which the transition and process noise covariance are given by

$$\mathbf{F}_\mu \;=\; \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.5}$$

$$\mathbf{Q}_\mu \;=\; \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q_{\mu_x} & 0 \\ 0 & 0 & 0 & q_{\mu_y} \end{bmatrix}. \tag{3.6}$$

The augmented $\mathbf{F}$ and $\mathbf{Q}_c$ matrices then become

$$\boldsymbol{F} = blkdiag([\boldsymbol{F}, \boldsymbol{F}_\mu]), \qquad \boldsymbol{Q}_c = blkdiag([\boldsymbol{Q}_c, \boldsymbol{Q}_\mu]). \tag{3.7}$$

The nominal trajectory is completely specified by the initial mean position and mean velocity. Next, the continuous model is converted to a discrete time state-space model, which is given by

$$\mathbf{s}_{k+1} \;=\; \mathbf{A}_k\,\mathbf{s}_k \;+\; \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0},\, \mathbf{Q}_k). \tag{3.8}$$

$\boldsymbol{A}_k$ is derived in section 3.2 and $\boldsymbol{Q}_k$ is derived in section 3.3. $\mathbf{s}$ is measured with the measurement function h, which is derived in section 3.4. The measurement noise $\boldsymbol{\nu}$ has covariance matrix $\mathbf{R}$, which is assumed to be time-independent and known. $\mathbf{z}_k$ can be described by

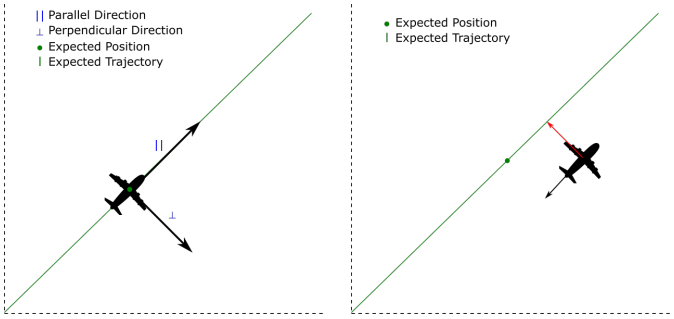
$$\mathbf{z}_k \;=\; h(\boldsymbol{s}_k) \;+\; \boldsymbol{\nu}_k, \qquad \boldsymbol{\nu}_k \sim \mathcal{N}(\mathbf{0},\, \mathbf{R}). \tag{3.9}$$

The true position and velocity of the target can be calculated by the summed position and summed velocity, given by $p_{sum} = p + p_\mu$ and $v_{sum} = v + v_\mu$.

3.1 Basis Selection

The motion dynamics in the parallel and perpendicular directions often differ. For this reason, target coordinates are often expressed in a basis aligned with these directions. For a target following a nominal trajectory, deviations perpendicular to the trajectory are typically corrected so that the target returns to the path. In contrast, a deviation in the parallel direction does not necessarily lead to a restoring effect. For example, if a target has been moving faster along its nominal trajectory than expected, this does not imply that it will later slow down to compensate, unless it is constrained to follow a strict schedule. Thus, position reversion may be present in the perpendicular direction but absent in the parallel direction. This distinction is illustrated in Figure 3.1a and Figure 3.1b. In Figure 3.1a, the target is located at its expected position on the nominal trajectory. In Figure 3.1b, the target has drifted to the right and is ahead of schedule. In this case, a restoring force, indicated in red, pulls the target back toward the nominal trajectory shown by the green line. However, no restoring force is required in the direction of travel, indicated by the black arrow. This corresponds to setting $\gamma_1 = 0$ in the parallel direction. A similar argument applies to weaving motion, where the oscillations occur only in the direction perpendicular to the nominal trajectory. These examples illustrate why a parallel-perpendicular basis is well suited for modeling the MOU process.

Note that the parallel and perpendicular directions here are defined relative to the heading of the nominal trajectory, not to the instantaneous heading of the target.



(a) (b)

Figure 3.1: The airplane in (a) is on track. The airplane in (b) has drifted off track, and should eventually drift back along the red arrow to resume the nominal trajectory.

To use a parallel-perpendicular basis, a state $\mathbf{s}_p = [x, y]^\top$ can be rotated to $\mathbf{p}_p = [\|, \perp]^\top$. Let the parallel direction be oriented at an angle ψ counterclockwise from the positive x-axis. When measurements are taken in the x–y plane, the position can be expressed in parallel–perpendicular coordinates by rotating the coordinate system ψ degrees clockwise. The counterclockwise rotation matrix, $\mathbf{R}(\psi)$, is given by

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}. \tag{3.10}$$

To rotate clockwise, the inverse of $\mathbf{R}(\psi)$ is used. Because a rotation matrix is unitary the inverse of $\mathbf{R}$ is equal to $\mathbf{R}^\top(\psi)$. The coordinate transformations are given by

$$\mathbf{p}_p = \mathbf{R}(\psi)^\top \mathbf{s}_p, \qquad \mathbf{s}_p = \mathbf{R}(\psi)\mathbf{p}_p. \tag{3.11}$$

The full state can be transformed using the block rotation matrix $\boldsymbol{\mathcal{R}}(\psi)$ as follows

$$\mathbf{s} = \boldsymbol{\mathcal{R}}(\psi)\mathbf{p}, \tag{3.12}$$

$$\mathbf{p} = \boldsymbol{\mathcal{R}}(\psi)^\top \mathbf{s}. \tag{3.13}$$

Where

$$\mathbf{s} = [p_x, p_y, v_x, v_y, \mu_{px}, \mu_{py}, \mu_{vx}, \mu_{vy}]^\top, \tag{3.14}$$

$$\mathbf{p} = [p_\|, p_\perp, v_\|, v_\perp, \mu_{p_\|}, \mu_{p_\perp}, \mu_{v_\|}, \mu_{v_\perp}]^\top, \tag{3.15}$$

$$\boldsymbol{\mathcal{R}}(\psi) = \text{blkdiag}\left(\mathbf{R}(\psi), \mathbf{R}(\psi), \mathbf{R}(\psi), \mathbf{R}(\psi)\right). \tag{3.16}$$

The matrix $\mathbf{F}$, defined in the $\|\perp$ frame, can then be transformed to the $x - y$ frame by

$$\mathbf{F}_{xy} = \boldsymbol{\mathcal{R}}\mathbf{F}\boldsymbol{\mathcal{R}}^\top. \tag{3.17}$$

Similarly, it is often more meaningful to model process noise in the parallel and perpendicular directions rather than in the $x - y$ frame. The process noise is transformed as follows:

$$\mathbf{Q}_{xy} = \boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{Q}}\boldsymbol{\mathcal{R}}^\top. \tag{3.18}$$

3.2 Transition Function

The transition matrix $\mathbf{A}$ is defined as $e^{\mathbf{F}\tau}$. Computing the matrix exponential is computationally expensive. Furthermore, the matrix exponential must be evaluated n times per Kalman filter run, where n is the number of time steps. It is more efficient to compute the matrix exponential via the eigenvalue decomposition

$$e^{\mathbf{F}t} = \mathbf{V}e^{\mathbf{D}\tau}\mathbf{V}^{-1}. \tag{3.19}$$

If $\mathbf{F}$ is constant, only $e^{\mathbf{D}dt}$ has to be calculated, which is computationally cheap since $\mathbf{D}$ is diagonal. By definition of the matrix exponential, an eigenvector of $\mathbf{F}$ is also an eigenvector of $\mathbf{A}$, with corresponding eigenvalue

$$r = e^{\lambda\tau}. \tag{3.20}$$

Where λ is an eigenvalue of $\mathbf{F}$ and r the corresponding eigenvalue of $\mathbf{A}$. The eigenvalues and eigenvectors can be obtained from the equations in subsection 2.8.2. As an alternative to evaluating the matrix exponential, the transition matrix can be computed using the eigenvalue-based expression proposed by Coraluppi et al. [13]. They derive transition functions for three cases: distinct real eigenvalues, repeated eigenvalues, and complex eigenvalues.

3.3 Process Noise

Gaussian process noise allows for the use of a KF. Non-Gaussian distributions can be estimated by a particle filter, but this filter is computationally more expensive and more complex than a KF. Gan et al. [19] filtered the state of a target with a particle filter, using a Singer model. They inspected the effect of modeling Levy noise instead of Gaussian noise. Levy noise offers a flexible formulation that encompasses a wide range of possible system noise models. The result was that heavy-tailed Levy noise performed better than Gaussian noise, but not by a large margin even though the trajectory has many sharp (90°) turns which should favor the Levy noise due to its heavy-tailed nature. The Gaussian noise and Levy noise achieved a RMSE of 95.25 and 91.71 respectively. This result indicates that modeling heavy-tailed noise is not essential. Therefore, we assume the process noise to be Gaussian to simplify state estimation.

In the discrete time MOU process the covariance of the discrete process noise $\mathbf{w}_k$ depends on the interval time, continuous process noise $\mathbf{Q}_c$ and the transition matrix $\mathbf{F}$. It is given by

$$\mathbf{w}_k = \int_{t_{k-1}}^{t_k} e^{\mathbf{F}(t_k - t)}\, \mathbf{w}(t)\, dt, \qquad \mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^\top]. \tag{3.21}$$

Here $\mathbf{Q}_k$ is the covariance of the discrete process noise. The intuition behind the formula is as follows: $\mathbf{w}(t)$ enters the state at time t. The transition matrix for the state from time t to t_k is $e^{\mathbf{F}(t_k - t)}$, so $\mathbf{w}(t)$ decays to $e^{\mathbf{F}(t_k - t)}\boldsymbol{w}_t$ at time t_k. The integral of the contributions all continuous process noise in the interval results in $\mathbf{w}_k$.

Because $\mathbf{w}(t)$ is assumed to be time independent, the integration interval can be shifted from $[t_{k-1},\, t_k]$ to $[0,\, \Delta t_k]$ without altering the distribution of $\mathbf{w}_k$. $\mathbf{Q}_k$ is then given by

$$\begin{aligned}\mathbf{Q}_k &= \mathbb{E}\left[\int_0^{\tau_k} e^{\mathbf{F}t}\,\mathbf{w}(\tau_k - t)\,\mathbf{w}(\tau_k - t)^\top\, e^{\mathbf{F}^\top t}\,dt\right] \\ &= \int_0^{\tau_k} e^{\mathbf{F}t}\,\mathbb{E}\big[\mathbf{w}(\tau_k - t)\,\mathbf{w}(\tau_k - t)^\top\big]\, e^{\mathbf{F}^\top t}\,dt \\ &= \int_0^{\tau_k} e^{\mathbf{F}t}\,\mathbf{Q}_c\, e^{\mathbf{F}^\top t}\,dt.\end{aligned} \tag{3.22}$$

This formula does not have an easy solution, but Coraluppi has derived formulas for $\mathbf{Q}_k$ for the MOU model. Also, van Loan derived a method to solve general integrals of this form.

3.3.1 Coraluppi et al.

Coraluppi et al. [13] give a matrix for the process noise covariance, but the equation for $\mathbf{Q}_{22}$ seems to have a mistake, the "-2*2" in the original should be replaced with "-2". After fixing this, the process noise covariance functions are given by

$$\begin{aligned}\mathbf{Q}_{11} &= \frac{q}{(\lambda_{i1}-\lambda_{i2})^2}\left[\frac{1-e^{-2\lambda_{i1}\Delta t}}{2\lambda_{i1}} + \frac{1-e^{-2\lambda_{i2}\Delta t}}{2\lambda_{i2}} - 2\,\frac{1-e^{-(\lambda_{i1}+\lambda_{i2})\Delta t}}{\lambda_{i1}+\lambda_{i2}}\right], \\ \mathbf{Q}_{12} &= \frac{q}{2(\lambda_{i1}-\lambda_{i2})^2}\left[e^{-2\lambda_{i1}\Delta t} + e^{-2\lambda_{i2}\Delta t} - 2e^{-(\lambda_{i1}+\lambda_{i2})\Delta t}\right], \\ \mathbf{Q}_{22} &= \frac{q}{(\lambda_{i1}-\lambda_{i2})^2}\left[\frac{\lambda_{i1}}{2}\left(1-e^{-2\lambda_{i1}\Delta t}\right) + \frac{\lambda_{i2}}{2}\left(1-e^{-2\lambda_{i2}\Delta t}\right) - 2\,\frac{\lambda_{i1}\lambda_{i2}}{\lambda_{i1}+\lambda_{i2}}\left(1-e^{-(\lambda_{i1}+\lambda_{i2})\Delta t}\right)\right].\end{aligned} \tag{3.23}$$

Where q denotes the continuous process noise variance. Note that in the paper they use the negative of the eigenvalue so $-\lambda$ instead of λ as input in the formula, which is why their formula is slightly different. After fixing the mistake in $\mathbf{Q}_{22}$ the formulas give an equal output to the alternative van Loan decomposition. They also give the same result as the van Loan decomposition for unstable processes, so γ values smaller than 0. For some combinations of λ_{i1} and λ_{i2} the denominator equals zero, but the limit is still finite. Coraluppi et al. give an extra formula for repeated eigenvalues that gives the solution to this limit.

3.3.2 Van Loan

The method of Van Loan [39] is computationally more expensive than using Equation 3.23. But if there is coupled noise, so the noise on x and y are dependent, the Coraluppi formulas are not correct anymore. In that case the Van Loan decomposition must be used.

The covariance matrix of the noise can be described by Equation 3.22. Van Loan showed that the relation between equations 3.24 and 3.25 holds, which can be used to calculate the integral exactly.

$$\exp\left(\begin{bmatrix}-\mathbf{F} & \mathbf{Q}_c \\ 0 & \mathbf{F}^\top\end{bmatrix}\tau\right) = \begin{bmatrix}\mathbf{F}_2(\tau) & \mathbf{G}_2(\tau) \\ 0 & \mathbf{F}_3(\tau)\end{bmatrix}. \tag{3.24}$$

$$\mathbf{Q}_k = \int_0^{\tau_k} e^{\mathbf{F}t}\,\mathbf{Q}_c\, e^{\mathbf{F}^\top t}\,dt = \mathbf{F}_3(\tau_k)^\top \mathbf{G}_2(\tau_k). \tag{3.25}$$

The integral of Equation 3.25 is exactly the integral we need to calculate in Equation 3.22. First Equation 3.24 is calculated. Then $\mathbf{Q}_k$ is obtained by Equation 3.25.

3.4 Measurement Function

A radar does not work in x, y coordinates. Instead, position is measured in polar coordinates, i.e. range and angle:

$$r = \sqrt{p_x^2 + p_y^2}, \tag{3.26}$$

$$\theta = \operatorname{atan2}(p_y, p_x). \tag{3.27}$$

It is important to note that the measurement function uses the true positions as input for p_x and p_y, which are equal to $p_x + \mu_x$ and $p_y + \mu_y$ respectively if the state is augmented. Also radial velocity is measured, $v_r = \frac{\partial r}{\partial t}$ where r is the radius. The radial velocity is given by

$$\frac{p_x v_x + p_y v_y}{\sqrt{p_x^2 + p_y^2}}. \tag{3.28}$$

Also here the true velocity is used as input. The complete measurement function is equal to

$$h(\mathbf{s}) = \begin{bmatrix}\sqrt{p_x^2 + p_y^2} \\ \operatorname{atan2}(p_y, p_x) \\ \frac{p_x v_x + p_y v_y}{\sqrt{p_x^2 + p_y^2}}\end{bmatrix}. \tag{3.29}$$

Because the measurement noise is in polar coordinates its uncertainty does not describe an ellipse in $x-y$ coordinates but can take on different forms, as shown in Figure 3.2.

3.5 Kalman Filter

Because the measurement function is nonlinear, the standard linear Kalman-filter cannot be used anymore. However, there exist a range of nonlinear Kalman filters. To decide which type of Kalman filter should be used, the level of nonlinearities has to be evaluated. The transition function is linear and the process noise is additive. Given that polar coordinates and radial velocity are measured, the observation matrix is nonlinear. As long as the target remains sufficiently far from the radar, the derivatives in radial velocity and polar coordinates vary slowly and

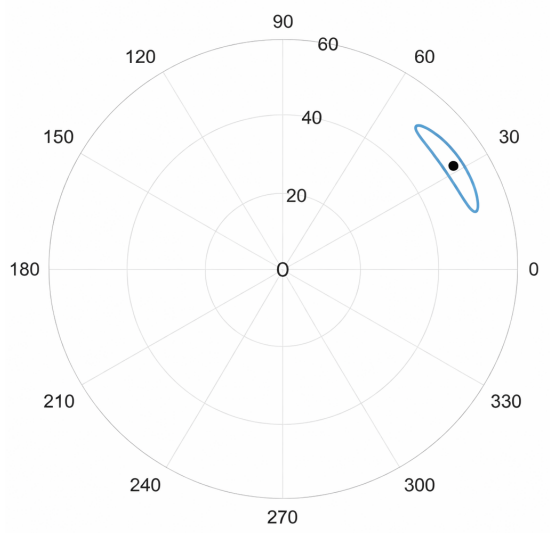


Figure 3.2: Example of measurement 95% confidence interval of target position in r and θ.

an EKF should suffice. However, the measurement covariance will not be accurately modeled by an EKF in case the confidence interval takes on forms like the one shown in Figure 3.2. A UKF will therefore be more accurate. However, the UKF sigma points will no longer work due to the circular angle measurements. The angle measurements are represented in radians in the interval $[-\pi, \pi]$. The Kalman filter does not inherently understand that the angle is circular, and -0.9π and 0.9π are actually 0.2π away from each other instead of 1.8π. For this reason it is important to make a new error function that takes this into account. This works for the EKF, but not for the UKF. The UKF estimates the mean by propagating sigma points through the nonlinear function and computing their weighted average. If two sigma points are propagated and one lands at $-\pi$ and one at π, the mean of the propagated sigma points are estimated as 0, even though the true circular mean should be π or $-\pi$. For this reason the polar measurements and their covariance will have to be converted to the Cartesian x, y values. The drawback is that the actual shape of the covariance, as shown in Figure 3.2, will not be captured exactly when using x, y values. A potential solution probably exists within the UKF framework, though implementing such a method is beyond the scope of this work.

Chapter 4

MOU Model Parameter Estimation

The set of parameters that needs to be optimized is called $\boldsymbol{\theta}$. This set contains:

- $\gamma_{1\parallel}$, $\gamma_{1\perp}$ Position reversion in the parallel and perpendicular direction.
- $\gamma_{2\parallel}$, $\gamma_{2\perp}$ Velocity reversion in the parallel and perpendicular direction.
- $\sigma_{\parallel}$, $\sigma_{\perp}$ Process noise standard deviation in the parallel and perpendicular direction.

A general linear Gaussian continuous-time model is given by

$$\mathbf{s}_{k+1} = e^{\mathbf{F}\tau_k}\mathbf{s}_k + \mathbf{L}\mathbf{w}_k \tag{4.1}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{s}_k + \mathbf{v}_k \tag{4.2}$$

Where $\mathbf{w}_k$ and $\mathbf{v}_k$ contain independent Gaussian distributed noise, $\mathbf{L}$ is the process noise shaping matrix that transforms w_k into state disturbances, and $\mathbf{H}$ is the measurement matrix. The process noise covariance is given by $\mathbf{Q}_c = \mathbf{L}\mathbf{L}^\top$. Christensen et al. [10] show when such a model is identifiable. The three main identifiability issues in state-space models will be discussed. Firstly, in regular time intervals aliasing may happen. This happens when the phase of the $\mathbf{F}$ matrix eigenvalues differ by an integer multiple of $2\pi/\tau$ for a sampling time of τ. In practice, the sampling time will be smaller than the period, so this is not a problem. Furthermore, aliasing will not occur in the case of irregular time intervals. Secondly, when the measurement matrix is unknown, there is similarity-transform invariance. In that case multiple models can produce identical measurements, despite underlying model differences. Because the measurement matrix is known, this similarity-transform is not possible. Lastly, $\mathbf{L}\mathbf{U}$ will result in the same likelihood for any unitary $\mathbf{U}$ and only the covariance, $\mathbf{Q}_c = \mathbf{L}\mathbf{L}^\top$, is identified. But this is not a problem as the exact process noise dynamics are not important, only the effect of the noise dynamics on the state are of interest. Optimizing all entries in $\mathbf{Q}_c$ introduces redundant degrees of freedom and an unnecessarily large parameter space. The Cholesky parameterization, given by $\mathbf{L}$, restricts the search to the minimal set of independent parameters, producing a simpler feasible set and a smoother optimization landscape.

Multiple parameterizations exist for CAR models, so also for the MOU model which is a CAR(2) model. Lebreton and Jones [28] use a parameterization where the AR polynomial is factored into second-order polynomials and the roots of these polynomials are fitted. A more detailed explanation can be found in Appendix A.4.1. They state that this approach is preferred because the feasible region becomes complex for higher-order CAR models. However, in the CAR(2) model the feasible region is simply $\gamma_1 > 0, \gamma_2 > 0$. Only for order > 2 the feasible region becomes complex. Another reason root fitting is useful is because it is easy to calculate the transition matrix if the eigenvalues are known. In that case the transition matrix $e^{\mathbf{F}\tau}$ can be calculated in a computationally cheap way by using the Vandermonde state transformation, which is further explained in Appendix section A.5. For the CAR(2) model this brings no benefit as the eigenvalue decomposition is readily computable which allows for a computationally cheap way to calculate the transition matrix as shown in section 3.2.

Belcher [2] found that fitting the roots of a CAR model encounters converging problems. Instead, Belcher proposes to fit the roots of a discrete ARMA model. This is done by iteratively updating the ARMA roots until the theoretical ARMA spectrum corresponding to the roots fit the observed spectrum of the data as close as possible. The exact way to do this is set out in Appendix subsection A.4.2. The sampling rate can be irregular even though discrete ARMA roots are fitted. The rationale for fitting ARMA roots instead of CARMA roots is that the AR spectrum is well-known to have excellent approximation power on any continuous-valued function on $[0, \pi]$. Then the ARMA roots can be converted to CARMA roots.

To summarize, no parameterization is necessary for the MOU model because the feasible region is simple and the transition matrix can already be calculated in a computationally cheap way. Belcher's approach of fitting discrete ARMA roots may offer better convergence properties. However, due to the added complexity and the uncertainty regarding its benefits for the MOU model, this approach will not be used.

