CEAS-GNC-2024-103

EuroGNC CEAS Bristol, UK June 11th-13th 2024
University of Bristol · Royal Aeronautical Society · AIAA

Trajectory Prediction for Missile Targets: A Probabilistic Approach Using Machine Learning

Marc Schneider Research Associate, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569 Stuttgart, Germany. marc.schneider@ifr.uni-stuttgart.de

Renato Loureiro Research Associate, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569 Stuttgart, Germany. renato.loureiro@ifr.uni-stuttgart.de

Torbjørn Cunis Lecturer, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569 Stuttgart, Germany. torbjoern.cunis@ifr.uni-stuttgart.de

Walter Fichter Professor, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569 Stuttgart, Germany. walter.fichter@ifr.uni-stuttgart.de

ABSTRACT

Knowledge about the future trajectory of the target is an essential part of the guidance and control of guided missiles. Most existing prediction models either implicitly or explicitly assume a deterministic behavior of the target. Using this prediction model, the guidance law computes the control commands for the missile. In reality, targets exhibit diverse maneuver possibilities, rendering prediction a stochastic problem. This paper introduces the application of Conditional Normalizing Flows, a data-driven probabilistic approach that leverages Machine Learning to predict the probability density function of the future position of the target. These transform a Gaussian distribution into the target's estimated position distribution based on time and additional parameters describing the target dynamics. Simulated scenarios illustrate the approach's utility for targets with stochastic and deterministic maneuvers, and for ballistic targets as a proof of concept for complex scenarios. Leveraging probabilistic prediction, our approach empowers guidance laws to compute suitable control commands for the missile. This research bridges the gap between deterministic predictions and the stochastic reality of target behavior, contributing to the advancement of missile guidance systems.

Keywords: Probabilistic Trajectory Prediction; Conditional Normalizing Flows; Generative Models; Guided Missiles; Stochastic Target Dynamics

Nomenclature

A	=	reference area
BC	=	ballistic coefficient
C_{D}	=	drag coefficient
d	=	dimension of the position $\mathbf{x}$
$\mathbf{d}$	=	drag vector
f	=	generic scalar-valued function

$\mathbf{f}$ = transformation function
$\mathbf{f}^{-1}$ = inverse transformation function
g = gravitational acceleration
$\mathbf{g}$ = vector of gravitational acceleration
m = mass
ψ = additional parameters regarding the target dynamics
p = probability density function
ρ = air density
Σ = covariance matrix of the acceleration noise
Θ = parameters of the transformation function
t = time
$\mathbf{v}$ = velocity vector
$\mathbf{w}$ = disturbance vector
$\mathbf{x}$ = position of the target
$\mathbf{z}$ = position of the target in the base distribution

1 Introduction

Prediction models are an integral part of the guidance and control of guided missiles. They are used to predict the future trajectory of the target, which is needed for the guidance law to compute the control commands. Most of the existing prediction models assume a deterministic behavior of the target, either implicitly, like the Proportional Navigation (PN) guidance law [1] which assumes a constant velocity vector of the target, or explicitly, like the Zero-Effort-Miss guidance law [2, 3], where arbitrary target dynamics can be assumed.

In reality, the target is not following a single deterministic model indefinitely, but rather a multitude of possible maneuvers is possible. With the knowledge that maneuvers can change, [4] proposes an approach where an Extended Kalman Filter (EKF) is used to not only estimate the position and velocity of the target, but also parameters for the target dynamics.

This approach is taken further with the work of [5], where a multi-hypothesis guidance is employed: An Interacting Multiple Model (IMM) filter is used to estimate the target state and the parameters of the target dynamics, as well as the probability of each hypothesis being true at this moment. The IMM consists of multiple EKFs running in parallel, each with a different target dynamics model (the hypotheses). Then multiple guidance laws are run in parallel, one for each hypothesis, and the control commands are combined using the estimated probabilities of the hypotheses.

The multi-hypothesis guidance is very powerful, but it has two major drawbacks: First, the number of hypotheses is limited by the computational power available, since more hypotheses require more EKFs and guidance laws to be run in parallel. Second, it is still assumed that the target is following a deterministic trajectory, just one of multiple possible ones.

In this paper, a different approach is taken: Instead of assuming deterministic trajectories, a probabilistic approach is used, where the target is assumed to follow a continuous probability distribution of possible maneuvers. However, this gives rise to the problem of predicting the future trajectory of the target. Specifically, the future states of the target cannot be described as a trajectory, but rather as a probability distribution over the state space that changes over time.

Such problems are called multimodal trajectory prediction problems and have mostly been studied in the context of autonomous driving and pedestrians, but not in the context of guided missiles. One challenge for this kind of problem is that the probability distribution over the state space usually cannot be derived analytically, but rather has to be approximated in some way. [6] gives a good overview of the different approaches that have been taken to solve this problem. The methods range from adding noise to a deterministic prediction over anchor methods [7], where likely end points of the trajectories are used to generate multiple trajectories, clustering and Gaussian Mixture Model approaches [8], over grid-based methods [9] to generative models Machine Learning (ML) techniques. Examples of ML approaches are Generative Adversarial Networks (GANs) [10], where a discriminator is trained to distinguish between real trajectories and trajectories generated by a generator network. In a minimax game, the generator network is trained to generate trajectories that are indistinguishable from real trajectories.