4.1 Cramer-Rao Lower Bound

The Cramér–Rao Lower Bound (CRLB) provides a lower bound on the variance of a parameter estimator. This bound applies when the regularity condition is satisfied,

which is given by

$$\int \frac{\partial p(\mathbf{z};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\, d\mathbf{z} = \frac{\partial}{\partial \boldsymbol{\theta}} \int p(\mathbf{z};\boldsymbol{\theta})\, d\mathbf{z}. \tag{4.3}$$

Here $\mathbf{z}$ denotes the measurement, $\boldsymbol{\theta}$ the parameters and $p(\cdot)$ the likelihood. The likelihood is computed using the Kalman filter's innovation sequence, which is calculated by

$$p(\mathbf{z},\boldsymbol{\theta}) = \prod_{k=1}^{N} \frac{1}{(2\pi)^{n/2}\,|S_k|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{v}_k^\top S_k^{-1}\mathbf{v}_k\right). \tag{4.4}$$

Here $\mathbf{v}_k = \mathbf{z}_k - \mathbf{H}_k\,\mathbf{s}_{k|k-1}$, $\mathbf{S}$ is the innovation covariance and n is the number of elements in $\mathbf{v}$. For the regularity condition to hold, the order of differentiation and integration must not matter and the likelihood must be differentiable. The likelihood is a product of Gaussian terms, which satisfies the regularity condition. This is because the support of a Gaussian distribution is not changed by the parameters, and the likelihood is differentiable. The lower bound can be calculated by

$$\begin{aligned}\operatorname{var}(\hat{\boldsymbol{\theta}}) \;\geq\; \frac{1}{-\mathbb{E}\left[\dfrac{\partial^2}{\partial \boldsymbol{\theta}^2}\ln p(\mathbf{x};\boldsymbol{\theta})\right]} \;&=\; \frac{1}{\mathbb{E}\left[\left(\dfrac{\partial}{\partial \boldsymbol{\theta}}\ln p(\mathbf{x};\boldsymbol{\theta})\right)^2\right]} \\ &= \frac{1}{\mathbf{I}(\boldsymbol{\theta})}.\end{aligned} \tag{4.5}$$

In case the likelihood of an observation matrix $V = [\mathbf{v_1}, \mathbf{v_2}...\mathbf{v}_K]$ consists of independent identically distributed vectors $\mathbf{v}$ drawn from a Gaussian distribution, the Slepian-Bangs formula provides a general formula for the Fischer Information Criterion (FIM) as shown in [22] pp. 142-143. The Slepian-Bangs formula is given by

$$I_{i,j}(\boldsymbol{\theta}) = \sum_{k=1}^{N} \frac{\partial \mu_k^T}{\partial \theta_i}\,\boldsymbol{\Sigma}^{-1}\,\frac{\partial \mu_k}{\partial \boldsymbol{\Sigma}_j} + 0.5\,\mathrm{Tr}\{\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\Sigma}_{k,i}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\Sigma}_{k,j}\}. \tag{4.6}$$

Where in our case $\mu_k = E[v_k]$, $\boldsymbol{\Sigma}_k = \mathbf{S}_k$ and $\boldsymbol{\Sigma}_{k,j} = \frac{\partial \boldsymbol{\Sigma}_k}{\partial \theta_j}$. Without an offset in the measurement, $\mu_k = 0$ because the expected value of the innovation residual is zero. But if there is an offset, take for example $\mathbf{z}_k = \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} + \boldsymbol{\mu}$, this term becomes non-zero and the FIM can be non-zero for parameters that affect $\boldsymbol{\mu}$. We can simplify Equation 4.6 to

$$I_{i,j}(\boldsymbol{\theta}) = \sum_{k=1}^{N} 0.5\,\mathrm{Tr}\{\mathbf{S}_k^{-1}\mathbf{S}_{k,i}\mathbf{S}_k^{-1}\mathbf{S}_{k,j}\}. \tag{4.7}$$

$\frac{\partial \mathbf{S}_k}{\partial \theta_j}$ can be calculated using equations 4.8 and 4.9. This result is also derived by Larsson et al. [27] who use this formula for the CRLB of continuous AR models with irregular intervals but without measurement noise. Cavanaugh and Shumway [7] use this formula to calculate the CRLB for state-space models with an unknown $\mathbf{H}$ and $\mathbf{R}$ matrix and with regular time intervals.

The formulas by Harvey [22][pp 142-143] are used to derive the partial derivative formulas. Here the derivative of the $\mathbf{S}$ matrix is given by

$$\partial_\theta \mathbf{S}_k = \mathbf{H}_k\,(\partial_\theta \mathbf{P}_{k|k-1})\,\mathbf{H}_k^\top + \partial_\theta \mathbf{R}_k. \tag{4.8}$$

Where $\partial_\theta \mathbf{P}_{k|k-1}$ is given by

$$\begin{aligned}\partial_\theta \mathbf{P}_{k|k-1} = (\partial_\theta \mathbf{A}_k)\,\mathbf{P}_{k-1|k-1}\,\mathbf{A}_k^\top + \mathbf{A}_k\,(\partial_\theta \mathbf{P}_{k-1|k-1})\,\mathbf{A}_k^\top \\ + \mathbf{A}_k\,\mathbf{P}_{k-1|k-1}\,(\partial_\theta \mathbf{A}_k)^\top + \partial_\theta \mathbf{Q}_k.\end{aligned} \tag{4.9}$$

To calculate $\partial_\theta \mathbf{A}_k$, $\partial_\theta \mathbf{P}_{k-1|k-1}$ and $\partial_\theta \mathbf{Q}_k$ difficult Fréchet derivatives are needed. To simplify the computation, $\partial_\theta \mathbf{S}_k$ is obtained numerically by the finite difference method. This is done by running a KF which returns the $\mathbf{S}$ matrices. In a KF, only the state estimate is updated by the measurement and not the innovation covariance, so normally it does not matter what measurements are given as input when the CRLB is calculated. However, for radar the measurement noise depends on the position of the target. The position measurement will be set equal to the real x and y position at every step so the measurement covariance matrix is accurate. The CRLB depends on the following variables:

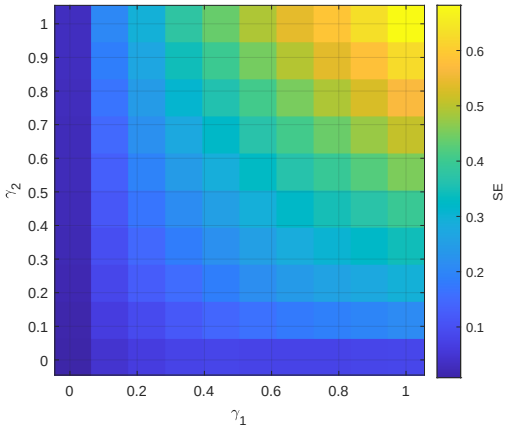
- $\boldsymbol{\gamma_{1x}, \gamma_{2x}, \gamma_{1y}, \gamma_{2y}}$, decay parameters in the state transition model.
- $\boldsymbol{q_x, q_y}$, the process noise variances.
- $\mathbf{t} = [t_1, t_2, ..., t_n]^\top$, measurement time instants.
- $\boldsymbol{p_x, p_y}$, target positions for each measurement time
- $\mathbf{R}$, measurement noise covariance in polar coordinates (range, angle, radial velocity).
- $\mathbf{P_0}$, the initial state covariance, defined in Equation 2.41.
- $\boldsymbol{P_{\mu 0}}$, initial covariance on the mean state $\boldsymbol{\mu} = [\mu_x, \mu_y, \mu_{vx}, \mu_{vy}]^\top$.

The CRLB is computed with respect to the γ_1 and γ_2 parameters by Equation 4.7 with the settings defined in Table 4.1. Figure 4.1 shows the effect of different parameter values on the CRLB.

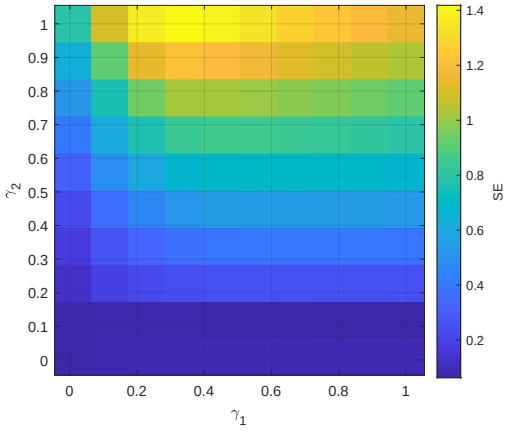
Table 4.1: CRLB settings.

Parameter	Value
q_x, q_y	$5\,(\frac{m}{s})^2$
t	$[0, 2, 4, ..., 200]\,s$
p_x, p_y	$2000\,m, 2000\,m$
R	$[15^2\,m^2, 5^2\,mrad^2, 5^2\,(\frac{m}{s})^2]$
$P_{\mu 0}$	$[3000\,m, 3000\,m, 10\,(\frac{m}{s})^2, 10\,(\frac{m}{s})^2]$

It can be observed that higher values of γ_1 or γ_2 lead to decreased accuracy in parameter estimation. This behavior can partly be attributed to faster dynamics for higher γ_1 or γ_2 values, leading to faster signal decay and reduced information.



(a) Lower bound for γ_1 standard error.



(b) Lower bound for γ_2 standard error.

Figure 4.1: CRLB parameter SE as a function of γ_1 and γ_2.

4.2 Offline Estimation Methods

Accuracy and computational efficiency are both important in the estimation process. None of the offline estimation methods that will be covered in this section have a closed-form solution. Instead, a cost function is iteratively optimized by adjusting the parameters, so the cost function needs to be evaluated many times. If the iterations are computationally expensive, the estimation is slow. The estimation methods covered here are originally for CARMA models, but also work for the MOU model which is a specific type of CARMA model.

4.2.1 Method of moments

One approach to estimating CARMA models is the method of moments. In this approach, the parameters are estimated by matching the theoretical and observed covariances as closely as possible. Bergstrom [3] gives an approach for regular time intervals. The method iteratively minimizes

$$L(\boldsymbol{\theta}) = \log|\mathbf{V}(\boldsymbol{\theta})| + \mathbf{z}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{z}. \tag{4.10}$$

Here, $\mathbf{z}$ represents the observation vector, and $\mathbf{V}(\boldsymbol{\theta})$ denotes the $N \times N$ covariance matrix as a function of $\boldsymbol{\theta}$, where N is the number of samples. This theoretical covariance is obtained by rewriting the CAR(2) model as an MA(∞) model. The sample covariance is then computed from the truncated MA component series. When the process noise is multivariate Gaussian, the resulting estimator is the maximum likelihood estimator [23]. However, when this approach is used for irregular sampling intervals, the covariance matrix no longer retains its block structure which is given by

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_0 & \mathbf{V}_1 & \mathbf{V}_2 & \cdots & \mathbf{V}_T \\ \mathbf{V}_1' & \mathbf{V}_0 & \mathbf{V}_1 & \cdots & \mathbf{V}_{T-1} \\ \mathbf{V}_2' & \mathbf{V}_1' & \mathbf{V}_0 & \cdots & \mathbf{V}_{T-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{V}_T' & \mathbf{V}_{T-1}' & \mathbf{V}_{T-2}' & \cdots & \mathbf{V}_0 \end{bmatrix}. \tag{4.11}$$

This means that the theoretical covariance has to be calculated for every pair of measurements which increases computation time. Furthermore, $\mathbf{V}$ has to be inverted. This inversion can be faster than the inversion of a general matrix because a Cholesky decomposition is possible, but it remains computationally expensive, especially with a large number of samples. A way to solve this is to set the autocovariance to zero beyond a certain lag to create a banded matrix, because $e^{\mathbf{A}\tau}$ tends to zero for large τ. But for low γ values the correlation remains significant over long lags.

4.2.2 Expectation-maximization

Another approach to parameter estimation is expectation-maximization (EM). Ghahramani and Hinton [20] present the E and M steps for a linear stochastic differential equation with regular time intervals. In the E-step the Kalman filter with the Rauch-Tung-Striebel smoother is used. The smoother performs backward recursions so future measurements are used to make state estimation more accurate. In the M-step, the derivative of the log-likelihood is set to zero with respect to the parameters. Given the states $\mathbf{s}_n$ and measurements $\mathbf{z}_n$ the log-likelihood is given by

$$\begin{aligned} \log P(\{\mathbf{s}\},\{\mathbf{z}\}) = &-\sum_{k=1}^{N} \tfrac{1}{2}(\mathbf{z}_k - \mathbf{H}\mathbf{s}_k)'\mathbf{R}^{-1}(\mathbf{z}_k - \mathbf{H}\mathbf{s}_k) \\ &-\sum_{k=2}^{N} \tfrac{1}{2}(\mathbf{s}_k - \mathbf{A}_{k-1}\mathbf{s}_{k-1})'\mathbf{Q}_k^{-1}(\mathbf{s}_k - \mathbf{A}_{k-1}\mathbf{s}_{k-1}) \\ &-\tfrac{N}{2}\log|\mathbf{R}| - \sum_{k=2}^{N}\tfrac{1}{2}\log|\mathbf{Q}_k| \\ &-\tfrac{1}{2}(\mathbf{s}_1 - \boldsymbol{\pi}_1)'\mathbf{V}_1^{-1}(\mathbf{s}_1 - \boldsymbol{\pi}_1) \\ &-\tfrac{1}{2}\log|\mathbf{V}_1| - \tfrac{N(n_p+n_s)}{2}\log(2\pi). \end{aligned} \tag{4.12}$$

Where $\mathbf{V}$ denotes the prior state covariance, $\boldsymbol{\pi}_1$ the mean initial state, n_p the number of parameters and n_s the number of state variables. The derivatives of $\mathbf{H}$ and $\mathbf{R}$ with respect to the parameters are zero, since these matrices are known. For regular time intervals, the M-step has a closed-form solution. But for irregular time intervals the transition matrix $\mathbf{A}$ is not constant. Consider the following part of the log-likelihood:

$$\ell(\boldsymbol{\theta}) = \sum_{k=2}^{N} \Bigg[-\frac{1}{2}\big[\mathbf{s}_k - \mathbf{A}_{k-1}\mathbf{s}_{k-1}\big]^\top \mathbf{Q}_k^{-1}\big[\mathbf{s}_k - \mathbf{A}_{k-1}\mathbf{s}_{k-1}\big] - \frac{1}{2} log|\mathbf{Q}_k| \Bigg]. \tag{4.13}$$

Define $\mathbf{e}_k = \mathbf{s}_k - \mathbf{A}_{k-1}\mathbf{s}_{k-1}$ to obtain

$$\frac{\partial \ell}{\partial \theta_i} = \frac{1}{2}\sum_{k=2}^{N} \Bigg[-\frac{\partial \mathbf{e}_k^\top}{\partial \theta_i}\mathbf{Q}_k^{-1}\mathbf{e}_k - \mathbf{e}_k^\top \mathbf{Q}_k^{-1}\frac{\partial \mathbf{e}_k}{\partial \theta_i} - \mathbf{e}_k^\top \frac{\partial \mathbf{Q}_k^{-1}}{\partial \theta_i}\mathbf{e}_k - \frac{\partial}{\partial \theta_i} log(|\mathbf{Q}_k|) \Bigg]. \tag{4.14}$$

Since $\mathbf{s}_k$ are fixed during the M-step:

$$\frac{\partial \mathbf{e}_k}{\partial \theta_i} = -\frac{\partial \mathbf{A}_k}{\partial \theta_i}\mathbf{s}_{k-1}. \tag{4.15}$$

The derivative of $\mathbf{A}$ with respect to the γ parameters is a derivative of the matrix exponential, which is nonlinear:

$$\frac{\partial \mathbf{A}_{k-1}}{\partial \theta_i} = \frac{\partial e^{\mathbf{F}\tau_{k-1}}}{\partial \theta_i} \tag{4.16}$$

The derivative of $\mathbf{Q}$ with respect to the γ parameters is also a nonlinear function. There is therefore no closed form solution to calculate the γ parameters. Because of this, generalized EM has to be used. In generalized EM, the M-step is done using numerical optimization. This makes each EM iteration computationally expensive.

4.2.3 Maximum likelihood

Another approach is to estimate parameters by maximizing the likelihood function. Every iteration a new parameter estimate is calculated based on gradient-descent, line-search or another optimization algorithm. the difference with using EM is that it does not use a smoother in its state estimation. It also does not optimize the parameters for a given state estimate so there is no M-step.

Harvey and Stock [23] show how to do maximium likelihood estimation (MLE) by maximizing the likelihood returned by the Kalman filter. They state that this approach is computationally cheaper than the approach by Bergstrom shown in subsection 4.2.1. The approach they build on is the approach by Jones [24]. What Harvey and Stock add is that the initial conditions are marginalized out if the process is non-stationary. The mean to which the MOU model reverts has to be estimated along the other parameters $\boldsymbol{\theta}$. High-dimensional MLE can be computationally intensive and could terminate at local maxima or saddle points [35]. A lower dimensional alternative to using MLE for $\boldsymbol{\mu}$ is to augment the state-vector to include $\boldsymbol{\mu}$ as is done in Equation 3.4. There is however a downside to this approach, the estimation is only equal to MLE asymptotically, and differs with a limited sample size as shown by Qian [35]. There are two exception to this. The first is when $p(\boldsymbol{\mu}|\mathbf{Z}, \theta)$ does not depend on θ, where $\boldsymbol{Z}$ contains the measurements. The second is when the augmented state-space Kalman filter output mean and covariance of $\boldsymbol{\mu}$ does not depend on θ, in this case the $\boldsymbol{\mu}$ estimation is the same, but not the θ estimation. In our case both these exceptions do not hold. For this reason MLE is more accurate than state-augmentation in optimizing the likelihood. Still, the state augmentation is used because it is easier to add a drift to the mean. But in case the estimation accuracy of $\boldsymbol{\mu}$ turns out to be a bottleneck, the ML $\boldsymbol{\mu}$ estimation should be implemented.

In maximum likelihood estimation it is possible to use numerical or analytic optimization. Koopman and Shephard [26] conclude that doing analytic optimization for a discrete state-space model may be slower than numerically differentiating the likelihood. For this reason numerical optimization is used. However, they also state that the optimization of process noise $\mathbf{Q}$ and measurement noise $\mathbf{R}$ parameters may be sped up significantly when using analytic optimization. Because $\mathbf{R}$ is known, and $\mathbf{Q}$ only has 2 unknown parameters it is not worth it to use a hybrid solution for complexity reasons, but this may be looked at if optimization time turns out to be a limiting constraint.

4.2.4 ARMA fitting

For regular-interval sampling it is possible to fit an ARMA model and convert it to a CARMA model. This method is used for sunspots by Phadke and Wu [34]. They state that if the continuous process is observed at a uniform sampling interval, τ, the resulting discrete process is ARMA (p, p-1). Furthermore, if τ is small enough so that the highest angular frequency corresponding to the imaginary part of the eigenvalues is less than $\frac{\pi}{\tau}$, then the correspondence between the parameters of the discrete model and the parameters of the continuous model is unique. They use MLE to estimate the ARMA parameters. The embedding of an ARMA(2,1) process into a CARMA(2,1) process is given by Thornton [38]. The conversion between ARMA(2,1) roots (r_1, r_2) and CARMA(2,1) roots (α_1, α_2) is given by

$$e^{\alpha_1} = r_1, e^{\alpha_2} = r_2, \tag{4.17}$$

$$\alpha_1 = log(r_1), \alpha_2 = log(r_2). \tag{4.18}$$

Thornton states that the question of embeddability is equivalent to asking under what circumstances will the CARMA process be real. So if the converted parameters from the ARMA to CARMA process are real the process is embeddable. For pure AR models, fast estimators such as OLS or the Yule–Walker equations are available. However, the presence of a MA component necessitates

maximum-likelihood estimation. Furthermore, ARMA estimation methods usually assume noise-free observations. So when measurement noise is present, the appropriate approach is to use a Kalman filter on the state-space form and use MLE. At that stage, this is effectively equivalent to estimating the CARMA parameters directly by MLE. One workaround is to fit an ARMA(2,2) model instead. An ARMA(2,1) process with measurement noise can be interpreted as an ARMA(2,2) process without measurement noise for which the proof is given in Appendix A.1. Nonetheless, this approximation is generally less accurate than correctly specifying the model to account for measurement noise. In conclusion, due to the measurement noise in the data, estimating ARMA parameters with measurement noise is as computationally expensive and effectively the same as estimating CARMA parameters directly.

4.2.5 Offline estimation conclusion

The method of moments is computationally expensive, particularly for large sample sizes and irregular time intervals. Expectation-maximization is also costly because the M-step has no closed-form solution. ARMA fitting is unlikely to achieve optimal accuracy, as it cannot account for measurement noise. Because of these reasons maximum likelihood estimation appears to be the most suitable option as its iterations are relatively inexpensive and the method is flexible: it works for general state-space models.

4.3 Online Estimation Methods

Chebbi et al. review three methods for online parameter estimation: joint estimation, dual estimation, and observer-based methods [9]. Below, we discuss the findings of this review together with their practical implementation for the MOU state-space model. Observer-based methods are not considered further, as they require a reliable initial estimate of the transition matrix. Additional details on observer methods are provided in section A.7. In addition to joint estimation and dual estimation, we also consider particle filters and gradient descent.

4.3.1 Joint estimation

In joint estimation (JE), the state vector is augmented with the parameter vector, after which a Kalman based method is used to estimate states and parameters simultaneously. This augmentation makes the transition function nonlinear, so it is necessary to choose a Kalman filter that can handle such nonlinearities. JE has been applied in various fields including vehicle dynamics [41]. The augmented state is defined as

$$\boldsymbol{\eta}_k = \begin{bmatrix} \mathbf{s}_k \\ \boldsymbol{\theta}_k \end{bmatrix}. \tag{4.19}$$

The augmented dynamics are described by

$$\boldsymbol{\eta}_{k+1} = f_\eta(\boldsymbol{\eta}_k, \tau_k) + \tilde{\mathbf{w}}_k, \qquad f_\eta(\boldsymbol{\eta}, \tau) = \begin{bmatrix} \boldsymbol{A}(\boldsymbol{\theta}, \tau)\, \boldsymbol{s} \\ \boldsymbol{\theta} \end{bmatrix}. \tag{4.20}$$

With

$$\tilde{\mathbf{w}} \sim \mathcal{N}\big(0, \mathbf{Q}_\eta\big), \qquad \mathbf{Q}_\eta = \begin{bmatrix} \mathbf{Q}_s(\boldsymbol{\theta}, \tau) & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_\theta(\tau) \end{bmatrix}. \tag{4.21}$$

Where $\mathbf{Q}_s$ is the process noise covariance of the state, and $\mathbf{Q}_\theta$ is a small process noise covariance on the parameters that can be added to allow drift.

To use JE with an EKF, the derivatives of the transition function have to be calculated. The derivative of the state with respect to the state is

$$\frac{\partial \mathbf{F}_s}{\partial \mathbf{s}} = A(\boldsymbol{\theta}). \tag{4.22}$$

The derivative of the state with respect to the parameters is

$$\frac{\partial \mathbf{F}_s}{\partial \boldsymbol{\theta}}(\mathbf{s}, \boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial \mathbf{A}}{\partial \theta_1}\mathbf{s} & \cdots & \frac{\partial \mathbf{A}}{\partial \theta_{n_\theta}}\mathbf{s} \end{bmatrix} \quad \in \mathbb{R}^{n_x \times n_\theta}. \tag{4.23}$$

The derivatives of the parameters are

$$\frac{\partial \mathbf{F}_\theta}{\partial \boldsymbol{x}} = \mathbf{0}, \qquad \frac{\partial \mathbf{F}_\theta}{\partial \boldsymbol{\theta}} = \mathbf{I}_{n_\theta}. \tag{4.24}$$

The full Jacobian of the transition matrix can then be described by

$$\mathbf{F}_\eta(\boldsymbol{\theta}, \tau) = \begin{bmatrix} \mathbf{A}(\boldsymbol{\theta}, \tau) & \frac{\partial \mathbf{F}_s}{\partial \boldsymbol{\theta}}(\mathbf{s}, \boldsymbol{\theta}) \\ 0 & \mathbf{I}_{n_\theta} \end{bmatrix}. \tag{4.25}$$

Parameter map To improve stability, the γ parameters are constrained to lie within the interval $[-0.05, 0.8]$, for this a parameter map is used. These bounds were chosen because they cover the range of γ values observed in the data. The parameter mapping is defined using a sigmoid function:

$$\sigma(\theta) = \frac{1}{1 + e^{-\theta}}, \tag{4.26}$$

$$p = p_{\min} + (p_{\max} - p_{\min})\sigma(\theta). \tag{4.27}$$

Its derivative is

$$p'(\theta) = (p_{\max} - p_{\min})\sigma(\theta)\big(1 - \sigma(\theta)\big). \tag{4.28}$$

Where p is the parameter value to which θ is mapped. As an alternative to parameter mapping, clipping can be used. With clipping, the parameters are projected back onto the admissible interval after the correction step of the Kalman filter. However, JE this not converge with this method. A possible explanation is that clipping introduces an inconsistency between the mean and the covariance: the mean is forced to the boundary, while the covariance remains unchanged. Whether this explanation is correct is hard to prove.

Unscented or extended Kalman filter The UKF handles nonlinearities better than the EKF, although it is computationally more expensive. In the JE experiments, the UKF was able to converge to the correct parameter values, which is shown later. To assess whether the EKF could also converge, it was tested using a parameter map.

The EKF failed to converge because the parameters became trapped near the bounds of the parameter map. A likely explanation is that the derivatives of the map approach zero near the boundaries, which effectively slows or prevents further updates. Without parameter map the estimated parameters diverged to extremely high values at the start, and did not converge to the right values.

4.3.2 Dual optimization

Dual estimation has been applied to vehicle systems, surface–atmosphere exchange processes, and the estimation of charge and circuit model parameters in lithium-ion batteries [9]. In this approach, two filters run in parallel: one estimates the system state, while the other estimates the parameters. Decoupling these filters improves numerical stability, although it increases computational complexity and introduces a time lag between state and parameter updates.

Across multiple studies, dual optimization has consistently outperformed joint optimization [9], largely because it lowers the risk of filter divergence. Chebbi et al. also note that, once the parameters have been estimated with sufficient accuracy, the parameter estimator can be deactivated to reduce computational cost. They do not mention if this is only possible in dual estimation, one could argue that the parameter estimation can also be paused in JE. Wenzel et al. demonstrate the implementation of a dual filter for vehicle dynamics [40]. This method is discussed in the following section.

State filter In the state filter the parameters are assumed to equal the current estimate given by the parameter filter.

$$\mathbf{s}_{k+1} = f(\boldsymbol{\theta}_{k+1}, \mathbf{s}_k, \tau_k) + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0},\, \mathbf{Q}_s(\boldsymbol{\theta}, \tau_k)). \tag{4.29}$$

$$\mathbf{z}_k \;=\; h(\mathbf{s}_k) + \mathbf{v}_k, \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0},\, \mathbf{R}). \tag{4.30}$$

Parameter filter In the parameter filter, the parameters can be assumed constant, or process noise can be introduced to allow them to drift. The state-space model is given by

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \mathbf{w}_\theta, \quad \mathbf{w}_\theta \sim \mathcal{N}(\mathbf{0},\, \mathbf{Q}_{\boldsymbol{\theta}}). \tag{4.31}$$

The parameters are not measured directly by the measurement function. Instead, a pseudo-measurement is used, in which the parameters are observed indirectly from their effect on the transition function.

$$h_\theta = h(f(\mathbf{s}_{k-1}, \boldsymbol{\theta}_{k|k-1}). \tag{4.32}$$

For each iteration k, we follow the same four steps as described by Wenzel et al. [40]:

1. Parameter prediction:

$$\boldsymbol{\theta}_{k|k-1} = \boldsymbol{\theta}_{k-1} \tag{4.33}$$

2. State prediction:

$$\mathbf{s}_{k|k-1} = f(\boldsymbol{\theta}_{k|k-1}, \mathbf{s}_{k-1}, \tau) \tag{4.34}$$

3. State correction using measurement $\mathbf{z}_k$ and measurement function h:

$$h\big(\mathbf{s}_{k|k-1}\big). \tag{4.35}$$

4. Parameter correction using measurement $\mathbf{z}_k$ and measurement function h_θ:

$$h_\theta = h(f(\boldsymbol{s}_{k-1}, \boldsymbol{\theta}_{k|k-1}). \tag{4.36}$$

4.3.3 Particle filter

A particle filter (PF) works with many particles that each represent one possible state. As new sensor data comes in, the filter

1. Predicts the next state of each particle.
2. Scores particles based on how well they match the new data, based on their likelihood.
3. Keeps the good particles and discards weak ones.

Over time, the cloud of particles concentrates around the most likely answer. McDougall and Godsill applied a PF to target tracking with a time-varying AR model [32]. They assume that the measurement noise variance σ_m^2, the process noise variance σ_w^2, and the AR coefficient innovation variance σ_θ^2 are linked by known ratios, such that $\sigma_m^2 = r_1\sigma_w^2 = r_2\sigma_\theta^2$. Trajectories are simulated using $\sigma_m^2 = 4 \times 10^{-4}$ and $\sigma_w^2 = 1$, which implies $r_1 = 4 \times 10^{-4}$. This ratio is not representative of a realistic surveillance radar setting, where the measurement variance is typically on the order of $10\,\mathrm{m}^2$. At each iteration, new positions are proposed from the posterior distribution. Rather than being estimated directly, the variances are marginalized out by incorporating the prior on σ_w in the posterior calculation. A Kalman filter is then used to estimate the AR coefficients based on the proposed positions. This combination of a KF and a PF is known as a Rao-Blackwellised PF. By marginalizing out the parameters with a KF, the dimensionality of the PF is reduced, which in turn reduces the variance on estimation [21].

A PF may suffer from particle weight degeneracy, meaning that many particles have a negligible likelihood while only a few carry most of the probability mass. In the work of McDougall and Godsill, this issue is addressed through resampling. When the effective sample size drops below a threshold, particles with low likelihood are discarded and replaced by resampled particles. The effective sample size (ESS) is commonly used to determine when resampling should be performed. It is defined as

$$ESS = \frac{1}{\sum_{i=1}^{N} w_i^2}. \tag{4.37}$$

Here w_i denotes the normalized weight of particle i. Resampling happens when the ESS drops below the threshold, typically $ESS < \frac{N}{2}$ [17].

We make several assumptions for the MOU process that differ from the assumptions made by McDougall and

Godsill. First, the measurement noise is assumed to be known. Second, although there is likely some relationship between σ_m^2 and σ_θ^2, their ratio is unknown and difficult to estimate in advance. Third, data association is not taken into account. Finally, McDougall and Godsill consider a discrete AR process with regular sampling intervals. In that setting, applying a Kalman filter to the AR coefficients gives an exact solution because the parameter effects are linear. For the continuous-time MOU process, however, the problem becomes nonlinear and is therefore likely to be less accurate. Still, a Rao-Blackwellised PF can be implemented: the target state can be marginalized out using a Kalman filter.

A common issue in particle filtering is weight degeneracy, which is usually addressed through resampling. Multiple resampling methods exist, but the most popular is systematic resampling [17]. However, when state estimation is marginalized using a Kalman filter, resampling by duplicating high-likelihood particles is no longer meaningful. This is because a Kalman filter initialized with the same state and parameters will always produce the same state estimate and likelihood. This is a known problem with resampling called path degeneracy, which is when many of the existing particles have common ancestors, so the diversity is low. This degrades state estimation when the amount of measurements increases [1][25]. Diversity can be added in two ways. Innovation noise can be added to parameters at every iteration, which is a good choice if the parameters are indeed time-varying. However, for constant parameters Liu and West state that this method "throws" away information about parameters in assuming them to be time-varying when they are in fact not [30]. When the parameters are constant an alternative method can be used: noise can be added to the parameters when a particle is copied in the resampling process. This method allows for drift on the parameters and creates more particles in the high-likelihood areas, which can improve accuracy. In both methods the noise on the parameters has to be finely tuned. Enough noise has to be added to prevent path degeneracy but too much increases the estimation variance.

4.3.4 Gradient descent

Methods based on gradient descent update the parameters by taking a step in the direction of the gradient for each new sample. One drawback of gradient descent is that its learning parameters must be tuned, which becomes especially difficult in high-dimensional settings. These parameters include, for example, the step size. In addition, gradient descent often does not converge quickly enough to be practical: even in offline maximum likelihood estimation, many function evaluations may be required before reaching a minimum. During offline MLE, of which the results will be shown later, the number of function evaluations could reach more than 500. Even though a line-search instead of gradient descent is used by offline MLE, this number indicates a gradient-descent based algorithm is unlikely to converge fast enough to be practical.

4.3.5 Online estimation conclusion

Joint and dual estimation are both computationally efficient methods for estimating a model's parameters and state, though DE is generally more robust. Both approaches use a Kalman filter that includes the parameters in the state vector, but JE relies on a single filter, whereas DE uses two. Because of its simple implementation, JE will be implemented. If JE turns out to be unstable DE will be used. PF have already been applied successfully to auto-regressive models, but under different assumptions about the target model and with measurement noise assumed to be extremely low. Nevertheless, if a sufficient number of particles is used, the PF should converge to the maximum likelihood estimate. It should therefore perform well for estimation, although the computational cost may become high if many particles are required. Gradient descent is unlikely to converge quickly enough and will therefore not be implemented. Joint and dual estimation are expected to be computationally cheaper than a PF.

4.4 Process Noise Estimation

If process noise depends on the state, the noise becomes non-additive. Additive noise is described by

$$\mathbf{s}_{k+1} = f(\mathbf{s}_k) + \mathbf{w}. \tag{4.38}$$

While non-additive noise is described by

$$\mathbf{s}_{k+1} = f(\mathbf{s}_k, \mathbf{w}). \tag{4.39}$$

In online estimation methods where γ is augmented to the state, $\mathbf{w}$ is non-additive as it depends on the γ values. Das et al. [16] describe how to adjust the UKF to non-additive noise by augmenting the state with a variable $\mathbf{w}$:

$$\mathbf{s}_a = \begin{bmatrix} \mathbf{s} \\ \mathbf{w} \end{bmatrix}, \tag{4.40}$$

and augmenting the estimate covariance by the process noise covariance $\mathbf{Q}$

$$\mathbf{P}_a = \begin{bmatrix} \mathbf{P} & 0 \\ 0 & \mathbf{Q} \end{bmatrix}. \tag{4.41}$$

This formulation allows the UKF to propagate the process noise through the dynamics. The drawback of including $\mathbf{w}$ in the state vector is the added dimensionality. The mean-reverting effect of the γ parameters reduces the magnitude of $\mathbf{Q}$ as γ increases; however, over short time intervals this effect is relatively small. In addition, the covariance matrix $\mathbf{Q}$ is unknown, so an exact treatment of the process noise is not possible anyway. For these reasons, $\mathbf{w}$ is assumed to be additive from this point onward as it is reduces dimensionality and is unlikely to significantly degrade filter performance.

4.4.1 Gradient descent

Gradient descent can be used to estimate the process noise covariance. Yuen et al. [42] demonstrate this approach for the UKF. In their method, the UKF defines an objective

function based on the likelihood, which is then minimized with respect to the noise covariance parameters in the set $\boldsymbol{\varphi}$. The objective function is given by

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{\varphi}_{k+1}) = \frac{1}{2}\Bigg[& \ln\left|\mathbf{P}_{\boldsymbol{z},k+1|k}\right| \\ & + \frac{1}{\boldsymbol{\mu}}\left(\boldsymbol{\varphi}_{k+1} - \boldsymbol{\varphi}_{k|k}\right)^T \boldsymbol{\Sigma}_{\boldsymbol{\varphi},k|k}^{-1}\left(\boldsymbol{\varphi}_{k+1} - \boldsymbol{\varphi}_{k|k}\right) \\ & + \left(\boldsymbol{z}_{k+1} - \boldsymbol{z}_{k+1|k}\right)^T \mathbf{P}_{\boldsymbol{z},k+1|k}^{-1}\left(\boldsymbol{z}_{k+1} - \boldsymbol{z}_{k+1|k}\right)\Bigg]. \end{aligned} \tag{4.42}$$

Here $\boldsymbol{\Sigma}_{\boldsymbol{\varphi},k|k}$ is the covariance in the noise parameter estimation and $\mathbf{P}_{\boldsymbol{z},k+1|k}$ the updated state covariance. This function has no closed-form solution, $\mathbf{P}_{\boldsymbol{z},k+1|k}$ depends on $\boldsymbol{\varphi}$ through all previous sigma-point propagations. At iteration $k+1$, the gradient is approximated numerically by introducing small perturbations to $\boldsymbol{\varphi}_{k|k}$.

If the uncertainty in the noise parameters is large, a random search is performed by evaluating different points on the grid and selecting the one with the highest likelihood. When the noise parameter estimates are close to convergence, a directional search is used instead, choosing among three methods based on which yields the highest likelihood:

1. Gradient descent.
2. Continue in the direction of the previous update.
3. The covariance remains the same.

One advantage of gradient descent is that it optimizes the likelihood objective directly. However, it also has several drawbacks. The step size must be selected carefully. In addition, numerically approximating derivatives with finite differences is computationally costly, as each step requires multiple Kalman filter propagations. Our tracking model relies on several assumptions, including that the target follows the MOU model with Gaussian i.i.d. process noise. Because these assumptions are unlikely to hold exactly in practice, likelihood-based optimization may perform worse than the more robust and computationally cheaper innovation-based matching approach. This method is discussed next.

4.4.2 Innovation-based matching

In innovation-based matching, the process noise covariance is tuned to match the innovation covariance. Shen et al. [36] use innovation-based matching for tracking unmanned surface vessels, which are boats/ships without a crew. They estimate the process-noise covariance online, using a moving-window covariance of the measurement residuals, and then mapping that residual covariance through the Kalman gain. Specifically, at each iteration, the change in the state estimate is determined by the Kalman gain and the innovation error:

$$\delta\mathbf{s}_k = \mathbf{s}_k - \mathbf{s}_{k-1} = \mathbf{K}_k\,\mathbf{e}_k. \tag{4.43}$$

The covariance of $\delta\mathbf{s}_k$ is given by

$$\mathrm{Cov}(\delta\mathbf{s}_k) \;\approx\; \mathbf{K}_k\,\mathrm{Cov}(\mathbf{e}_k)\,\mathbf{K}_k^\top \;\approx\; \mathbf{K}_k\,\mathbf{E}_k\,\mathbf{K}_k^\top. \tag{4.44}$$

Where $\mathbf{E}_k$ is the covariance of the innovation error, computed with a sliding window

$$\mathbf{E}_k = \frac{1}{L}\sum_{i=k-L+1}^{k}\Big(\mathbf{y}_i - h(\mathbf{s}_i)\Big)\Big(\mathbf{y}_i - h(\mathbf{s}_i)\Big)^\top. \tag{4.45}$$

The estimated noise covariance is then given by

$$\mathbf{Q}_k = \begin{cases}(1-d_k)\,\mathbf{Q}_{k-1} + d_k\,\mathbf{K}_k\,\mathbf{E}_k\,\mathbf{K}_k^\top, & k \geq L,\\ \mathbf{Q}_{k-1}, & k < L.\end{cases} \tag{4.46}$$

Here d_k is given by

$$d_k = 0.9\,e^{\,L-k}. \tag{4.47}$$

The rationale behind this method is that the process-noise covariance should be equal to the covariance of $\delta\mathbf{s}$. A drawback, however, is that the innovation error is influenced not only by process noise, but also by errors in parameter estimation. Moreover, during the initial iterations the filter has a high-variance state estimate, which can lead to additional changes in the estimated state. These two factors will inflate $\mathbf{Q}$ at the start of the filter. To address these issues, Shen et al. keep $\mathbf{Q}_k$ fixed for the first 50 iterations, thereby giving the filter time to converge on the state and parameter estimates before beginning to estimate $\mathbf{Q}$. However, this requires a good initial estimate of $\mathbf{Q}$.

Li et al. [29] take another approach by inflating the state estimate covariance based on the innovation residual. The expectation of the innovation residual covariance is equal to

$$E[\mathbf{e}\,\mathbf{e}^\top] = \mathbf{H}\,\mathbf{P}_s\,\mathbf{H}^\top + \mathbf{R}. \tag{4.48}$$

They then estimate the inflation factor by

$$\tilde{\Delta} \approx \frac{\mathrm{Tr}\big(\mathbf{e}\,\mathbf{e}^\top - \mathbf{R}\big)}{\mathrm{Tr}(\mathbf{H}\,\mathbf{P}_s\,\mathbf{H}^\top)} - 1. \tag{4.49}$$

The trace calculates the sum of the diagonal entries, which is in this case the sum of variances. The estimate of $\tilde{\Delta}$ is smoothed over time to obtain the inflation factor Δ. At each iteration before correction, the state estimate covariance is inflated by

$$\mathbf{P}_s^{\text{infl}} = (1+\Delta)\,\mathbf{P}_s. \tag{4.50}$$

Whether it is better to inflate the state estimate covariance $\mathbf{P}$ or the process noise covariance $\mathbf{Q}$ depends on the setting. When the mismatch in innovation covariance is caused by a wrong $\mathbf{Q}$ scale it is better to inflate only $\mathbf{Q}$. If the mismatch is due to more factors, such as model misspecification, outliers or filter divergence, inflating $\mathbf{P}$ is a more robust fix.

4.4.3 Multiple model filter

Rather than estimating $\mathbf{Q}$ directly, one can run multiple filter banks simultaneously within a multiple model (MM)

filter. In this approach, several parallel filters are initialized with different values for $\mathbf{Q}$. The MM filter then either selects the filter bank with the highest likelihood or combines the state estimates of multiple filter banks according to their respective likelihoods. This approach is computationally more demanding, since a separate Kalman filter must be executed for each filter bank. Its main advantage is its robustness and accuracy, provided that at least one filter bank is initialized at a $\mathbf{Q}$ value that is approximately correct.

4.4.4 Process noise estimation conclusion

Innovation-based covariance matching is useful when $\mathbf{Q}$ is unknown. In this approach, $\mathbf{Q}$ is inflated based on the innovation residuals. Alternatively, the state covariance $\mathbf{P}$ can be inflated, which is generally more robust than inflating $\mathbf{Q}$ when the model is misspecified, outliers are present, or the filter begins to diverge. A limitation of innovation-based methods is that they tend to overestimate $\mathbf{Q}$ or $\mathbf{P}$ during the initial iterations, because both state and parameter estimation errors increase the innovation residuals. Shen et al. [36] address this issue by keeping $\mathbf{Q}$ fixed during the first $L = 50$ iterations. However, this strategy relies on an accurate initial estimate of $\mathbf{Q}$. In our case, no prior knowledge of the process noise is available, making a multiple-model filter the more robust choice. Innovation-based covariance matching could still be applied later if additional refinement of $\mathbf{Q}$ proves necessary. Gradient descent has the benefit that it provides an optimal solution if the model assumptions are correct. However, maximizing the likelihood may not be optimal in real-world applications, as these rarely satisfy the model assumptions exactly. Gradient descent also introduces learning parameters that must be tuned, thereby increasing the complexity of the problem. For these reasons, gradient descent will not be used.

Chapter 5

MOU Model Simulations

A total of 100 trajectories of the MOU model are simulated. For this the parameters and initial states are sampled from a distribution based on the real-world data presented in chapter 6. From this point onward, the x-axis refers to the direction parallel to the nominal trajectory, while the y-axis refers to the perpendicular direction.

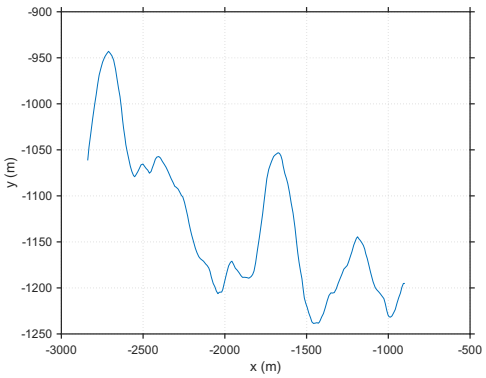
The γ distributions in the real-world data are shown in Appendix B.3. Because the number of data points is limited, the underlying distribution cannot be clearly identified. Since γ must be nonnegative to ensure system stability, and because most observed γ distributions contain at least one value far from the mean, a gamma distribution was selected. The parameter distributions used in the simulations are summarized in Table 5.1. The γ parameters above 0.75 are discarded and resampled, because JE cannot reach those values due to the parameter map. The parameters are assumed to remain constant throughout each run, therefore the innovation noise is set to zero. As discussed later in chapter 6, the standard deviations of p_y and v_y are correlated with the γ and q parameters. Based on this observation, σ_{py} is defined as $\sigma_{py\text{-}theoretical}/170 + 40$, and σ_{vy} as $\sigma_{vy\text{-}theoretical}/17 + 4$, where the theoretical standard deviation is given by Equation 2.41. The initial state distribution is summarized in Table 5.2. After creating the trajectories, polar measurement noise is added with a range variance of $15^2\,m^2$, angle variance of $5^2\,mrad^2$, and a radial velocity variance of $5^2\,(\frac{m}{s})^2$. The targets are assumed to be at a distance of $3000\,m$ from the radar. At $3000\,m$ the noise std in meters due to the noise on the angle is approximately $3000 * 0.005 = 15\,m$, so equal to the noise std in range. Two examples of simulated trajectories (before adding measurement noise) are shown in Figure 5.1.

Table 5.1: Simulation parameter distributions before excluding $\gamma > 0.75$.

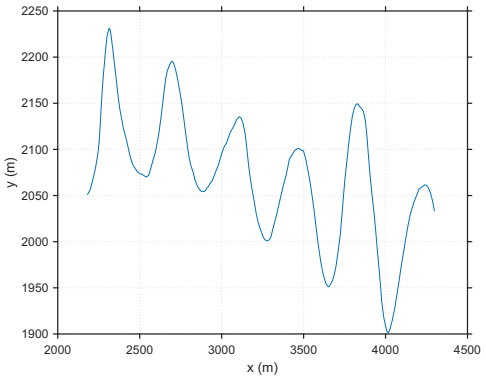
Parameter	Value
γ_{x_1}	$\Gamma(\text{mean}=0.019, \text{var}=0.018^2)$
γ_{x_2}	$\Gamma(\text{mean}=0.288, \text{var}=0.173^2)$
γ_{y_1}	$\Gamma(\text{mean}=0.023, \text{var}=0.013^2)$
γ_{y_2}	$\Gamma(\text{mean}=0.044, \text{var}=0.016^2)$
q_x	4 (fixed value)
q_y	4 (fixed value)
Time interval	$\mathcal{U}(0.8, 1.2)$
Samples	200

Table 5.2: Simulation initial state distribution.

Parameter	Value
p_x	$\mathcal{N}(0, 50^2)$
p_y	$\mathcal{N}(0, \sigma_y^2)$
v_x	$\mathcal{N}(0, 5^2)$
v_y	$\mathcal{N}(0, \sigma_{vy}^2)$
μ_x	$r * cos(\theta)$
μ_y	$r * sin(\theta)$
μ_{vx}	$\mathcal{U}(7, 13)$
μ_{vy}	$\mathcal{N}(0, 1^2)$
r	3000
θ	$\mathcal{U}(-\pi, \pi)$



(a) Inconsistent oscillations



(b) Consistent oscillations

Figure 5.1: Two examples of the simulated trajectories. Most generated trajectories display inconsistent movement as illustrated in (a), whereas only a small number exhibit more consistent behavior as shown in (b).

Tested Models The γ parameters and the state of the simulated trajectories are estimated using joint estimation (JE), maximum likelihood estimation (MLE), and a particle filter (PF). A basic PF is implemented without innovation noise on the parameters and without resampling. A total of 5000 particles are used, which is relatively high given that only four parameters are estimated. The state-estimation performance of the MOU model is compared with that of the NCV model, Singer model, and MOU model with fixed γ. For the fixed-γ model, no parameter estimation is required. Instead, the γ values are fixed at their mean values: $\gamma_{x_1} = 0.019$, $\gamma_{x_2} = 0.288$, $\gamma_{y_1} = 0.023$, and $\gamma_{y_2} = 0.044$.