Another ML approach is Variational Autoencoders (VAEs) [11], where an encoder network is trained to encode the input trajectory into a latent space, which is then decoded by a decoder network to generate a trajectory.

Lastly, Normalizing Flows (NFs) [12, 13] are a class of generative ML models that can be used to approximate probability distributions by transforming a simple base distribution into the desired, complex distribution and vice versa using a series of invertible transformations. Compared to other ML approaches like GANs or VAEs, NFs have the advantage that they can perform both sampling (of possible future states) and inference (of the probability of a given state). Moreover, they allow for efficient and exact computation of the probability of a given state, giving rise to heatmap-like visualizations of the probability density function (pdf) over the state space.

[14] uses NFs to predict the future trajectories of pedestrians in a multimodal way. It employs recurrent neural networks to encode the past trajectory of the pedestrians which is used to calculate the conditional probability distribution over the future trajectory conditioned on the past trajectory. Each predicted trajectory consists of a sequence of positions with equal time intervals between them. [15] proposes a method to improve the quality and diversity of trajectories generated by NFs. By learning over a set of trajectories instead of samples, improved predictions for the future trajectory of vehicles with discrete time steps can be predicted conditioned on measurements and additional physical attributes. To accommodate for the need for fast inference, [16] proposes a method to speed up the inference of NFs by reusing the results of previous computations to predict future trajectories of humans.

While most prediction methods aim to predict the future positions for discrete time steps, none of them is able to predict the probability distribution of the future positions for arbitrary times, which might be necessary for the evaluation of guidance laws.

Contributions: This paper presents an approach to predict the distribution of the future position of a stochastically moving target using NFs, allowing the prediction of stochastic target dynamics. More precisely, the model can predict samples and the corresponding pdf of the position of the target and for additional parameters regarding the target dynamics for any given time compared to discrete time steps as in existing approaches. Since the model is trained with a dataset of target trajectories created through a Monte Carlo simulation, it can predict the future distribution for any kind of target dynamics, as long as the training data is available, making the approach target-agnostic. This makes it especially useful for the application of guided missiles, since often there is no public data available for the target dynamics.

The remainder of this paper is structured as follows: In section 2, the problem of trajectory prediction for guided missiles is described mathematically. Section 3 explains the theory of NFs and describes their application to the problem at hand. The generation of the training data is described in section 4. Section 5 presents the results of the approach and section 6 concludes the paper.

2 Problem Statement

To guide a missile to its target, the future trajectory of the missile and the target has to be predicted to compute the control commands for the missile in a guidance law. While the future trajectory of the missile can be predicted with high accuracy due to the known deterministic dynamics of the missile, the future trajectory of the target is much harder to predict due to the stochastic nature of the target dynamics. Using a deterministic prediction model for the target, like the PN guidance law, can lead to a loss of performance of the guidance law, since the control commands are computed based on a deterministic prediction of the target trajectory, which is not necessarily the trajectory the target is actually following. The main problem to be solved in this paper is the prediction of the future stochastic trajectory of the target in a computationally efficient manner. It is assumed that the stochastic model of the target dynamics is known or a dataset of target trajectories is available.

The future position $\mathbf{x}$ at time t of a moving target performing random maneuvers is to be predicted given the initial state $\mathbf{x}(t = 0)$ and additional parameters ψ regarding the target dynamics. More precisely, the probability density function $p(\mathbf{x}|t, \mathbf{x}(t = 0), \psi)$ is to be estimated, which describes the the relative likelihood of the target being at a given position $\mathbf{x}$ at time t conditioned on the initial position $\mathbf{x}(t = 0)$ and the parameters ψ. Thus, a function f is to be found, which maps $\mathbf{x}$, t, $\mathbf{x}(t = 0)$, and ψ to a probability density:

$$p(\mathbf{x}|t, \mathbf{x}(t = 0), \psi) \approx f(\mathbf{x}, t, \mathbf{x}(t = 0), \psi) \tag{1}$$

3 Method

Normalizing Flows are a class of generative ML models that can be used to approximate probability distributions by transforming a simple base distribution into the desired, complex distribution and vice versa. Here, the complex distribution is the distribution of the position of the target at a given time and the base distribution is a Gaussian distribution.

3.1 Normalizing Flows

NFs consist of a series of invertible transformations, which transform a sample from the base distribution into a sample from the complex distribution and vice versa. Since the transformations are invertible, the model can be used for both sampling and inference. Sampling means that the model can be used to generate samples from the desired distribution by sampling from the base distribution and transforming the samples with the learned invertible transformations. Inference, or density estimation, means that the model can be used to calculate the probability of a given sample from the complex distribution by transforming it to the base distribution and calculating the probability of the sample in the known base distribution.