Prior Distribution The distributions of the parameters and initial states are assumed to be known to all estimators and are incorporated through prior assumptions. The p_y and v_y distributions depend on the γ parameters, the corresponding means and variances are obtained by Monte Carlo simulation and implemented as a Gaussian prior. The γ parameters themselves are also assigned Gaussian priors, even though their true distribution is Gamma. This approximation is made because JE requires Gaussian priors. In principle, MLE and the PF could exploit the true Gamma distribution, but allowing only these estimators to use the exact parameter distribution would make the comparison unfair and would likely overstate their performance relative to practice, where the true distribution is generally unknown. For consistency, Gaussian priors for γ are therefore used for all estimators. Finally, a separate assumption is made for the process noise variance q: rather than being assigned a prior, it is treated as fixed and known. For the NCV model, using the same q value as in the MOU model would be suboptimal, since acceleration induced by the γ parameters introduces additional process noise. To ensure a fair comparison, q is therefore set equal to the process noise variance observed under the NCV model, defined as

$$q = \frac{1}{K} \sum_{k=1}^{K} \frac{(dv)_k^2}{\tau_k}. \tag{5.1}$$

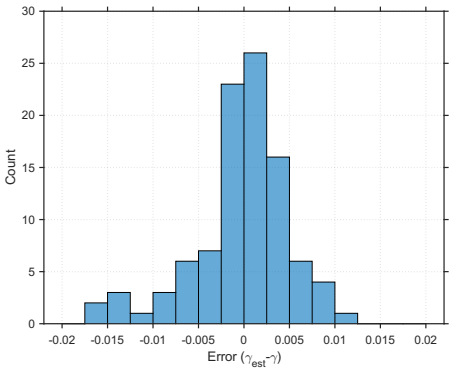
Here, dv denotes the first difference of the velocity, which is equal to the acceleration, and τ_k is the time interval between samples. The value of q is computed from the v_x and v_y, which are known exactly the data is simulated. Likewise, using the MOU q value for the Singer model would be inappropriate because the motion dynamics differ. the Singer model has two unknown parameters per dimension, the decay rate of acceleration and the process noise variance, which are estimated using MLE.

5.1 Parameter Estimation

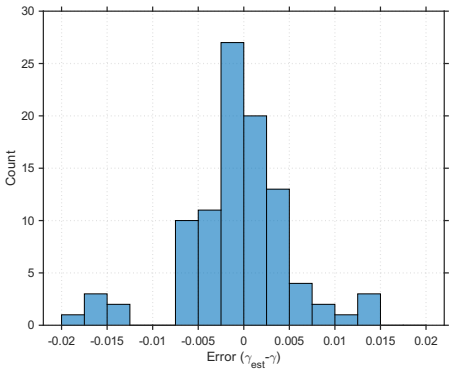
The root mean square errors (RMSEs) of the γ parameter estimates are reported in Table 5.3. Among the three methods, JE yields the most accurate estimates, whereas MLE performs the worst. Of the individual parameters, γ_{1y} is estimated most accurately. A likely explanation is that the simulated weaving patterns produce large oscillations in p_y, causing position reversion to have a strong influence on the target's acceleration. The distributions of the estimation errors for γ_{1y} are shown in Figure 5.2.

Method	γ_{1x}	γ_{2x}	γ_{1y}	γ_{2y}
JE	0.0152	0.0994	0.0069	0.0475
PF	0.0181	0.1670	0.0081	0.0453
MLE	0.0430	0.2101	0.0120	0.0518

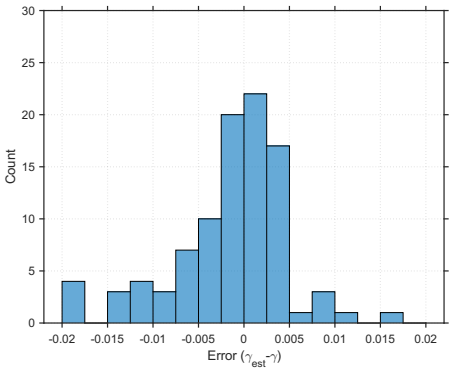
Table 5.3: RMSEs of γ estimation over 100 simulated trajectories. MLE has a much higher RMSE due to a few large errors for γ_{1x} and γ_{1y}, JE has one very large error of 0.45 for γ_{2y} and the PF has no large errors.



(a) JE



(b) MLE

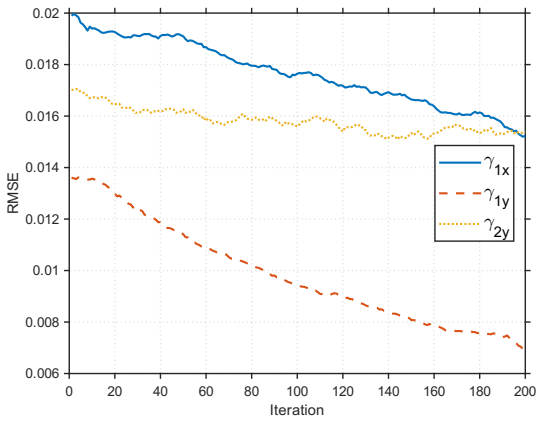


(c) PF

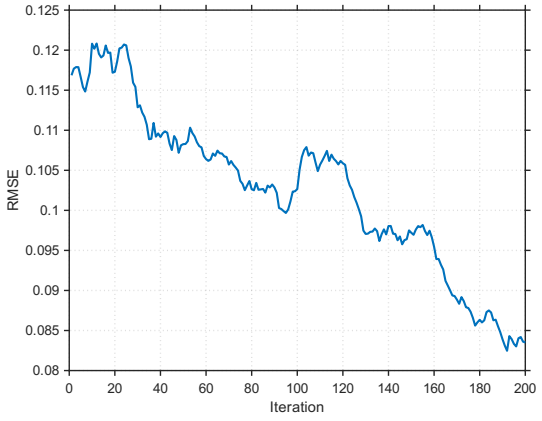
Figure 5.2: γ_{1y} estimation error distribution for JE (a), MLE (b) and the PF (c). The errors in the histogram are bound in the range [-0.02, 0.02] for easier comparison. JE has two values outside this range: -0.0347, -0.0274, MLE has six: -0.0653, -0.0601, -0.0396, -0.0274, -0.0256, 0.0207 and the PF has four: -0.0341, -0.0244, -0.0221, 0.0204.

Based on the histograms all three estimation methods seem to perform approximately equally well. Still, MLE has a RMSE of 0.0120 while JE has a RMSE of 0.0069. This difference is caused by a few large errors outside of the [-0.02, 0.02] range, meaning the MLE is not consistent in its estimation performance. Histograms of the JE errors for the remaining γ parameters are provided in Appendix B.4, the histograms for other methods are excluded to avoid unnecessary repetition. For γ_{1x}, JE achieves the lowest RMSE of the three methods. The comparatively high RMSE for MLE is partly driven by two large estimation errors, 0.264 and 0.262. Excluding these two values reduces the RMSE to 0.022, although this is still higher than the RMSEs obtained by the other methods. For γ_{2x}, no unusually large errors are observed for any of the methods. The higher RMSE values for the PF and MLE, relative to JE, are not caused by a small number of extreme errors, but rather from consistently weaker estimation performance. For γ_{2y}, JE includes one particularly large error of 0.450, omitting this value lowers the RMSE to 0.0154. MLE has a few unusually large errors, whereas the PF appears to estimate γ_{2y} consistently less accurately rather than being affected by only a few unusually large errors. In conclusion, JE has the highest estimation accuracy for all γ values, especially after excluding the the run with a large error value for γ_{2y}.

The convergence behavior of JE for the γ parameters is shown in Figure 5.3. The estimate of γ_{1y} improves over the iterations and is still improving at iteration 200. The remaining parameters are more difficult to estimate, showing only modest, though still noticeable, improvement throughout the iterations. Appendix B.5 shows the convergence of the γ estimates by JE in the absence of measurement noise. In this case, γ_{1x} and γ_{2y} are not estimated more accurately than in the noisy case, whereas γ_{1y} and γ_{2x} are estimated only slightly better. This indicates that measurement noise is not the dominant source of the parameter estimation error in the simulations. Therefore, a reduction in measurement noise by the target moving closer to the radar or implementing a more accurate radar system, is not expected to lead to a substantial improvement in estimation performance.



(a) γ_{1x}, γ_{1y} and γ_{2y}. The run with a large error for γ_{2y} of 0.45 is excluded.



(b) γ_{2x}

Figure 5.3: The JE RMSE of γ estimation at every iteration.

Model / Method	Position	Velocity
Baseline models		
NCV	184.8507	26.8694
Singer	162.8935	22.8269
MOU model		
JE	143.2329	16.2549
MLE	185.1249	18.9795
PF	183.6585	21.5456
Fixed γ	194.9547	24.4725

Table 5.4: Average MSE values for position and velocity for the baseline models (NCV and Singer) and the MOU model under different estimation methods.

5.2 State Estimation

The mean square errors (MSEs) of position and velocity estimation are calculated over the last 50 samples of each segment. The average MSEs over all segments are displayed in Table 5.4, while the corresponding error distributions are shown in Appendix B.1. For clarity, these distributions include only trajectories with a position MSE below 350 m^2 and a velocity MSE below 50 $(\frac{m}{s})^2$. The excluded values are presented in Appendix B.2. First, the performance of the NCV model and the Singer model compared to the MOU model will be discussed. After that, the performance of the different MOU model estimation methods will be discussed.

The NCV model has an approximately 30% higher position MSE and a 65% higher velocity MSE compared to the MOU model estimated by JE. This pronounced difference in velocity estimation performance can be explained by the measurement process. Position is observed directly by the radar and corrected at every time step, preventing the estimate from drifting too far from the true trajectory. Velocity, however, is only inferred indirectly, making it more sensitive to model misspecification and therefore more prone to large estimation errors. The Singer model provides better position and velocity estimates, although it is still less accurate than the MOU model estimated by

JE. This can be explained by the structure of the MOU model, where the acceleration depends on both position and velocity, as expressed by

$$a(t) = -\gamma_1 x(t) - \gamma_2 v(t) + w(t). \tag{5.2}$$

As a result, the acceleration is correlated with itself, since both position and velocity have a strong autocorrelation. The Singer model is able to capture part of this effect, but it assumes the autocorrelation of the acceleration to be constant. In contrast, for the MOU model the autocorrelation depends on the specific combination of position and velocity.

Among all methods, JE achieves the lowest estimation MSE for both position and velocity. The high position error for the PF and MLE is partly driven by a few extreme error values. For MLE, the trajectory with the highest MSE is displayed in Figure 5.4. The position errors are only calculated for the last 50 of the 200 samples, which is exactly the point where a clear change in the trajectory's behavior appears. A possible explanation for why JE captures this behavioral shift while MLE does not is that MLE keeps the γ parameters fixed. As a result, MLE is more confident in its position estimates than JE, as JE accounts for uncertainty in the γ parameters. This makes MLE less flexible when abrupt changes occur. The same reasoning also applies to the PF, since the PF can be viewed as an approximation of MLE. The fixed-γ filter performs well for some trajectories, but also produces many poor position and velocity estimates. This is likely because only a subset of trajectories have γ values close to the γ values in the fixed-γ filter.

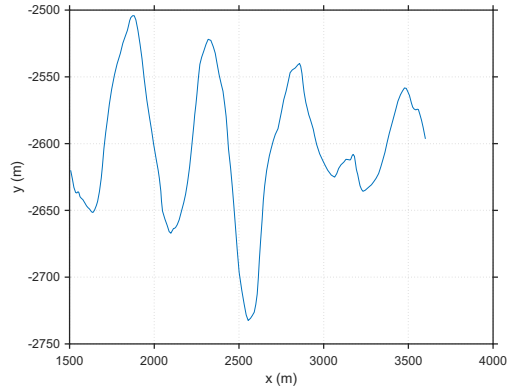
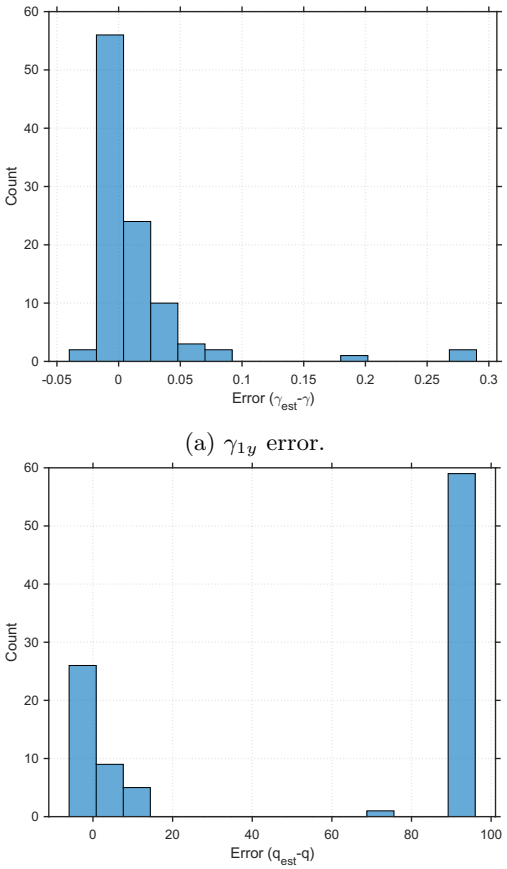


Figure 5.4: The run with the highest position MSE (MSE=1538) for MLE.

5.3 Parameter Estimation of Real Data

In chapter 6, real-world target trajectories are examined. The first step in this analysis is to estimate the distributions of the γ and q parameters using MLE. For the real data, however, the true values of γ and q, as well as their prior distributions, are unknown. For this reason a simulation study is carried out in this section to assess the estimation accuracy. The same prior settings for the states and parameters as used previously are adopted, but the measurement noise differs as GPS measurements are used rather than radar. Assuming typical GPS measurement uncertainties of $1\,m^2$ in the x- and y-positions and $0.1\,\left(\frac{m}{s}\right)^2$ in the x- and y-velocities, the simulated data are generated with a range variance of $1\,m^2$, an angle variance of $0.3^2\,\text{mrad}^2$, and a radial velocity variance of $0.1\,\left(\frac{m}{s}\right)^2$. The estimation errors for γ_{1y} and q_y are presented in Figure 5.5. MLE was implemented as a constraint optimization problem where q has to lie in the range [0,100]. In most cases, the estimate of q_y reached the upper bound of 100, despite the true value being 4. In addition, the wrong q_y estimation results in a high estimation error for γ_{1y}.

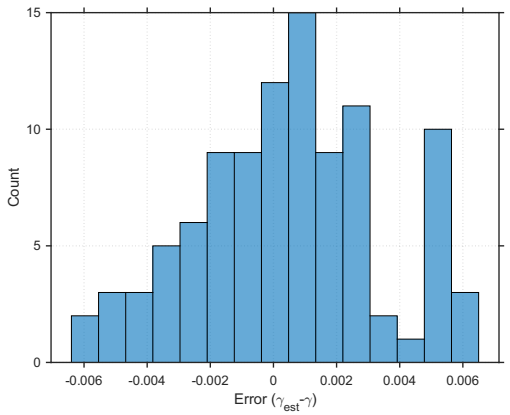


(a) γ_{1y} error.

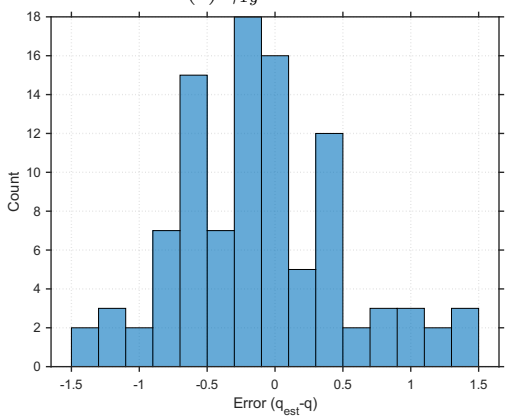
(b) q_y error.

Figure 5.5: Estimation error for simulated data with GPS measurement noise and an unknown initial state.

Further analysis indicates that the overestimation of q_y is caused by uncertainty in the initial state. When the estimator is provided with the initial state, including the mean positions and mean velocities, the parameter estimates converge to the correct values as is shown in Figure 5.6. In an online estimation setting, the initial state is generally unknown. However, for data inspection, the mean positions and velocities can be computed in advance from the full trajectory.



(a) γ_{1y} error.



(b) q_y error.

Figure 5.6: Estimation error for simulated data with GPS measurement noise and a known initial state.

5.4 Conclusion

JE outperforms MLE and the PF when estimating the parameter and state of the target. One possible reason is that JE accounts for uncertainty in the γ parameters when estimating the target state. This makes the filter more flexible, whereas MLE and the PF tend to be overconfident in its state estimates. This becomes especially problematic when the trajectory changes abruptly due to high process noise. Additional state uncertainty could be injected into the target state to compensate for the overconfidence, but this has not been tested. The PF uses a relatively large set of 5000 particles. However, its implementation is still fairly basic. Its performance could potentially be improved by adding jitter to the γ parameters in combination with resampling, allowing for a more focused search over the γ parameters once more information becomes available. At present, the PF does not use resampling, meaning that many particles are effectively wasted in low-information regions with low likelihood. Still, the fact that MLE does not perform well and a PF is an approximation for MLE indicates that a refined PF may still not be as accurate as JE.

The MOU model estimated by JE outperforms both the NCV and Singer models, with the largest gains in velocity estimation. The larger improvement in velocity estimation compared to position when using the MOU model can be attributed to the measurement setup. The x- and y-velocity components are not observed directly, only the radial velocity is measured. Consequently, under a misspecified model (as in the NCV and Singer case), velocity estimates are less effectively corrected by the measurements than position estimates.

γ_{2x} and especially γ_{1x} and γ_{1y} cannot be estimated well by any of the applied estimation methods. This raises the question if the parameters have to be estimated at all, or if it is better to implement a MOU model with a few different fixed γ values in a multiple model filter for these parameters

Lastly, if the γ and q parameter distribution of data is analyzed, the initial state should be known to the MLE approach. Otherwise, the uncertainty on the initial state will cause q to be overestimated and in turn make the γ parameters converge to the wrong values.

Chapter 6

The MOU Filter for Real-World Data

On simulated data, the MOU model achieved a higher state estimation accuracy than the other models. In this section, its performance is evaluated on real-world data. The dataset consists of trajectories from jet skis, water scooters, and rigid hull inflatable boats (RHIBs). The trajectories were collected by instructing the operators of the vehicles to perform weaving maneuvers while their position and velocity were recorded using GPS. The purpose of the data analysis is threefold

1. Analyze the γ and q parameters for each target in the parallel (x) and perpendicular (y) directions.
2. Analyze the state estimation accuracy.
3. Analyze the prediction accuracy.

Measurement noise of data It is important to know how much measurement noise the data contains before starting the data analysis. The quality indicator of the data is reported as "GPS fix", which provides only a general indication of measurement accuracy. A more precise estimate could be obtained from the horizontal dilution of precision, also known as HDOP. This is a commonly used accuracy indicator, but its value is not included in the data file. As a result, only a rough estimate is possible. To evaluate what noise variance values are reasonable, the low-speed segment shown in Figure 6.1 is examined. The smoothness of this segment indicates that the measurement noise is either negligible or strongly correlated, since the measurements show almost no random variation.

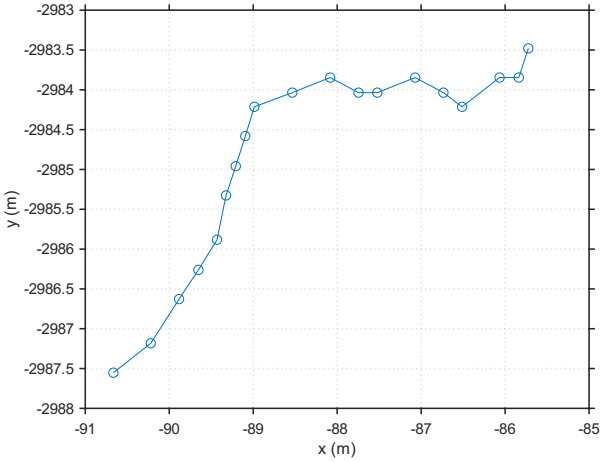


Figure 6.1: GPS measurements of a low-speed segment (RHIB II). The measurement noise seems to be almost non-existent.

One option would be to set the measurement noise to a small value. If the measurement noise covariance is set to be independent and small, while in reality the measurement noise is strongly correlated and large, part of the measurement noise will be absorbed into the process noise. This additional process noise will not drift far from zero. Instead, it is mean reverting. In that sense, the measurement error can also be approximated partly by a MOU process. As a consequence, the estimate of γ reflects a combination of the true trajectory dynamics and the measurement noise dynamics. Nevertheless, in the weaving patterns the oscillation amplitude is much larger than the measurement error, so the bias in γ_1 is expected to be small. The same argument applies to the velocity: the velocity during oscillations is likely much larger than the derivative of the measurement error, so the bias in γ_2 is also likely to be limited. Another option is to the the measurement noise to a high value. If the measurement noise covariance is set to be independent and large to compensate for the ignored correlation, the motion dynamics will appear slow. This misspecification could distort the state and parameter estimates. For instance, high γ values can become difficult to identify because their fast dynamics are masked by measurement noise. In conclusion, assuming a low, independent measurement noise covariance is expected to have only a limited effect on parameter estimation, even if the measurement error is actually high and correlated, and is therefore adopted for the data analysis. The positions and velocities p_x, p_y, v_x, v_y are directly measured by GPS with an assumed measurement covariance of

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}. \tag{6.1}$$

The position variances are given in m^2, and the velocity variances are given in $(\frac{m}{s})^2$.

Selected segments The dataset contains trajectories of five vehicles: RHIB I, RHIB II, water scooter I, water

scooter II, and a jet ski. From these trajectories, segments exhibiting weaving motion were selected and rotated such that the x-axis represents the parallel direction and the y-axis the perpendicular direction. In total, eight weave segments were identified, which are shown in Figure 6.2. Panel (a) shows p_y plotted against p_x. Panel (b) shows the detrended p_x plotted against time, and panel (c) shows the detrended p_y plotted against time. Panel (b) and (c) contain detrended trajectories so it is easier to inspect if mean-reverting motion is present. When looking at segments 1 and 2 in panel (c), which are both segments of RHIB I, we see they closely resemble each other in period and amplitude. Their motion appears similar to undamped oscillation and has a relatively constant amplitude and period, although the mean y-position shifts throughout the segment. Segment 3 of RHIB II resembles a triangular pattern and exhibits substantial variation in both amplitude and period. Segments 4 and 5 of water scooter I display approximately sinusoidal motion. Segment 6 of water scooter II shows no clear consistency in either shape or amplitude, whereas segment 7 appears more sinusoidal. Finally, segment 8 of the jet ski follows a triangular pattern and shows an abrupt change in period and amplitude halfway through the segment.

6.1 Parameter Distribution

A MOU model is fitted on the eight segments. For the analysis of the γ and q parameter values MLE is used, as this method does not require assumptions on the q values. q can be easily added to the parameters over which the model is optimized. This is not possible for JE where the parameter has to be specified beforehand. In a PF it would be possible to optimize over q, but the increased dimensionality of the problem would require more particles resulting in a high computational load. Besides the γ and q parameters another parameter will be optimized: the innovation variance on the mean velocity. The mean velocity can vary within a segment, as illustrated by segment 2 in Figure 6.2a. The mean velocity is initially negative in this segment, reflected in a drift in the negative y-direction. Later, the trajectory reverses and drifts back. To capture this behavior, the innovation variance of the mean velocity, q_μ, is also estimated. The mean position may likewise vary over time, but this is already accounted for through q_μ. The only situation in which a change in mean position would not be captured by a change in mean velocity is an abrupt shift in mean position rather than a smooth drift. Such abrupt changes, however, are only observed in one segment: segment 8. In some segments the weaving frequency changes, suggesting that the γ parameters also vary. However, varying γ parameters cannot be modeled by MLE as this method assumes γ to be fixed.

When q must also be estimated, the estimates of both γ and q become unreliable if the initial state is unknown as shown in section 5.3. To obtain an initial state estimate, linear regression is applied to all position measurements in both the x- and y-directions. The resulting intercept and slope are then supplied to the MLE as the initial mean position and mean velocity, respectively. Since the linear regression estimate is not exact, the initial covariance is set to $10\,m$ for position and $4\,m/s$ for velocity.

The estimated γ parameters, shown in Table 6.1, are converted into the damping factor ζ and the damped frequency w_d using Equation 2.23 and Equation 2.24 respectively. This allows for a more intuitive interpretation of the results. The resulting ζ and w_d values are listed in Table 6.2. The estimated process noise variance parameters q and q_μ are presented in Table 6.3.

Table 6.1: Estimated γ parameters and a summary of their statistics.

#	Segment	γ_{x1}	γ_{x2}	γ_{y1}	γ_{y2}
1	RHIB I	0.0168	0.2076	0.0222	0.0252
2	RHIB I	0.0125	0.1697	0.0233	0.0228
3	RHIB II	0.0117	0.3908	0.0053	0.0347
4	Watersc. I	0.0173	0.1657	0.0251	0.0423
5	Watersc. I	0.0074	0.1317	0.0184	0.0422
6	Watersc. II	0.0633	0.2494	0.0334	0.0715
7	Watersc. II	0.0230	0.2944	0.0486	0.0542
8	Jetski	0.0011	0.6985	0.0091	0.0591

Statistics

Parameter	Mean	Std
γ_{x1}	0.019	0.018
γ_{x2}	0.288	0.173
γ_{y1}	0.023	0.013
γ_{y2}	0.044	0.016

Table 6.2: Estimated ζ and w_d values for all segments

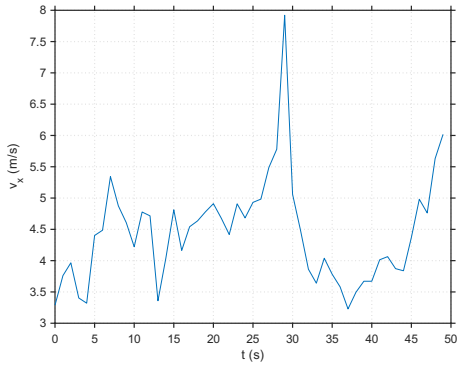
# Segment	ζ_x	w_{d_x}	ζ_y	w_{d_y}
1 RHIB I	0.80	0.08	0.08	0.15
2 RHIB I	0.76	0.07	0.07	0.15
3 RHIB II	1.81	0.16i	0.24	0.07
4 Waterscooter I	0.63	0.10	0.13	0.16
5 Waterscooter I	0.77	0.05	0.16	0.13
6 Waterscooter II	0.50	0.22	0.20	0.18
7 Waterscooter II	0.97	0.04	0.12	0.22
8 Jetski	10.48	0.35i	0.31	0.09

Parallel direction (x) First we take a look at Figure 6.2(b). Segments 1, 2, 3, and 6 visually return to a constant mean, where segment 6 has the most consistent pattern and 3 the least consistent pattern. These observations are verified by the low ζ value for segment 6 and the high ζ value for segment 3. Segment 4 shows oscillatory behavior, but also a change in mean velocity. First there is a drift in the positive x-direction and at the end in the negative x-direction. This indicates a varying mean velocity, which is verified by the high q_μ value. Segment 5 looks similar to segment 4, but $q_{\mu x}$ is estimated to be zero. For

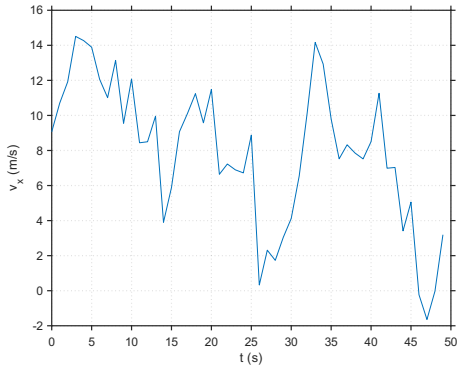
Table 6.3: Estimated q and q_μ values.

#	Segment	q_x	q_y	$q_{\mu x}$	$q_{\mu y}$
1	RHIB I	0.5423	1.9778	0.0000	0.0000
2	RHIB I	0.3118	1.3783	0.0001	0.0016
3	RHIB II	5.3196	5.4271	0.0000	0.0000
4	Watersc. I	6.7282	11.6440	0.0252	0.0000
5	Watersc. I	5.3175	12.0543	0.0000	0.0000
6	Watersc. II	9.0937	13.2584	0.0014	0.0000
7	Watersc. II	3.3780	16.4534	0.0000	0.0000
8	Jetski	3.0133	3.9726	0.0024	0.0007

segment 5 w_d is estimated to be 0.05, which corresponds to an oscillation period of $\frac{2\pi}{0.05} = 126$ seconds. This does not correspond to the period of the oscillations we see in the trajectory (around 60s). Instead, the MOU model captures the long-term change in mean velocity as oscillation. Segment 7 also shows oscillations, but changes its pattern towards the end of the segment. This is likely the cause for its relatively high ζ value. Segment 8 shows oscillations with a small amplitude and high frequency, which explains why ζ and w_d are relatively high. q_x values vary from 0.31 to 9.09. This difference in q_x is illustrated in Figure 6.3, which compares the velocity for segment 2, which has the lowest q_x value, and segment 6, which has the highest q_x value. The higher q_x value in segment 6 is caused by a higher volatility and larger v_x scale in general.



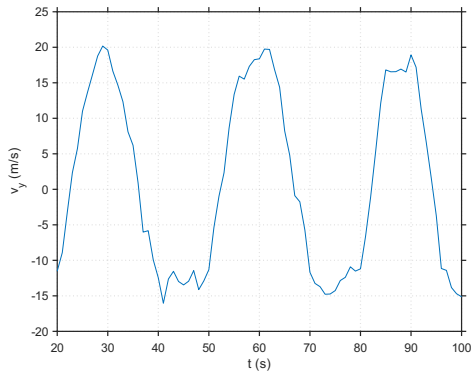
(a) v_x of segment 2.



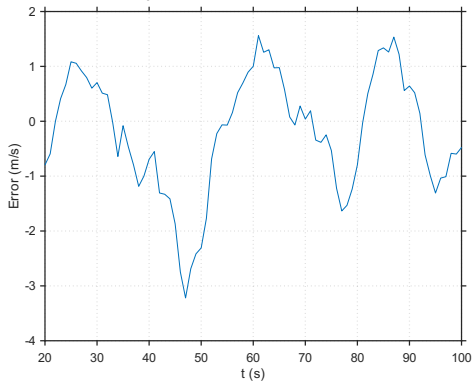
(b) v_x of segment 6.

Figure 6.3: The x velocity of a section of segment 2 and segment 6. Segment 6 shows a higher volatility and a larger scale of velocity.

Perpendicular direction (y) The estimated parameters in Table 6.2 show that all segments are underdamped in the y-direction. This is consistent with the oscillatory motion visible in the detrended y trajectories in Figure 6.2c. Segments 3 and 8, which display variation in both amplitude and period, have the highest damping factors. This is expected, since an inconsistent pattern suggests the oscillations decay relatively quickly. The oscillations in segment 7 display the highest frequency, containing more than three oscillations in 100 seconds. This is in line with its estimated w_d value, which is the largest of all segments and corresponds to a period of $\frac{2\pi}{0.22} = 28.6\,s$.



(a) v_y plotted against time.



(b) v_y one step ahead prediction error.

Figure 6.4: Segment 7: v_y plotted against time (a), and the smoothed one-step-ahead prediction error of v_y: $v_{ytrue} - v_{ypred}$ (b). The error is smoothed by taking the average error for $[t_{k-3}, ..., t_{k+3}])$ at t_k.

The q_y values vary substantially between the segments. Segment 7 has the highest q_y value which is unexpected as the trajectory looks like a clean sinusoid. To understand the high value of q_y, Figure 6.4 shows the smoothed one-step-ahead prediction error of v_y, which is used here as an indicator of the process noise. The error exhibits a clear pattern: it is positive near the maxima and negative near the minima of the oscillations. This indicates that the trajectory does not perfectly follow the MOU model. More specifically, the velocity does not decrease at the rate predicted by the MOU model. This could be explained by the fact that in practice, the velocity likely only starts decreasing when the water scooter starts turning while the MOU model assumes the velocity decay starts earlier. Another reason could be that the turn is sharper than it

should be, which means the velocity-reversion is estimated relatively high to compensate for this. Besides a mismatch in model, the high q_y value for segment 7 may be caused by the relatively high velocity of the water scooter due to its high-frequency oscillations. $q_{\mu y}$ was estimated to be zero for all segments except 2 and 8, which is in line with the observation that both show a clear change in mean velocity.

6.2 State Estimation and Prediction

The practical viability of the MOU model is evaluated by analyzing the filter and prediction performance. To do this a MM is employed, in which JE and NCV filter banks are run in parallel. In contrast to an interacting multiple-model filter, the models in this approach do not interact, making the structure simpler. For each model, the log-likelihood of the measurement is computed at every time step. To reduce sensitivity to short-term fluctuations, this log-likelihood is smoothed using an exponential moving average (EMA) with smoothing factor $\alpha = 0.15$. The EMA is computed as

$$EMA_k = \alpha\, L_k + (1-\alpha)\, EMA_{k-1}, \tag{6.2}$$

where L_k denotes the log-likelihood of the measurement under the corresponding model at time step k. At each time step, the model with the highest EMA value is selected, and only this model is used for state estimation and prediction. In JE it is possible to add innovation noise on the parameters with a variance q_γ. After trial and error q_γ was set to 0.005, as this value offered the best state prediction performance. The MM filter consists of three types of filter banks, where the MOU model is estimated by JE:

- A MOU model with $q_x \in [0.25, 0.5, 1, 3, 6, 9]$ and $q_y \in [1, 3, 5, 10, 15]$ is implemented. $q_\mu = 0$ and $q_\gamma = 0.01$.
- A NCV model with known q. The calculation of q is explained in Appendix A.8.
- A MOU model on one dimension and a NCV model on the other.

Before filtering and prediction, radar-like measurement noise is added in polar coordinates (range, angle and radial velocity), with covariance

$$\mathbf{R} = E[\boldsymbol{\nu}\boldsymbol{\nu}^\top] = \begin{bmatrix} 15^2 & 0 & 0 \\ 0 & 5^2 & 0 \\ 0 & 0 & 5^2 \end{bmatrix}. \tag{6.3}$$

The range variance is given in m^2, the angle variance in $mrad^2$, and the radial velocity variance in $(\frac{m}{s})^2$. The segment center, defined by the mean position, is located at $[x, y] = [2\,\text{km}, 2\,\text{km}]$ relative to the radar. This corresponds to a distance of approximately 3 km from the radar at an angle of 45° counter-clockwise from the x-axis.

Table 6.4: Prior state distribution.

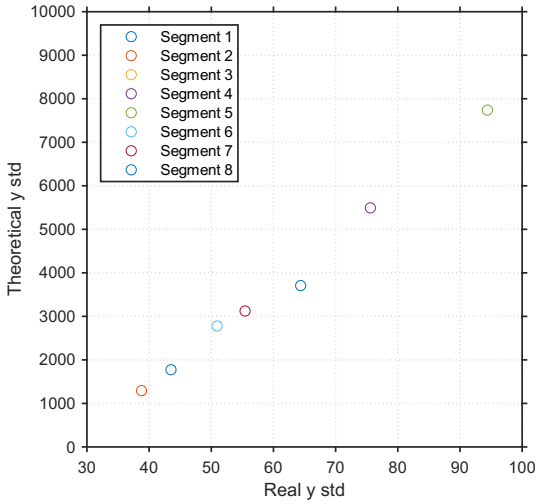
Parameter	Value
p_x	$\mathcal{N}(0, 50^2)$
p_y	$\mathcal{N}(0, \sigma_y^2)$
v_x	$\mathcal{N}(0, 5^2)$
v_y	$\mathcal{N}(0, \sigma_{vy}^2)$
μ_{px}	$\mathcal{N}(0, 3000^2)$
μ_{py}	$\mathcal{N}(0, 3000^2)$
μ_{vx}	$\mathcal{N}(0, 10^2)$
μ_{vy}	$\mathcal{N}(0, 1^2)$

State prior A state prior must be chosen for the Kalman filter. The steady-state covariance formulas for the MOU process are provided in Equation 2.41, their validity will now be assessed empirically. Appendix B.1 reports the empirical standard deviations of position and velocity for each segment. In addition, Figure 6.5 compares the theoretical and empirical standard deviations in the y-direction, where the theoretical values are computed from the γ and q parameters estimated by MLE. The figure shows that the theoretical formulas do not agree with the observed values. This is can be explained as follows: in the theoretical model, the weave amplitude becomes unbounded as γ_2 approaches zero, whereas in practice it remains bounded. Even so, the theoretical and empirical standard deviations are clearly correlated and appear to follow a linear relationship, which suggests that the prior can be improved using the γ parameters. Based on a linear regression on the observed relationship in Figure 6.5, σ_{py} is defined as $\sigma_{py,\text{theoretical}}/170 + 40$, and σ_{vy} as $\sigma_{vy,\text{theoretical}}/17 + 4$. To prevent these quantities from diverging as $\gamma \to 0$, σ_{py} is capped at 300 and σ_{vy} at 30. Because both σ_{py} and σ_{vy} depend on the γ parameters, their means and variances need to be estimated by a Monte Carlo simulation. Specifically, 1000 samples are drawn from the prior distribution of γ, and the corresponding values of σ_{py} and σ_{vy} are computed. The covariance between γ and the state is also estimated and supplied to JE. Based on the observed std in the data, σ_{px} is set to $50\,m$ and σ_{v_x} is set to $5\,m/s$. The mean velocities in the x-direction is also shown in Appendix B.2. Based on this std for μ_{v_x} is set to $10\,m/s$. The y direction has a mean velocity close to 0 due to the rotation of the segment, so its mean velocity std is set to $1\,m/s$. Finally, the std in μ_{px} and μ_{px} are set to $3000\,m$ based on the fact that all trajectories are centered at position $[2000\,m, 2000\,m]$. The prior is summarized in Table 6.4.

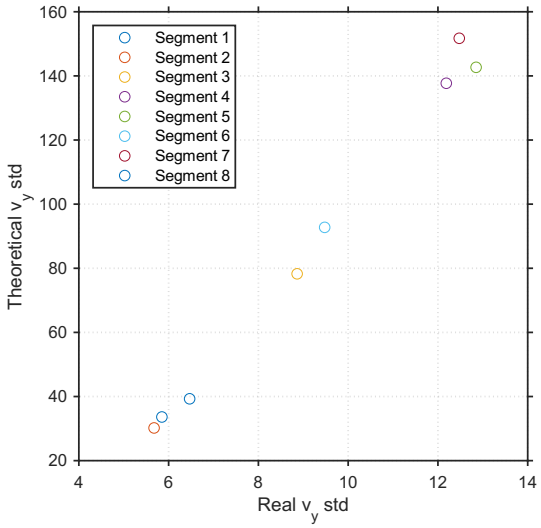
Parameter prior The parameter prior for the Kalman filer is chosen based on the distribution obtained from MLE. The prior is listed in Table 6.5. Some state variables are correlated with the parameter values. Because of this the covariance between γ and the state prior is taken into account.

Table 6.5: Joint filter parameter distributions.

Parameter	Distribution
γ_{x_1}	$\mathcal{N}(0.019,\ 0.018^2)$
γ_{x_2}	$\mathcal{N}(0.288,\ 0.173^2)$
γ_{y_1}	$\mathcal{N}(0.023,\ 0.013^2)$
γ_{y_2}	$\mathcal{N}(0.044,\ 0.016^2)$



(a) Position of dimension y.



(b) Velocity of dimension y.

Figure 6.5: Real std plotted against the theoretical std calculated according to Equation 2.41.

6.2.1 Filter accuracy

The filter accuracy of the MOU model is compared to the accuracy of the NCV model based on the state estimates. The MSEs of the position estimates are shown in Table 6.6, and the MSEs of the velocity estimates are shown in Table 6.7.

In the x-direction the MOU model provides no meaningful improvement in position compared to the NCV model, and for some segments even degrades the accuracy. The

Table 6.6: Position MSE for x and y (MOU vs. NCV).

#	Segment	x		y	
		MOU	NCV	MOU	NCV
1	RHIB I	40.86	43.03	69.52	115.25
2	RHIB I	32.37	41.64	77.32	87.22
3	RHIB II	54.66	57.76	88.30	90.57
4	Watersc. I	94.57	87.35	129.24	140.39
5	Watersc. I	76.69	66.35	61.80	67.01
6	Watersc. II	132.53	112.11	84.04	148.83
7	Watersc. II	36.88	37.61	107.72	131.24
8	Jetski	36.93	43.14	77.88	83.04

Table 6.7: Velocity MSE for x and y (MOU vs. NCV).

#	Segment	x		y	
		MOU	NCV	MOU	NCV
1	RHIB I	1.94	2.52	9.77	15.06
2	RHIB I	1.32	1.67	8.92	11.41
3	RHIB II	5.53	6.58	13.09	11.73
4	Watersc. I	16.84	14.80	21.25	25.22
5	Watersc. I	10.72	9.80	10.78	15.57
6	Watersc. II	18.99	18.76	17.48	24.23
7	Watersc. II	5.40	6.93	12.29	29.14
8	Jetski	2.91	4.36	9.83	10.94

MOU model does also not improve velocity estimates significantly compared to the NCV model. This is consistent with the simulations, which show that the parameters of the x-direction are difficult to estimate. Furthermore, the patterns in the x-direction seem inconsistent, which makes parameter estimation even harder. For these reasons the parameters are likely estimated incorrectly which may results in inaccurate state estimation. Furthermore, the trajectory may not even follow a MOU model in the x-direction.

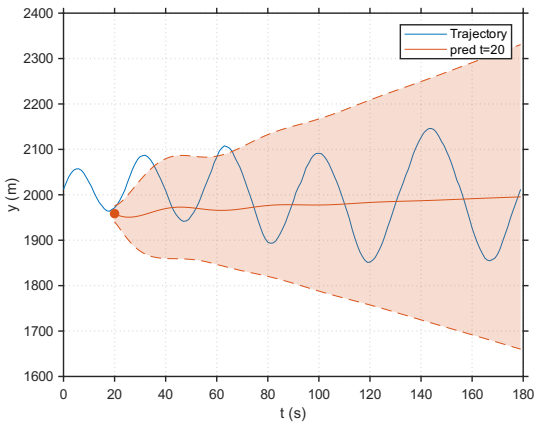
In the y-direction the MOU model generally improves the position and velocity estimates compared with the NCV model. Only segments 3 and 8 see no significant improvement. These segments have a highly varying frequency and amplitude in their oscillations. Moreover, these segments have sharp turns, while the MOU model is better suited to capture sinusoidal motion than motion with sharper turns.

6.2.2 Prediction

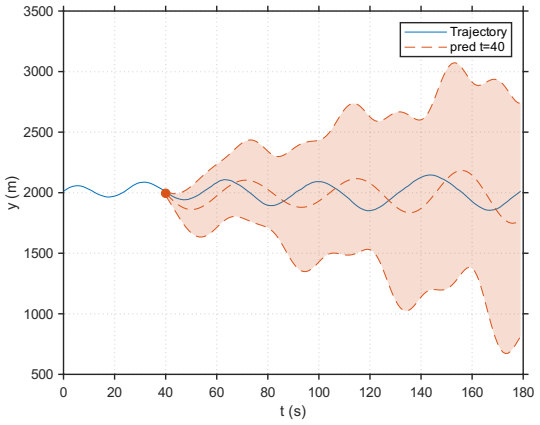
The y-position is predicted using measurements available up to the start time of the prediction. The prediction is shown together with its 95% confidence interval (CI), which is derived from the Kalman filter state covariance. To assess how much information the MM filter requires before its predictions become reliable, predictions were generated from several different starting points. The predictions for segment 4 with starting points at $t = 20\,s$, $t = 40\,s$, $t = 60\,s$, and $t = 80\,s$ are shown in Figure 6.6. The prediction becomes more accurate when the predic-

tion starts later, because the parameters and state are estimated better when more measurements are available. At $t = 80\,s$ the prediction matches the period of the first oscillation well, but not the second. This is because the real trajectory changes frequency, and not because the prediction is incorrect. A time of 80 second therefore seems to be sufficient for a good prediction. For all starting points the CI appears reasonable, as the true trajectory remains within the 95% CI for all starting points. Since this assessment is based on only a single trajectory, it is not sufficient to draw conclusions and additional testing is required.

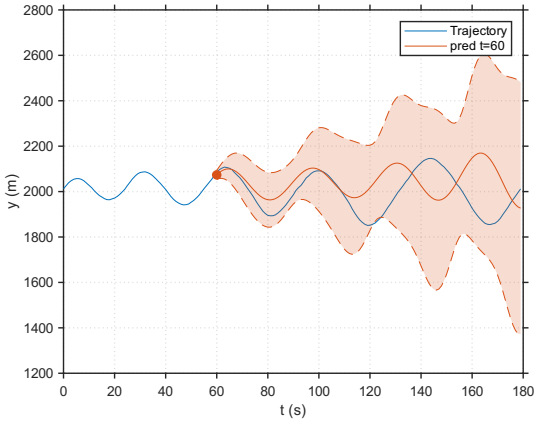
Next, the predicted y-positions for the last 40 samples are shown in Figure 6.7 for all segments. Segments 3 is predicted poorly by the algorithm. This segment contains sharp turns and also exhibits large variations in mean, amplitude, and frequency which indicates the segment has large jumps in its parameters and nominal trajectory. This type of behavior cannot be captured by the MOU filter banks. Segment 7 contains relatively few samples so prediction begins at around $t = 60$ s. Still, the MM filter is able to predict the trajectory fairly well. The turn for segment 8 is also predicted quite well, even though the MOU model did not provide a strong MSE improvement for position and velocity estimation.



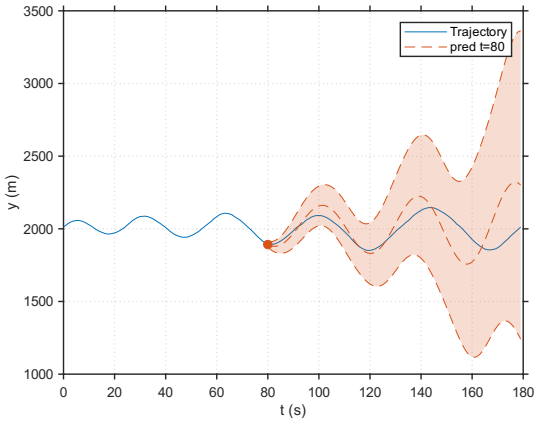
(a) start at t=20



(b) start at t=40



(c) start at t=60



(d) start at t=80

Figure 6.6: Predictions of segment 4 starting at 20, 40, 60 and 80 seconds.

6.3 Fixed γ MOU Model

The simulations indicate that the γ parameters in the MOU model are difficult to estimate. As an experiment, the γ values are set to their prior mean of $\gamma_{x_1} = 0.019$, $\gamma_{x_2} = 0.288$, $\gamma_{y_1} = 0.023$ and $\gamma_{y_2} = 0.044$. The state estimation accuracy of the fixed γ model is compared to the accuracy of the MOU model where γ is estimated. The estimation MSEs is shown in Table 6.8 for position and in Table 6.9 for velocity. The fixed γ MOU model has a slightly larger estimation error than the estimated γ MOU model, but it is still more accurate than the NCV model.

Table 6.8: Position MSE for x and y of the MOU model for an estimated γ and fixed γ.

#	Segment	x		y	
		est. γ	fixed γ	est. γ	fixed γ
1	RHIB I	40.86	43.44	69.52	69.24
2	RHIB I	32.37	32.38	77.32	73.01
3	RHIB II	54.66	56.89	88.30	95.58
4	Water sc. I	94.57	86.43	129.24	125.57
5	Water sc. I	76.69	68.98	61.80	62.25
6	Water sc. II	132.53	156.79	84.04	106.36
7	Water sc. II	36.88	44.86	107.72	107.65
8	Jet ski	36.93	42.22	77.88	89.52

Table 6.9: Velocity MSE for x and y of the MOU model for an estimated γ and fixed γ.

#	Segment	x		y	
		est. γ	fixed γ	est. γ	fixed γ
1	RHIB I	1.94	2.08	9.77	8.61
2	RHIB I	1.32	1.29	8.92	8.10
3	RHIB II	5.53	6.41	13.09	16.54
4	Water sc. I	16.84	15.85	21.25	19.41
5	Water sc. I	10.72	10.39	10.78	11.12
6	Water sc. II	18.99	21.23	17.48	19.95
7	Water sc. II	5.40	6.28	12.29	17.15
8	Jet ski	2.91	3.77	9.83	12.88

The predictive performance of the fixed-γ model was evaluated using the last 40 samples. The resulting predictions are shown in Figure 6.8. Overall, these predictions are less accurate than those of the MOU model with estimated γ, except for segments 5 and 8. The good prediction for segment 8 is unexpected because its ζ and w_d values differ substantially from the mean. The reason for this is unclear. Segment 5 has γ_{y1} and γ_{y2} values that are close to the chosen fixed γ_{y1} and γ_{y2} values. Furthermore, the segment is relatively consistent in its period, indicating the γ parameters do not change throughout the segment. This means the parameters at the moment of prediction are the same as the initial parameters, which match with the fixed γ_{y1} and γ_{y2} values.

Even though the fixed γ model is not as good at prediction as the model with estimated γ values, its predictions are still fairly accurate for many segments. This good performance is likely due to the fact that all segments, expect for 3 and 8, have an approximately similar ζ and w_d value. The narrow w_d range indicates that all weave motions have the same frequency, which may be a result of the instructions given to the operators of the vehicles. It is likely weaves will in practice not lie in this narrow range. The performance of the predictions by the MOU model, which now has a strong prior, and especially the fixed γ MOU model, may therefore be lower in practice.

6.4 Conclusion

The NCV filter was compared with a MM filter that contains both NCV and MOU model filter banks. For segments exhibiting a sinusoidal weaving pattern, the MM filter improved the estimation of the y-position and y-velocity significantly compared to the NCV filter. Moreover, no segment showed a significant increase in either position or velocity MSE for y when the MM filter was used. These results suggest that the MOU model has to potential to improve filtering of weaving patterns in the perpendicular direction to the nominal trajectory. In the parallel direction the estimates are not improved compared to the NCV model.

The MOU model can produce accurate predictions in the perpendicular direction, provided that the motion pattern remains consistent and the tracker has observed the target for a sufficient period of time. The 95% confidence intervals of the predictions appear to be accurate, although statistical testing on a larger dataset is required to confirm this.

A MOU model with fixed γ values proved to be almost as effective for filtering position and velocity as the MOU model with estimated γ values. For some segments, the predictions of the fixed γ model were also fairly accurate. This can likely be attributed to the narrow range in parameters of the dataset and will probably not work as well for new data.

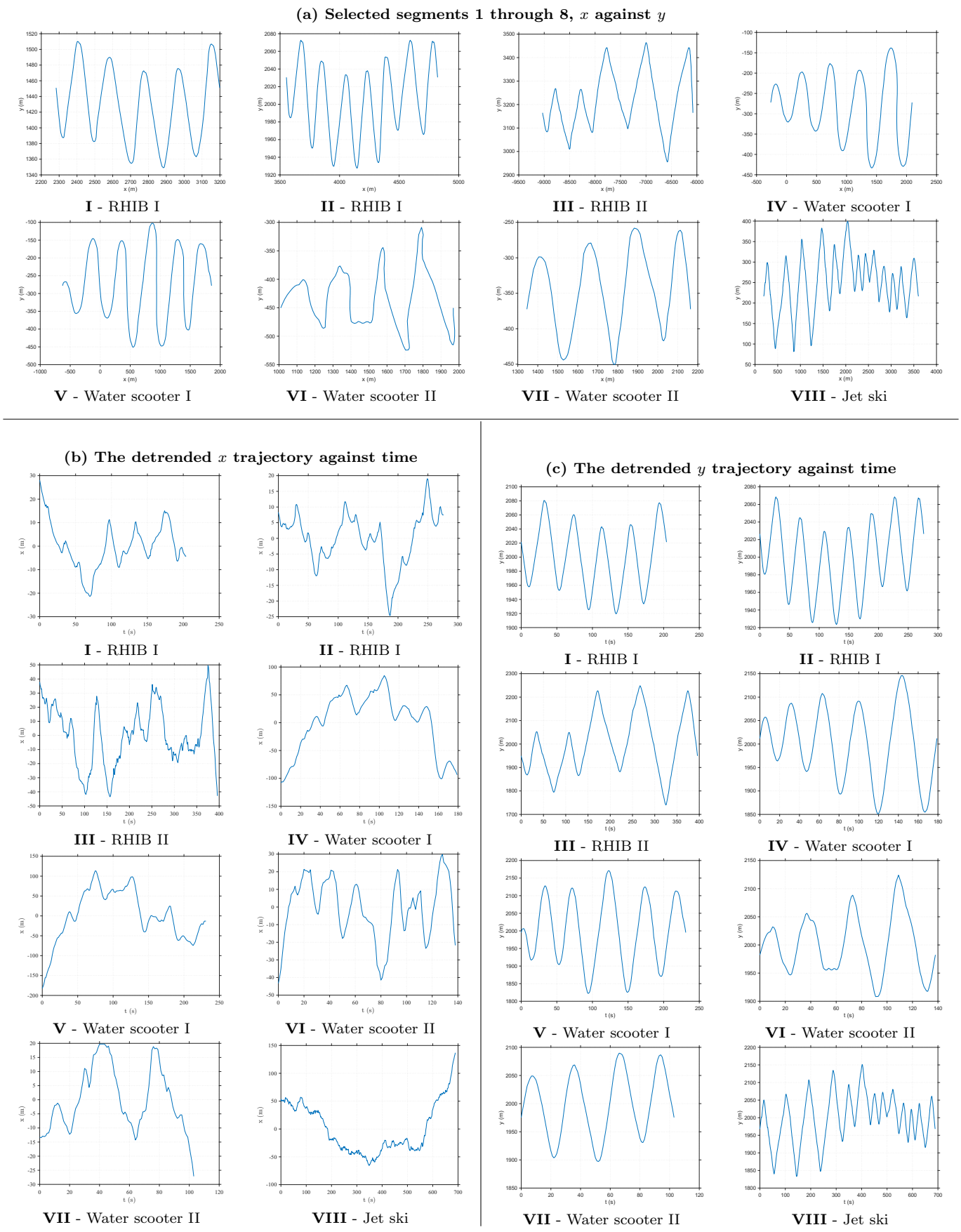


Figure 6.2: Overview of the selected segments 1 through 8. The 2D plot of the trajectory with p_y plotted against p_x (a), the detrended p_x plotted against time (b) and the detrended p_y plotted against time (c).

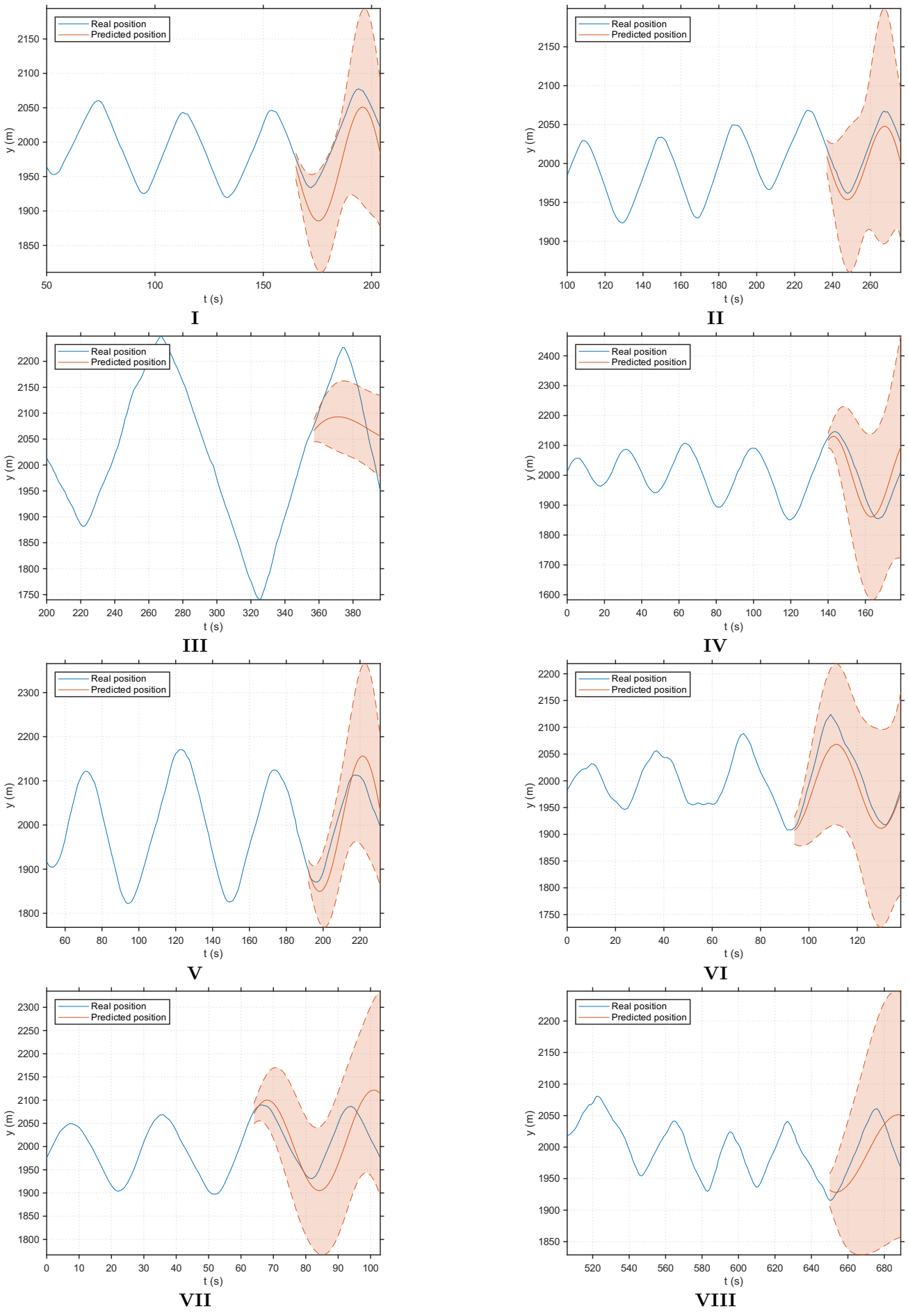


Figure 6.7: The prediction with 95% CI of the last 40 samples for every segment.

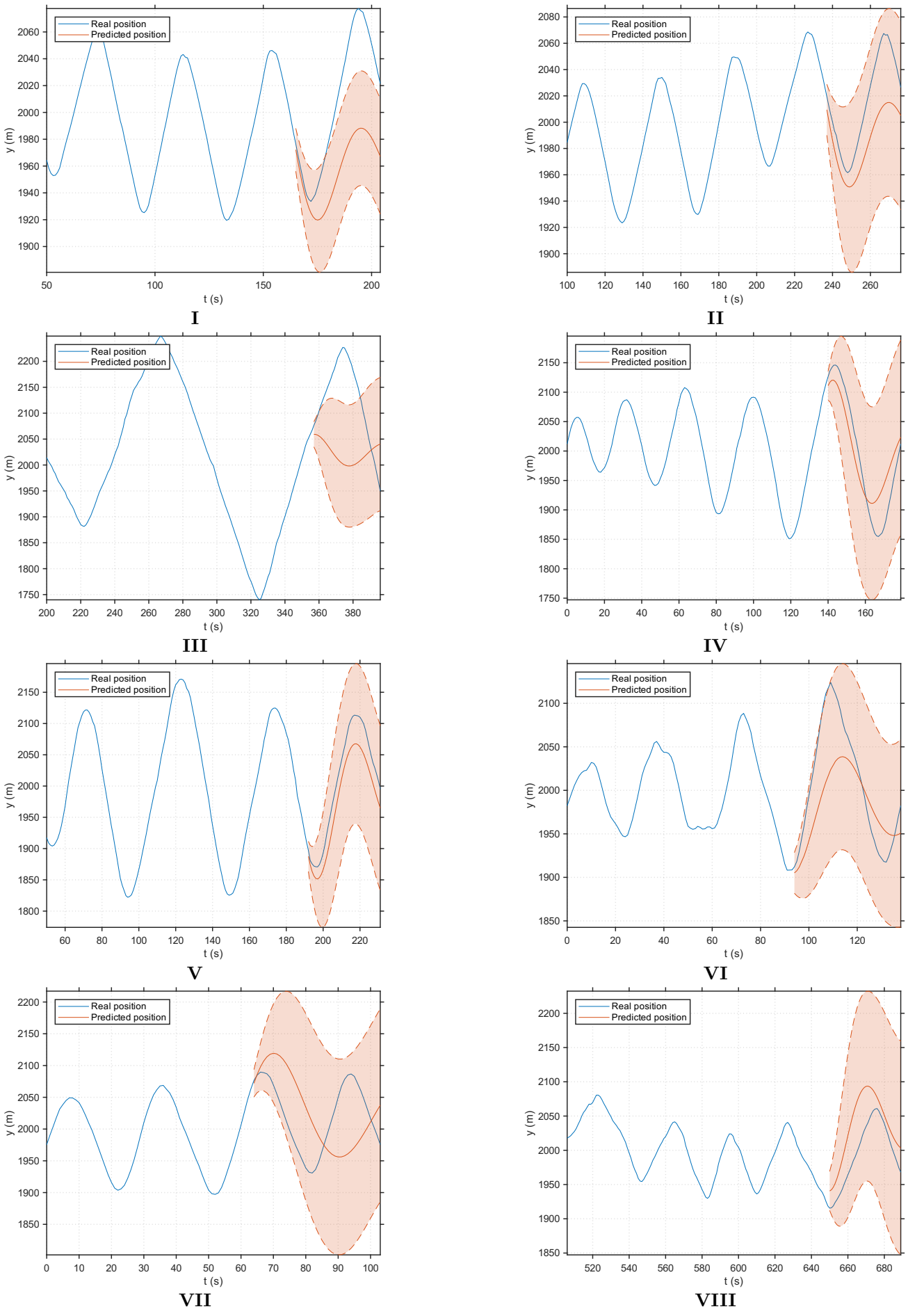


Figure 6.8: The prediction with 95% CI of the last 40 samples for every segment for the fixed γ MOU model.

Chapter 7

Conclusion

The aim of this thesis was to investigate whether the mixed Ornstein–Uhlenbeck (MOU) model can improve target tracking. This section presents the main conclusions. It begins by comparing the MOU model with alternative models, and then outlines its implementation. Next, it discusses the results of the MOU parameter estimation and, finally, evaluates the model's effectiveness for state estimation and prediction.

The dynamics of the MOU model were compared with those of the nearly constant velocity (NCV), Singer, and integrated Ornstein–Uhlenbeck (iOU) models. The NCV and Singer models are not well suited to long-term prediction, whereas the iOU model performs better because it incorporates mean reversion in the velocity. The iOU model has also been used in maritime traffic applications, where it represents trajectories more accurately than the NCV model. However, when a target follows a nominal trajectory, the MOU model is more appropriate because it includes mean reversion in both position and velocity. As a result, for such targets its predictions converge to a steady state rather than increasing without bound.

The implementation of a MOU-based filter for target tracking was described. The most suitable basis is defined by dimensions parallel and perpendicular to the nominal trajectory. For a straight nominal trajectory, this trajectory can be estimated by augmenting the state with a mean position and velocity, which also allows the nominal trajectory to drift over time if needed. A Kalman filter is then applied to process the measurements and estimate the target state.

100 weaving trajectories were simulated using a parameter distribution derived from real-world data. A surveillance radar scenario was then created by adding radar measurement noise, with the target positioned at 3 km from the radar. Under these conditions, joint estimation produced the most accurate parameter estimates, outperforming both maximum likelihood estimation and the particle filter. The MOU model has two transition function parameters per dimension: γ_1 and γ_2. In the direction parallel to the nominal trajectory, the estimates of the position reversion parameter γ_1 and the velocity reversion parameter γ_2 are not substantially more accurate than their prior values. In the perpendicular direction, γ_2 cannot be estimated with meaningful accuracy. By contrast, the position reversion parameter γ_1 can be estimated more reliably, reaching a final RMSE of 0.007 after 200 measurements, and the error was still decreasing at this point. The prior standard deviation for this parameter was nearly 0.014.

The state estimation performance of the MOU model was compared to the NCV model on real-world weaving motion recorded by gps. Radar measurement noise was added to the gps data to create a surveillance radar setting, with the target positioned at roughly 3 km. The MOU model is able to significantly improve both position and velocity estimates in the perpendicular direction of the nominal trajectory compared with the NCV model. The improvement is especially pronounced for the velocity estimate. By contrast, the MOU model cannot estimate the position and velocity better than the NCV model in the parallel direction. The prediction accuracy of the MOU model was also evaluated. The predictions accurately captured the trajectories of almost all weaving segments in the real-world dataset, with the trajectories falling within the 95% prediction interval. However, it remains unclear whether the 95% prediction interval actually contains the true trajectory 95% of the time, since the evaluation of predictive accuracy was qualitative rather than quantitative. Only one segment was inaccurate, but this segment shows substantial variation in mean position, frequency, and amplitude.

7.1 Discussion

The effectiveness of a target tracking model depends on whether the model fits the data. For this reason the dynamics of the MOU model were inspected, which led to the observation that the MOU model is able to represent both damped and undamped motion. In the undamped case, the resulting trajectory corresponds to decaying sinusoidal motion. Many of the trajectories in the data exhibit roughly sinusoidal motion, with only minor deviations. However, other forms of oscillatory behavior are modeled less effectively, such as motion with an irregular period and motion that consists of long straight segments followed by a sharp turn rather than a gradual change in direction. Accurately representing such behavior may therefore require a different type of model. Furthermore, the process noise observed in the data is correlated rather than independent, because velocity does

not begin to decay until the turn starts. To capture this behavior more accurately, it may be preferable to create a model where process noise is correlated.

Under the simulation settings considered, the parameters of the MOU model are difficult to estimate reliably, with the exception of the position reversion parameter in the perpendicular direction. This suggests that estimating the remaining three parameters may be unnecessary and that fixed γ values may be more appropriate instead. However, if the distributions of these parameters vary substantially across targets, a single fixed value is unlikely to perform well. In that case, accurately classifying the target or trajectory type may be necessary in order to assign γ values with sufficient precision.

Although the prior distributions for γ used in the simulations are based on real-world data, the resulting patterns are much less consistent than those observed in practice. This suggests that the real-world damping factor and process noise may be overestimated, likely because vehicle behavior does not perfectly follow the MOU model. This mismatch may lead to an overestimation of the damping ratio, because deviations from the MOU model can make the pattern appear similar to that of a non-stationary MOU model. The estimator may then interpret this as additional decay, even if the true pattern is stationary. Likewise, the process noise may be overestimated, since the model mismatch is effectively treated as extra process noise.

The multiple-model (MM) approach used in the data analysis is not an interacting MM (IMM) filter, but a simpler variant. It determines the most likely model and uses only that model for state estimation and prediction. Unlike an IMM, this method does not account for the relative probabilities of the MOU and NCV models in the data nor the possibility of switching models. If MOU-type trajectories occur only rarely in practice, the transition probability from the NCV model to the MOU model will also be low. As a result, the IMM may be slower to detect an emerging MOU pattern, which can reduce filtering and prediction performance in practice.

7.2 Future work

Future work could extend this thesis in several directions. First, the present study focuses on the applicability of the MOU model to weaving trajectories, which are characterized by underdamped motion and therefore exhibit oscillatory behavior. An interesting next step would be to investigate overdamped motion, in which a target returns to a nominal trajectory without oscillation. Such an extension could be valuable for improving predictions of targets that tend to follow a stable nominal path like ships or airplanes that follow a planned route.

A second direction concerns the treatment of the nominal trajectory. In this thesis, the nominal trajectory is assumed to be straight and unknown, and it is estimated online by the Kalman filter. While this provides flexibility, it also increases the dimensionality of the estimation problem. Instead, future work could explore learning it from historical data. Incorporating prior knowledge of typical target paths may reduce model complexity and further improve state estimation accuracy.

Another important area for future research is the integration of the MOU model with other components of a full tracking framework. In particular, this thesis does not address data association, even though it is a fundamental aspect of target tracking. Investigating how data association methods can be combined with the MOU model would therefore be an important step toward the practical deployment of the model.

The current approach assumes a fixed set of possible process noise variances within a multiple-model filter. Future work could examine innovation-based matching or related adaptive techniques to refine the process noise covariance online, potentially reducing the number of filter banks required and/or making the estimation more accurate.

Further improvements may also be achieved by introducing class-dependent prior information. The parameter distributions of the MOU model may differ across vehicle classes, and with sufficient data it may be possible to construct class-specific parameter distributions. Such priors could improve estimation performance by tailoring the model more closely to the expected behavior of different target categories. Priors on the state itself, such as position and velocity, could also be adapted to each class. Once the prior of each class is known, an interesting research direction would be to do target classification based on the estimated parameters.

Finally, a key assumption in the models developed in this thesis is that the target heading is already known. The reversion and process noise parameters are defined in the parallel and perpendicular directions, and in both the simulations and the analyzed data these directions are aligned with the coordinate axes before filtering. In practice, however, the heading is generally unknown and must also be estimated. Although the presented models can infer heading indirectly through estimation of μ_{vx} and μ_{vy}, this assumes that no prior information about heading is available which would make the parameter and state estimation less accurate. Future work could investigate how to incorporate a heading prior, for example by estimating the target's direction of travel from the trajectory before the onset of the weaving motion.

Bibliography

[1] C. Andrieu, A. Doucet, and V.B. Tadic. On-Line Parameter Estimation in General State-Space Models. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 332–337, Seville, Spain, 2005. IEEE.

[2] J. Belcher, J. S. Hampton, and G. Tunnicliffe Wilson. Parameterization of Continuous Time Autoregressive Models for Irregularly Sampled Time Series Data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 56(1):141–155, January 1994.

[3] A. R. Bergstrom. Gaussian Estimation of Structural Parameters in Higher Order Continuous Time Dynamic Models. *Econometrica*, 51(1):117, January 1983.

[4] D.S. Bernstein and W. So. Some explicit formulas for the matrix exponential. *IEEE Transactions on Automatic Control*, 38(8):1228–1232, 1993.

[5] Samuel S. Blackman and Robert Popoli. *Design and analysis of Modern Tracking Systems*. Artech House, 1999.

[6] P.J. Brockwell. Continuous-time ARMA processes. In *Handbook of Statistics*, volume 19, pages 249–276. Elsevier, 2001.

[7] Joseph E. Cavanaugh and Robert H. Shumway. On computing the expected Fisher information matrix for state-space model parameters. 26(4):347–355.

[8] Marcus J. Chambers and Michael A. Thornton. DISCRETE TIME REPRESENTATION OF CONTINUOUS TIME ARMA PROCESSES. *Econometric Theory*, 28(1):219–238, February 2012.

[9] Amal Chebbi, Matthew A. Franchek, and Karolos Grigoriadis. Simultaneous State and Parameter Estimation Methods Based on Kalman Filters and Luenberger Observers: A Tutorial & Review. *Sensors*, 25(22):7043, November 2025.

[10] Bent Jesper Christensen, Luca Neri, and Juan Carlos Parra-Alvarez. Estimation of continuous-time linear DSGE models from discrete-time measurements. *Journal of Econometrics*, 244(2):105871, September 2024.

[11] S. Coraluppi and C. Carthel. Stability and stationarity in target kinematic modeling. In *2012 IEEE Aerospace Conference*, pages 1–8, Big Sky, MT, March 2012. IEEE.

[12] Stefano Coraluppi, Craig Carthel, Jordan LeNoach, and Brandon Bale. The Ornstein-Uhlenbeck Process in Multi-Target Tracking. In *2021 IEEE Aerospace Conference (50100)*, pages 1–12, Big Sky, MT, USA, March 2021. IEEE.

[13] Stefano Coraluppi, Craig Carthel, West Cummings Park, Woburn Ma, Stefano Coraluppi, Paolo Braca, and Leonardo Millefiori. The Mixed Ornstein-Uhlenbeck Process and Context Exploitation in Multi-Target Tracking.

[14] Stefano P. Coraluppi, Craig A. Carthel, and Alan S. Willsky. Multi-Sensor tracking of move-stop-move targets. In *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6, Bonn, October 2017. IEEE.

[15] Pasquale Coscia, Paolo Braca, Leonardo M. Millefiori, Francesco A. N. Palmieri, and Peter Willett. Multiple Ornstein–Uhlenbeck Processes for Maritime Traffic Graph Representation. *IEEE Transactions on Aerospace and Electronic Systems*, 54(5):2158–2170, October 2018.

[16] Manasi Das, Aritro Dey, Smita Sadhu, and T. K. Ghoshal. Adaptive Unscented Kalman Filter at the Presence of Non-additive Measurement Noise:. In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics*, pages 614–620. SCITEPRESS - Science and and Technology Publications.

[17] Arnaud Doucet and Adam M Johansen. A Tutorial on Particle Filtering and Smoothing: Fifteen years later.

[18] Nicola Forti, Leonardo M. Millefiori, Paolo Braca, and Peter Willett. Bayesian Filtering for Dynamic Anomaly Detection and Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 58(3):1528–1544, June 2022.

[19] Runze Gan, Bashar I. Ahmad, and Simon J. Godsill. Lévy State-Space Models for Tracking and Intent Prediction of Highly Maneuverable Objects. *IEEE Transactions on Aerospace and Electronic Systems*, 57(4):2021–2038, August 2021.

[20] Zoubin Ghahramani. Parameter Estimation for Linear Dynamical Systems.

[21] Simon Godsill and Tim Clapp. Improvement strategies for monte carlo particle filters. In Arnaud Doucet,

Nando de Freitas, and Neil Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 139–158. Springer New York, New York, NY, 2001.

[22] A. C. Harvey. *Forecasting, structural time series models, and the Kalman filter Andrew C. Harvey.* Cambridge University Press, 1989.

[23] A. C. Harvey and James H. Stock. The Estimation of Higher-Order Continuous Time Autoregressive Models. *Econometric Theory*, 1(1):97–117, April 1985.

[24] Richard H. Jones. FITTING A CONTINUOUS TIME AUTOREGRESSION TO DISCRETE DATA. In *Applied Time Series Analysis II*, pages 651–682. Elsevier, 1981.

[25] N. Kantas, A. Doucet, S.S. Singh, and J.M. Maciejowski. An Overview of Sequential Monte Carlo Methods for Parameter Estimation in General State-Space Models. *IFAC Proceedings Volumes*, 42(10):774–785, 2009.

[26] Siem Jan Koopman and Neil Shephard. Exact Score for Time Series Models in State Space Form. *Biometrika*, 79(4):823, December 1992.

[27] E.K. Larsson, M. Mossberg, and T. Söderström. The cramér-rao bound for estimation of continuous-time arx parameters from irregularly sampled data. 38(1):243–248.

[28] Alain Le Breton and Dinh Tuan Pham. Maximum likelihood estimation for continuous-time autoregressive models by relaxation on residual variances ratio parameters. *Mathematics of Control, Signals and Systems*, 6(1):62–75, March 1993.

[29] Hong Li, Eugenia Kalnay, and Takemasa Miyoshi. Simultaneous estimation of covariance inflation and observation errors within ensemble Kalman filter.

[30] Jane Liu and Mike West. Combined parameter and state estimation in simulation-based ltering.

[31] Mahendra Mallick, Linfeng Xu, and Xiaoqing Tian. Cartesian and polar formulations of the ncv model in radar tracking. In *2025 14th International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 57–62, 2025.

[32] Ralph J. Mcdougall and Simon J. Godsill. Target Tracking Using a Time-Varying Autoregressive Dynamic Model. *IEEE Open Journal of Signal Processing*, 6:147–155, 2025.

[33] Leonardo M. Millefiori, Paolo Braca, Karna Bryan, and Peter Willett. Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction. *IEEE Transactions on Aerospace and Electronic Systems*, 52(5):2313–2330, 2016.

[34] M S Phadke and S M Wu. Modeling of Continuous Stochastic Processes from Discrete Observations with Application to Sunspots Data.

[35] Hang Qian. Maximum Likelihood Estimation of Regression Effects in State Space Models. *Journal of Econometric Methods*, 14(1):13–19, July 2025.

[36] Han Shen, Yuezu Lv, Jun Zhou, Linan Wang, and Yuting Feng. An Online Parameter Estimation Method Based on Adaptive Unscented Kalman Filter for Unmanned Surface Vessel. In *2022 34th Chinese Control and Decision Conference (CCDC)*, pages 2584–2589. IEEE.

[37] Robert Singer. Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-6(4):473–483, July 1970.

[38] Michael A Thornton. On the Equivalence of Discrete Time Representations of Continuous Time ARMA Processes.

[39] C. Van Loan. Computing integrals involving the matrix exponential. 23(3):395–404.

[40] T. A. Wenzel, K. J. Burnham, M. V. Blundell, and R. A. Williams. Dual extended Kalman filter for vehicle state and parameter estimation. *Vehicle System Dynamics*, 44(2):153–171, February 2006.

[41] Mark Wielitzka, Matthias Dagen, and Tobias Ortmaier. Joint unscented Kalman filter for state and parameter estimation in vehicle dynamics. In *2015 IEEE Conference on Control Applications (CCA)*, pages 1945–1950, Sydney, Australia, September 2015. IEEE.

[42] Ka-Veng Yuen, Yu-Song Liu, and Wang-Ji Yan. Estimation of time-varying noise parameters for unscented Kalman filter. *Mechanical Systems and Signal Processing*, 180:109439, November 2022.

Appendix A

A.1 ARMA + Measurement noise

We start from

$$y_t = x_t + e_t. \tag{A.1}$$

Apply the AR polynomial $\Phi(B)$ to both sides

$$\Phi(B)y_t = \Phi(B)x_t + \Phi(B)e_t. \tag{A.2}$$

By definition of the latent ARMA(2,1) process

$$\Phi(B)x_t = \Theta(B)w_t, \tag{A.3}$$

so

$$\Phi(B)y_t = \Theta(B)w_t + \Phi(B)e_t. \tag{A.4}$$

Now write the polynomials explicitly as

$$\Phi(B) = 1 - \phi_1 B - \phi_2 B^2, \qquad \Theta(B) = 1 + \theta_1 B, \tag{A.5}$$

to get

$$(1 - \phi_1 B - \phi_2 B^2)y_t = (1 + \theta_1 B)w_t + (1 - \phi_1 B - \phi_2 B^2)e_t. \tag{A.6}$$

Thus the AR side remains the same polynomial $\Phi(B)$, while the noise side is the sum of two filtered white noise processes

$$(1+\theta_1 B)w_t \quad (\text{MA}(1)), \qquad (1-\phi_1 B-\phi_2 B^2)e_t \quad (\text{MA}(2)), \tag{A.7}$$

which are independent. Hence we obtain an AR(2) process driven by a linear combination of two MA noises. The MA order of the sum of these two MA noises is equal to 2 because the auto-covariance of the sum is 0 for lags greater than 2.

A.2 Derivative of the Likelihood

Assume we are looking at a single dimension x. For a given state-estimate $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$, which can be obtained by the Kalman filter, the likelihood of the state-space model is given in Equation 4.12. $\mathbf{C}$ and $\mathbf{R}$ are known. Only the following terms are dependent on γ and σ_w:

$$\ell(\theta) = \sum_{n=2}^{N} \Big[-\frac{1}{2}\big[x_n - \mathbf{A}_n x_{n-1}\big]^\top \mathbf{Q}_n^{-1}\big[x_n - \mathbf{A}_n x_{n-1}\big] - \frac{1}{2} log|\mathbf{Q}_n| \Big] \tag{A.8}$$

Define $\mathbf{e}_n = \mathbf{x}_n - \mathbf{A}_n \mathbf{x}_{n-1}$. Then:

$$\frac{\partial \ell}{\partial \theta_k} = \frac{1}{2}\sum_{n=2}^{N} \Big[-\frac{\partial \mathbf{e}_n^\top}{\partial \theta_k}\mathbf{Q}_n^{-1}\mathbf{e}_n - \mathbf{e}_n^\top \mathbf{Q}_n^{-1}\frac{\partial \mathbf{e}_n}{\partial \theta_k} - \mathbf{e}_n^\top \frac{\partial \mathbf{Q}_n^{-1}}{\partial \theta_k}\mathbf{e}_n - \frac{\partial}{\partial \theta_k} log(|\mathbf{Q}_n|) \Big] \tag{A.9}$$

Since x_t are fixed:

$$\frac{\partial \mathbf{e}_n}{\partial \theta_k} = -\frac{\partial A_n}{\partial \theta_k}\mathbf{x}_{n-1}. \tag{A.10}$$

Using that the first two terms are transposes of the same scalar, we get

$$\frac{\partial \ell}{\partial \theta_k} = \sum_{n=2}^{N} \Big[\mathbf{e}_n^\top \mathbf{Q}_n^{-1}\frac{\partial \mathbf{A}_n}{\partial \theta_k} x_{n-1} - \frac{1}{2}\mathbf{e}_n^\top \frac{\partial \mathbf{Q}_n^{-1}}{\partial \theta_k}\mathbf{e}_n \tag{A.11}$$

$$- \frac{1}{2}\frac{\partial}{\partial \theta_k} log(|\mathbf{Q}_n|) \Big] \tag{A.12}$$

The derivative of $\mathbf{A}_n$ with respect to the γ-parameters involves the derivative of a matrix exponential and is therefore nonlinear:

$$\frac{\partial \mathbf{A}_n}{\partial \theta_k} = \frac{\partial e^{\mathbf{F}t_n}}{\partial \theta_k} \tag{A.13}$$

The derivative of $\mathbf{Q}_n$ w.r.t. the γ parameters is also a nonlinear function. There is therefore no closed-form solution for the γ parameters. The process noise covariance is a linear function of process variance $q = \sigma^2$. Let's split the process noise in two components, the process variance q and the covariance matrix $\mathbf{S}$ (in the case of one dimension, otherwise an independent $q\mathbf{S}$ is needed for each dimension). $\mathbf{Q}_n = q\mathbf{S}_n$. Then the derivative of the likelihood w.r.t. q is:

$$\begin{aligned}
\frac{\partial \ell}{\partial q} &= \sum_{n=2}^{N} \Big[-\frac{1}{2}\mathbf{e}_n^\top \frac{\partial (q\mathbf{S}_n)^{-1}}{\partial q}\mathbf{e}_n - \frac{1}{2}\frac{\partial}{\partial q} log(|q\mathbf{S}_n|) \Big] \\
&= \sum_{n=2}^{N} \Big[-\frac{1}{2}\mathbf{e}_n^\top \frac{\partial q^{-1}}{\partial q}\mathbf{S}_n^{-1}\mathbf{e}_n - \frac{1}{2}\frac{\partial}{\partial q} log(q^2|\mathbf{S}_n|) \Big] \\
&= \sum_{n=2}^{N} \Big[-\frac{1}{2}\mathbf{e}_n^\top \frac{\partial q^{-1}}{\partial q}\mathbf{S}_n^{-1}\mathbf{e}_n - \frac{1}{2}\frac{\partial}{\partial q} [log(q^2) + log(|\mathbf{S}_n|)] \Big] \\
&= \sum_{n=2}^{N} \Big[\frac{1}{2q^2}\mathbf{e}_n^\top \mathbf{S}_n^{-1}\mathbf{e}_n - \frac{1}{q} \Big]
\end{aligned} \tag{A.14}$$

Let's define

$$B := \sum_{n=2}^{N} e_n^\top \mathbf{S}_n^{-1} e_n. \tag{A.15}$$

Setting

$$\sum_{n=2}^{N} \Big[\frac{1}{q} \Big] = \frac{N-1}{q} \tag{A.16}$$

and solving for q gives

$$\boxed{q = \frac{B}{2(N-1)}.} \tag{A.17}$$

This formula can be used to update in an EM algorithm where the parameters are optimized while assuming the state to be fixed. However, during maximum likelihood optimization or in an online filter this closed-form solution will not be effective. Take a look at the error covariance

$$\begin{aligned}\mathbb{E}\big[\mathbf{e}_n\mathbf{e}_n^\top\big] &= \mathbb{E}\big[(\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})(\mathbf{x}_n - \mathbf{A}\mathbf{x}_{n-1})^\top\big] \\ &= \underbrace{(\hat{\mathbf{x}}_n - \mathbf{A}\hat{\mathbf{x}}_{n-1})(\hat{\mathbf{x}}_n - \mathbf{A}\hat{\mathbf{x}}_{n-1})^\top}_{\text{mean residual of estimated means}} \\ &\quad + \underbrace{\mathbf{P}_n + \mathbf{A}\mathbf{P}_{n-1}\mathbf{A}^\top - \mathbf{A}\mathbf{P}_{n-1,n} - \mathbf{P}_{n,n-1}\mathbf{A}^\top}_{\text{uncertainty correction}}.\end{aligned} \tag{A.18}$$

If the state estimation is inaccurate, the uncertainty correction may grow significantly large. This will mean the process noise covariance is over-estimated. To correct for the uncertainty correction the cross covariance $P_{n|n-1}$ has to be known, but this value depends on q. There is therefore no closed-loop solution if the state is unknown and not assumed to be fixed as in an EM algorithm.

A.3 Autocorrelation

A.3.1 Brockwell's approach

Brockwell et al. [6] state that if the eigenvalues λ of the $\mathbf{F}$ matrix are distinct the autocorrelation of a CARMA system can then be calculated by

$$\boldsymbol{\Sigma}(\tau) = \sum_{\lambda:\, a(\lambda)=0} \exp\big(|\tau|\,\lambda\big)\, \frac{b(\lambda)\, b(-\lambda)}{a'(\lambda)\, a(-\lambda)}. \tag{A.19}$$

where a' is the derivative of a and lambda is the eigenvalue of $\mathbf{F}$. An MOU model has b=1 as there are no MA components. a is given by

$$a(z) = z^2 + \gamma_1 z + \gamma_2. \tag{A.20}$$

Brockwell et al. give as example the autocorrelation for the CARMA(2,1) process, where he sets $b_1 = 0$ and $b_0 = 1$ to obtain the autocorrelation for our CAR(2)/MOU process. The result for eigenvalues $a \pm i\beta$ is:

$$\begin{aligned}\boldsymbol{\Sigma}(\tau) &= \boldsymbol{\Sigma}(0)e^{\alpha|\tau|}\left[\cos(\beta\tau) + \sin(\beta|\tau|)\frac{\alpha\big(b_1^2a_2 - b_0^2\big)}{\beta\big(b_1^2a_2 + b_0^2\big)}\right], \\ &= \boldsymbol{\Sigma}(0)e^{\alpha|\tau|}\left[\cos(\beta\tau) - \sin(\beta|\tau|)\frac{\alpha}{\beta}\right].\end{aligned} \tag{A.21}$$

A.3.2 Bernstein's approach

The value of $e^{\mathbf{F}\tau}$ as a function of the damping ratio ζ and natural frequency ω_n is derived by Bernstein et al. [4]. These values can be converted to γ by using equations 2.22–2.24. For the critically damped (repeated eigenvalue) case the value of $e^{\mathbf{F}\tau}$ is given by

$$e^{\mathbf{F}\tau} = e^{-\omega_n\tau}\begin{bmatrix}1+\omega_n\tau & \tau \\ -\omega_n^2\tau & 1-\omega_n\tau\end{bmatrix}. \tag{A.22}$$

For other cases the value is given by:

$$e^{\mathbf{F}\tau} = e^{-\zeta\omega_n\tau}\begin{bmatrix}\cos(\omega_d\tau) + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d\tau) & \frac{1}{\omega_d}\sin(\omega_d\tau) \\ -\frac{\omega_d}{\sqrt{1-\zeta^2}^2}\sin(\omega_d\tau) & \cos(\omega_d\tau) - \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d\tau)\end{bmatrix} \tag{A.23}$$

If the formula is expressed in γ it becomes

$$(e^{\mathbf{F}\tau})_{11} = e^{-\frac{\gamma_2\tau}{2}}\left[\cos\big(\tau\sqrt{\gamma_1 - \tfrac{\gamma_2^2}{4}}\big) + \frac{\gamma_2/2}{\sqrt{\gamma_1 - \frac{\gamma_2^2}{4}}}\sin\big(\tau\sqrt{\gamma_1 - \tfrac{\gamma_2^2}{4}}\big)\right]. \tag{A.24}$$

This result equals the result derived by Brockwell in Equation A.21 and also the result by Coraluppi in Equation 2.45 but expressed in γ_1 and γ_2 parameters instead of the eigenvalues. This formula still needs to be multiplied with the steady-state covariance $\overline{\mathbf{Q}}$ to obtain the covariance function $\boldsymbol{\Sigma}(\tau)$.

A.4 Parameterization

A.4.1 Root fitting

Direct ML estimation of AR coefficients is hard because the stability constraint defines a complicated feasible region. Le Breton and Pham [28] provide a reparameterization which solves this problem by creating a domain where all positive coefficient values correspond to a stable process, and have a 1 to 1 mapping to the real AR coefficients, for this they use residual variances. Jones [24] parametrizes the coefficients in the AR polynomial by factoring the complete polynomial into second-order polynomials: $(s^2+r_1s+r_2)(s^2+r_3s+r_4)$ etc. Then the conditions of stability become

$$\begin{aligned}a_{2k-1} &= -(r_{2k-1} + r_{2k}) = -2\Re(r_{2k}), \\ a_{2k} &= r_{2k-1}r_{2k} = |r_{2k}|^2.\end{aligned} \tag{A.25}$$

If positive, these automatically guarantee a stable process in the case of both conjugate and real roots, a is optimized for $a > 0$ and then converted back to the roots. However, because the 2nd order AR model already has a easy feasible region, namely γ_1 and γ_2 have to be positive, both these parameterizations are not a worthwhile addition.

A.4.2 ARMA root fitting Belcher

Belcher et al. [2] state that the parameterization where the roots of a CAR model are fitted encounters converging problems, and selecting the order is difficult because the models of different orders are not nested. Belcher proposes a new parameterization where discrete AR parameters are fitted, which can be transformed into the continuous AR parameters. Even though the discrete AR parameters are fitted, this still works for irregular-sampled data. The continuous AR polynomial is described by

$$\alpha(s) = s^p + \alpha_1 s^{p-1} + \cdots + \alpha_p. \tag{A.26}$$

The spectral density corresponding to this AR polynomial equals

$$S_m(f) = \left|\frac{1}{\alpha(if)}\right|^2 \sigma^2. \tag{A.27}$$

The continuous frequency can be transformed to discrete frequency using the bilinear transform

$$s = -\kappa \frac{1-z}{1+z}. \tag{A.28}$$

It can be shown that the spectral density in the discrete domain $[0, \pi]$ is equal to:

$$F(\omega) = S(f)\,(\kappa/2)\,\{1 + (f/\kappa)^2\} \tag{A.29}$$

$$= \frac{\kappa}{2}\left|\frac{(1+if/\kappa)}{\alpha(if)}\right|^2 \sigma^2. \tag{A.30}$$

Where σ^2 is the variance of the process noise. The selection of κ is covered later. If a CAR filter is implemented and the noise is filtered by MA terms of the form $(1 + s/\kappa)^{p-1}$, which is a filter with $p-1$ zeros at $s = -\kappa$, the CARMA model becomes

$$\alpha(s)\,y(t) = (1 + s/\kappa)^{p-1}\,\epsilon(t). \tag{A.31}$$

The spectral density becomes

$$F_m(\omega) = \frac{\kappa}{2}\left|\frac{(1+s/\kappa)^p}{\alpha(if)}\right|^2 \sigma^2. \tag{A.32}$$

The discrete AR filter can be transformed by the bilinear transform to obtain

$$\phi(z^{-1}) = \sum_{i=0}^{p} \phi_i z^{-i} \tag{A.33}$$

$$= \frac{\beta(s)}{(1+s/\kappa)^p} \tag{A.34}$$

$$\beta(s) = \sum_{i=0}^{p} \phi_i (1-s/\kappa)^i (1+s/\kappa)^{p-i} = \beta_0 s^p + \beta_1 s^{p-1} + \cdots + \beta_p \tag{A.35}$$

If α is defined by

$$\alpha(s) = \frac{\beta(s)}{\beta_0} = s^p + \alpha_1 s^{p-1} + \cdots + \alpha_p \tag{A.36}$$

With $\beta_0 = \phi(-1)/\kappa^p$. The spectral density can be written as

$$F_m(\omega) = \frac{\kappa}{2}\left|\frac{(1+s/\kappa)^p}{\alpha(if)}\right|^2 \sigma^2 = \frac{\kappa}{2}\left|\frac{\beta_0}{\phi(z^{-1})}\right|^2 \sigma^2 \tag{A.37}$$

The idea of this is that the spectral density $F_m(\omega)$ equals the typical discrete AR spectrum, which is well known to have excellent approximation power on any continuous-valued function on $[0, \pi]$. If the true spectrum can be approximated in the transformed data-space $F(\omega)$ it can also be approximated in the original frequency space f. First the parameters of ϕ are fitted which can be converted to continuous AR parameters by using Equation A.28. The MA component is introduced by setting the observation matrix to

$$\mathbf{H}_i = \begin{cases} \frac{\binom{p-1}{i-1}}{\kappa^{i-1}}, & i = 1, \ldots, p-1, \\ 0, & i = p. \end{cases} \tag{A.38}$$

κ selection

To go from continuous to discrete frequency $\omega = 2tan^{-1}(\frac{f}{\kappa})$ can be used where the continuous frequency is mapped to the discrete domain: $[0, \pi]$. For a regularly sampled time-series with an interval time of h, the Nyquist frequency is $f_N = \frac{1}{2h}$. When $\kappa = \frac{1}{2h}$, $\frac{f}{\kappa} = 1$ and $\omega_N = 2tan^{-1}(1) = \frac{\pi}{2}$.

- Frequencies below Nyquist $0 \le f \le f_N$ map to $0 \le w \le \pi/2$.
- Frequencies above Nyquist $f > f_N$ map to $\pi/2 < w < \pi$.

Each side occupies an interval of length $\pi/2$, hence in the w-space the frequencies above and below the Nyquist frequency have the same range length of $\pi/2$. So the value κ can be optimized by choosing what frequency should be at position $\frac{\pi}{2}$.

A.5 Vandermonde Matrix

Belcher et al. use the Vandermonde matrix to speed up transition matrix calculations [2]. When the roots are not distinct enough they use the standard form

$$\mathbf{x}'(t) = \mathbf{F}\,\mathbf{x}(t) + \mathbf{b}\,d\mathbf{w}(t), \tag{A.39}$$

where $\mathbf{F}$ is the $p \times p$ matrix built from a_j

$$\mathbf{F} = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -a_p & -a_{p-1} & \ldots & -a_2 & -a_1 \end{pmatrix}. \tag{A.40}$$

and

$$\mathbf{b} = (0, \ldots, 0, 1)^\top. \tag{A.41}$$

If the roots are significantly distinct they calculate the Vandermonde matrix with r_n the nth root

$$\mathbf{R_V} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ r_1 & r_2 & \cdots & r_p \\ \vdots & \vdots & \ddots & \vdots \\ r_1^{p-1} & r_2^{p-1} & \cdots & r_p^{p-1} \end{pmatrix}, \tag{A.42}$$

and transform the state with $\mathbf{x} = \mathbf{R_V}\boldsymbol{\eta} \rightarrow \boldsymbol{\eta} = \mathbf{R_V}^{-1}\mathbf{x}$.

$$\boldsymbol{\eta}'(t) = \mathbf{D}\,\boldsymbol{\eta}(t) + \mathbf{J}\,d\mathbf{w}(t), \tag{A.43}$$

where

$$\mathbf{D} = \mathrm{diag}(r_1, \ldots, r_p), \tag{A.44}$$

and $\mathbf{J} = \mathbf{R}^{-1}\mathbf{b}$ is the last column of $\mathbf{R_V}^{-1}$. The observation matrix $\mathbf{H}$ is given by

$$\mathbf{H} = [1, 0, ..., 0]. \tag{A.45}$$

The steady-state covariance is given by

$$\mathbf{P}_{0,ij} = -\frac{J_i J_j}{r_i + r_j}. \tag{A.46}$$

This can be used to calculate the process noise covariance at each step with the Lyapunov equation.

A.6 CARMA to ARMA

The continuous version of the discrete ARMA model is the CARMA model. The MOU model is a continuous autoregressive (CAR) model as it does not have a MA component. The general formula for converting a CARMA model to a discrete ARMA model is given by Theorem 1 in a paper by Chambers [8] where it is shown that a CAR(p) process can be converted to an ARMA(p,p-1) process in regular-interval samples. Bergstrom has derived the formulas to convert specifically the CAR(2) ($\ddot{x} = -\gamma_1 x - \gamma_2 \dot{x}$) process to an ARMA(2,1) process in Theorem 2 of his paper [3] under assumptions

1. Assumption 1: The innovation noise is i.i.d. and zero mean.
2. Assumption 2: γ_2 is non-singular.
3. Assumption 3: The innovation noise is independent of past measurements
4. Assumption 4: $[e^F]_{12}$ is non-singular.

Under these assumptions the ARMA(2,1) process is described by

$$x(t) = f_1 x(t-1) + f_2 x(t-2) + \eta_t \tag{A.47}$$

With f_1 and f_2 scalars equal to

$$f_1 = [e^{\mathbf{F}}]_{12}[e^{\mathbf{F}}]_{22}[e^{\mathbf{F}}]_{12}^{-1} + [e^{\mathbf{F}}]_{11}, \tag{A.48}$$

$$f_2 = [e^{\mathbf{F}}]_{12}[e^{\mathbf{F}}]_{21} - [e^{\mathbf{F}}]_{12}[e^{\mathbf{F}}]_{22}[e^{\mathbf{F}}]_{12}^{-1}[e^{\mathbf{F}}]_{11}. \tag{A.49}$$

And $\eta_t = \epsilon_t + \theta\epsilon_{t-1}$ is noise with autocorrelation at lag 1 due to the MA(1) component and $\mathbf{F}$ is given by

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ -\gamma_1 & -\gamma_2 \end{bmatrix}. \tag{A.50}$$

A.7 Luenberger Observer

The review on simultaneous state and parameter estimation by Chebbi et al. [9] discuss the Luenberger observer. The state-space model for the augmented state can be described by

$$\hat{\eta}_{k+1} = A(\hat{\boldsymbol{\theta}}_k)\hat{\eta}_k + L\big(y_k - C(\hat{\boldsymbol{\theta}}_k)\hat{\eta}_k\big). \tag{A.51}$$

Where

$$A(\boldsymbol{\theta}_k) = \begin{bmatrix} A & A_{\boldsymbol{\theta}}(\boldsymbol{\theta}_k) \\ 0 & I \end{bmatrix}, \quad C(\boldsymbol{\theta}_k) = \begin{bmatrix} C(\boldsymbol{\theta}_k) & 0 \end{bmatrix}. \tag{A.52}$$

In expectation, the error can be described by

$$e_{k+1} = \big(A(\boldsymbol{\theta}) - L\,C(\boldsymbol{\theta})\big)e_k \tag{A.53}$$

The idea of the Luenberger observer is to pick L, such that the error converges to 0. But for that a good approximation of the transition matrix A is needed. Also, in the A matrix Equation A.52 it is assumed that the parameters only function as an additive term on x and do not influence the transition matrix for x itself. The effect of x_t on x_{t+1} is unchanged. The MOU process transition from x_t to x_{t+1} is nonlinear so this approximation will not be accurate. Because of this, and the fact that a good initial guess is needed, the Luenberger observer will not be used.

A.8 NCV Model Process Poise

The process noise of the NCV is calculated using the whole batch before adding measurement noise, so every measurement is known beforehand and the process noise is calculated offline. This gives it an advantage over the MOU model where q is unknown. In the NCV model the acceleration is completely described by the process noise. The variance is calculated as follows: first the positions are converted to velocities by

$$v_k \approx \frac{y_{k+1} - y_k}{\Delta t_k}. \tag{A.54}$$

Where y_k are positions. Then the acceleration is calculated by

$$a_k \approx \frac{v_{k+1} - v_k}{\Delta \bar{t}_k}. \tag{A.55}$$

With

$$\Delta \bar{t}_k = \frac{\Delta t_k + \Delta t_{k+1}}{2}. \tag{A.56}$$

With $\Delta t_k = t_k - t_{k-1}$. $\Delta \bar{t}_k$ is the interval between $\frac{t_{k-1}+t_k}{2}$ and $\frac{t_k+t_{k+1}}{2}$. Equation A.55 is exact under the assumption that the average velocity on interval t_k to t_{k+1} and t_{k-1} to t_k are equal to the velocity at the midpoints at these intervals.

Appendix B

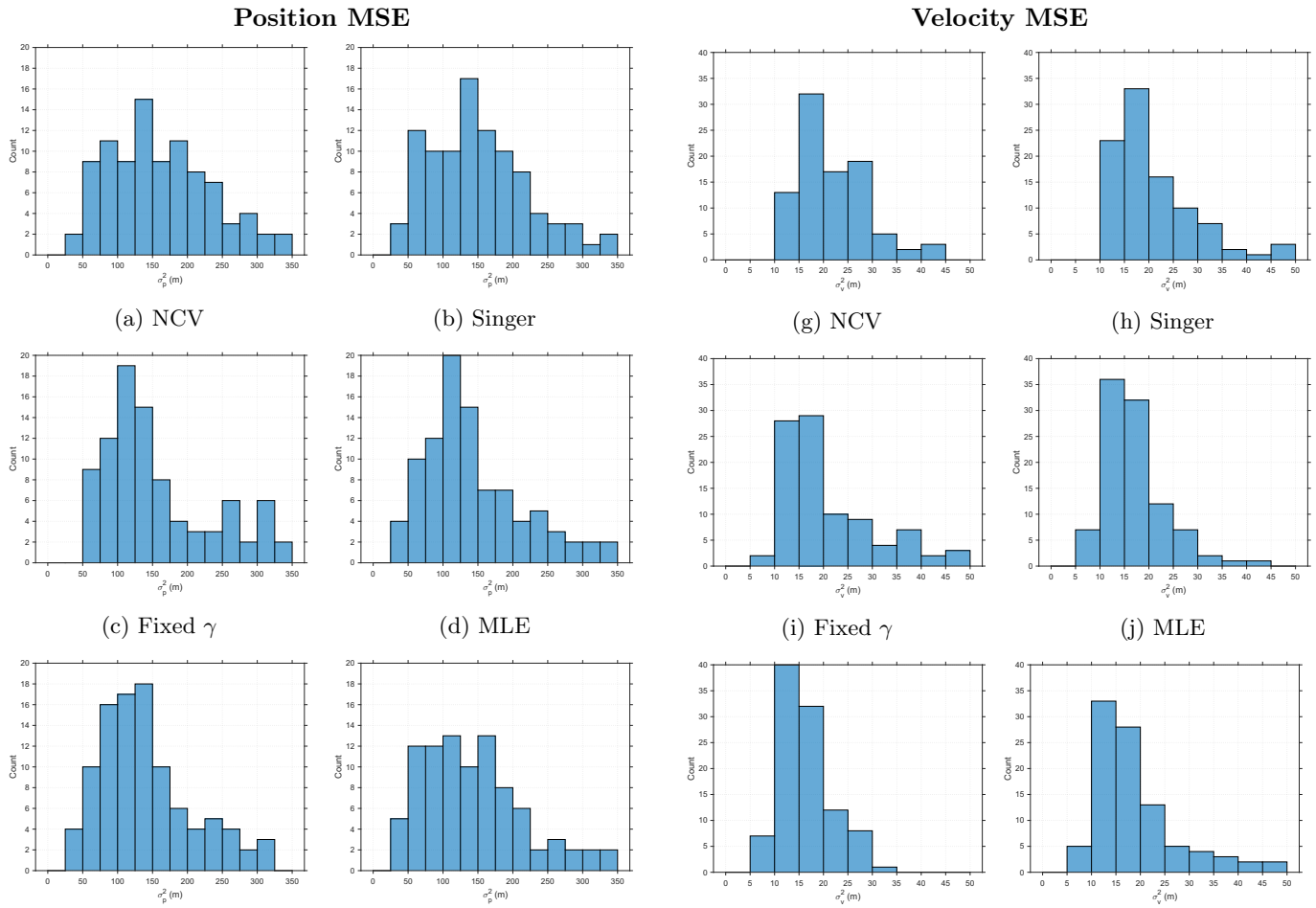
Figures and Tables

#	Segment	x	y	vx	vy
1	RHIB I	9.6414	43.5186	1.2399	6.4680
2	RHIB I	8.2766	38.7782	1.0097	5.6758
3	RHIB II	20.6227	119.1041	2.6253	8.8631
4	Waterscooter I	52.7109	75.5904	4.9786	12.1861
5	Waterscooter I	62.3582	94.4078	4.8937	12.8504
6	Waterscooter II	20.2543	50.9323	4.5789	9.4768
7	Waterscooter II	11.3310	55.4346	2.3388	12.4750
8	Jetski	50.3487	64.3751	1.6373	5.8500

Table B.1: Std for position and velocity.

#	Segment	Value
1	RHIB I	4.49
2	RHIB I	4.60
3	RHIB II	13.15
4	Waterscooter I	10.74
5	Waterscooter I	7.46
6	Waterscooter II	6.98
7	Waterscooter II	7.94
8	Jetski	4.96

Table B.2: Mean velocity in x direction.

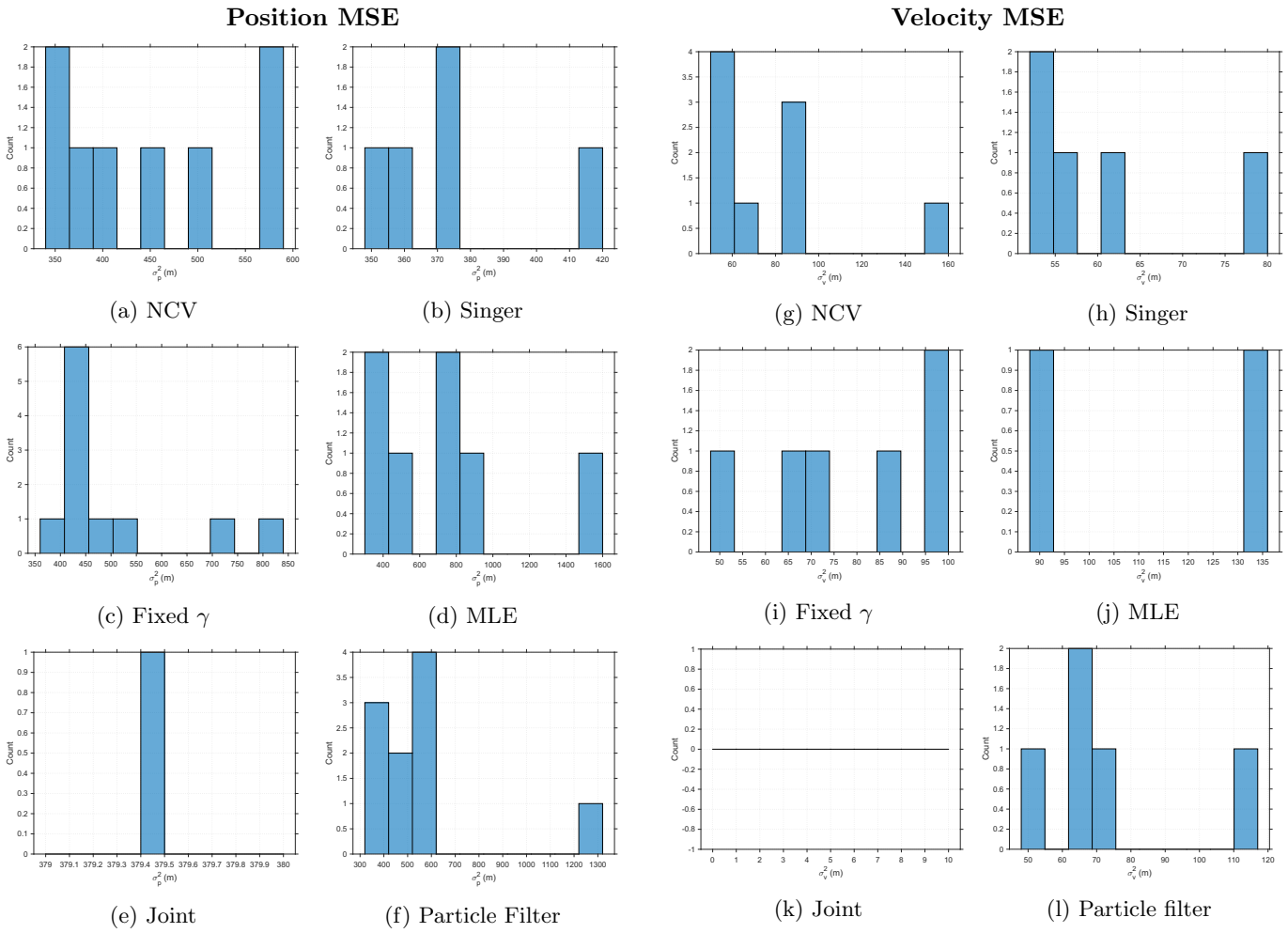


(a) NCV (b) Singer (g) NCV (h) Singer

(c) Fixed γ (d) MLE (i) Fixed γ (j) MLE

(e) Joint (f) Particle Filter (k) Joint (l) Particle filter

Figure B.1: Comparison of the six estimation methods. Left: position MSE, excluding values > 350. Right: velocity MSE, excluding values > 50.

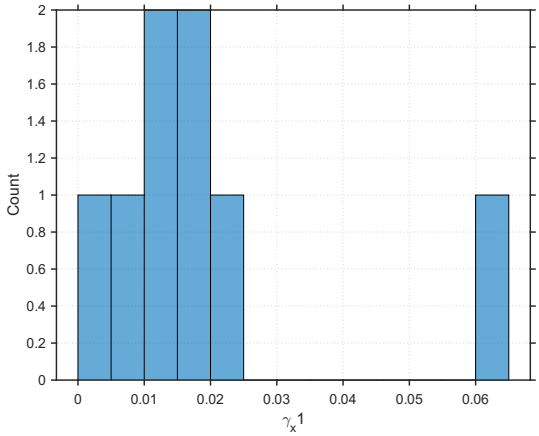


(a) NCV (b) Singer (g) NCV (h) Singer

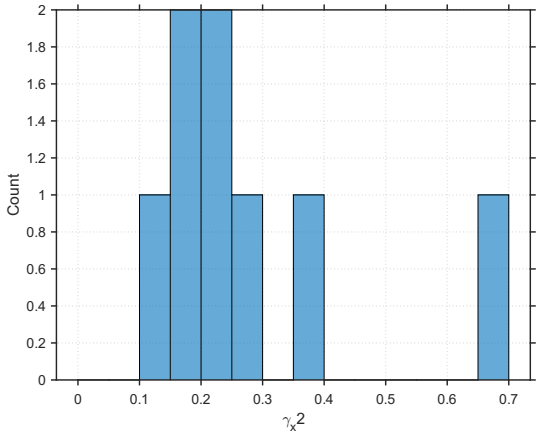
(c) Fixed γ (d) MLE (i) Fixed γ (j) MLE

(e) Joint (f) Particle Filter (k) Joint (l) Particle filter

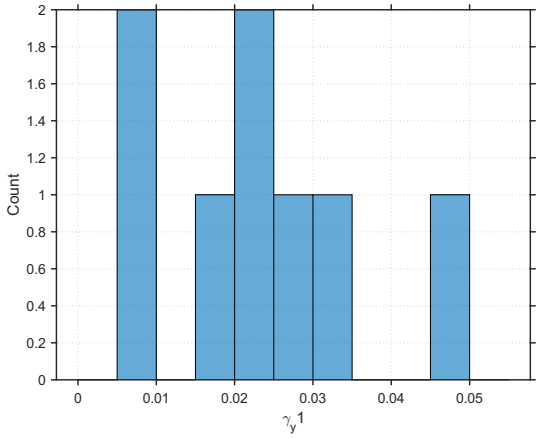
Figure B.2: The excluded position and velocity estimation errors. Left: position MSE values > 200. Right: velocity MSE values > 50.



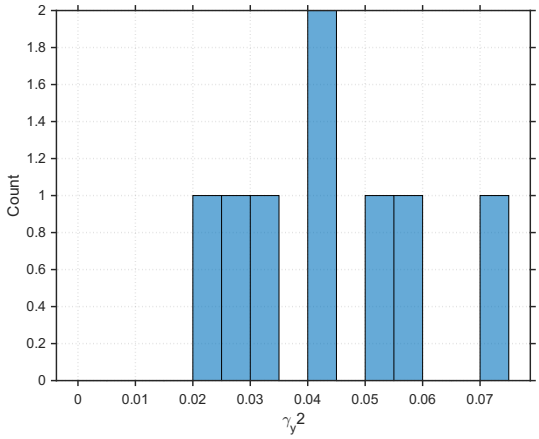
(a) γ_{1x}



(b) γ_{2x}

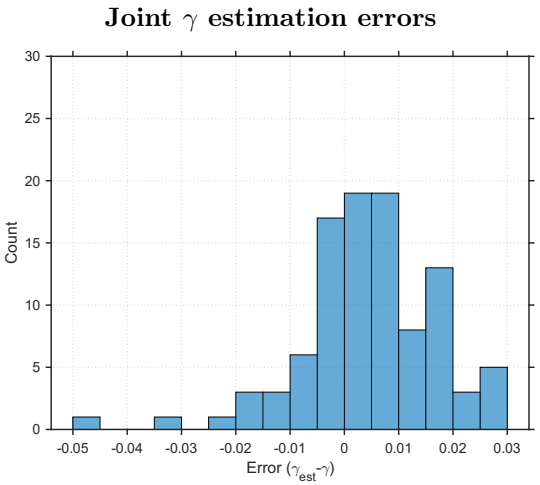


(c) γ_{1y}

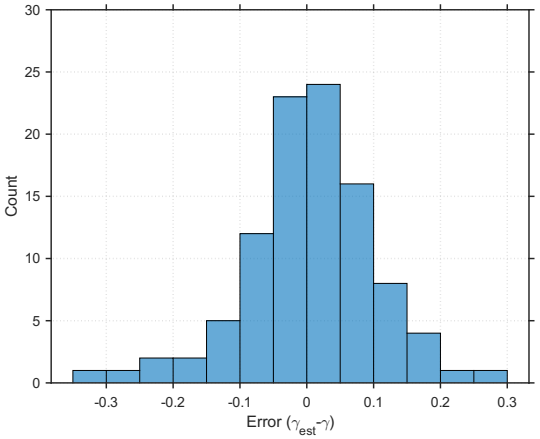


(d) γ_{2y}

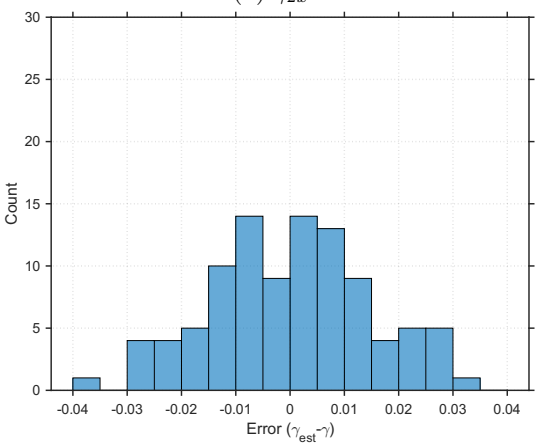
Figure B.3: Distribution of γ parameters in the data.



(a) γ_{1x}

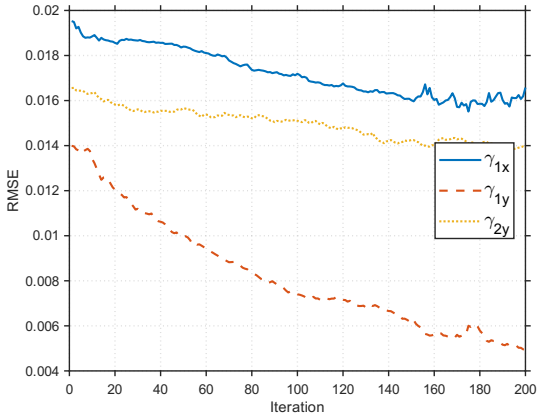


(b) γ_{2x}

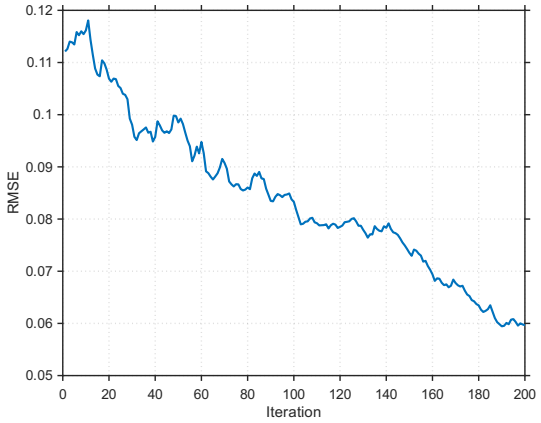


(c) γ_{2y}

Figure B.4: γ estimation errors for JE. Note the different x scales between the histograms. The sample at $\gamma_{2y} = 0.4495$ and the sample at $\gamma_{1x} = -0.0819$ are not displayed.



(a) γ_{1x}, γ_{1y} and γ_{2y}



(b) γ_{2x}

Figure B.5: The JE RMSE of γ estimation at every iteration without measurement noise.