As displayed in Figure 1, NFs describe a transformation $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$, which transforms samples $\mathbf{x}$ with dimension d from the complex distribution $p(\mathbf{x})$ into samples $\mathbf{z}$ with the same dimension from the base distribution $p(\mathbf{z})$. In this application, $p(\mathbf{z})$ is a Gaussian distribution with zero mean and unit variance and $p(\mathbf{x})$ denotes the distribution of the position $\mathbf{x}$ of the target.

Equation 2 describes the transformation $\mathbf{f}$ and its inverse $\mathbf{f}^{-1}$ as a composition of n invertible transformations $\mathbf{f}_{\Theta_i}$, where Θ_i are the parameters of the i-th transformation. The parameters Θ_i are the outputs of neural networks (NNs), which are trained to learn the parameters of the transformations. The input for the NNs is the output from the previous transformation $\mathbf{f}_{\Theta_{i-1}}$ as illustrated in Figure 2.

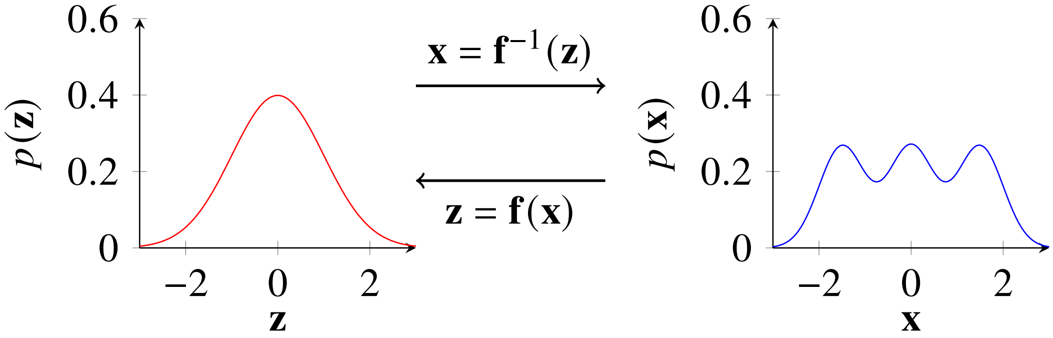


Fig. 1 Illustration of the transformation from a normal distribution to a complex distribution.

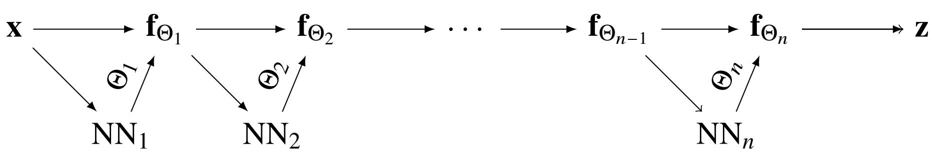


Fig. 2 Illustration of the Normalizing Flows concept.

The transformations $\mathbf{f}_{\Theta_i}$ must be easy to evaluate, invert, and differentiate. Moreover, the determinant of the Jacobian of the transformation must be easy to compute since it is needed for the calculation of the pdf of the complex distribution.

Using samples from the complex distribution, the weights of the NNs are optimized during the training process in a Maximum Likelihood Estimation approach, such that the transformed samples $\mathbf{f}(\mathbf{x})$ from the training dataset match the base distribution $p(\mathbf{z})$. This is done by minimizing the negative log-likelihood of the training data points, which is equivalent to maximizing the likelihood of creating the training data points by sampling from the base distribution and transforming them with the NFs.

However, singularities of the true pdf $p(\mathbf{x})$, i.e., points where the pdf has a value of infinity, can lead to problems during the training process. The reason for this is that it would violate the invertibility property since the inverse transformation $\mathbf{f}^{-1}$ would not be defined at these points. A common solution to this problem, which is also applied in this paper, is the so-called noise injection, where a small amount of noise is added to the training data to dilute the singularities of the true pdf.

3.2 Conditional Normalizing Flows

Using the NFs, the model can be used to transform samples from the complex distribution to the base distribution (density estimation) and to sample from the complex distribution (sampling). However, this is not enough for the problem at hand, since the model should not describe only one distribution, but rather a distribution for any given time t and additional parameters ψ regarding the target dynamics. This can be achieved by using Conditional Normalizing Flows (CNFs) [13], which are a special kind of NFs that can be conditioned on some input to calculate a conditional probability $p(\mathbf{x}|t,\psi)$ in contrast to the unconditional probability $p(\mathbf{x})$ of normal NFs. In this case, t and ψ serve as additional input for the model, the so-called conditioning variables. More precisely, they are used as an additional input for the NNs besides the output of the previous transformation $\mathbf{f}_{\Theta_{i-1}}$. Equation 3 describes the transformation $\mathbf{f}$ and its inverse $\mathbf{f}^{-1}$ as a composition of n invertible transformations $\mathbf{f}_{\Theta_i}$. Θ_i are the parameters of the i-th transformation calculated by NNs using the output from the previous transformation $\mathbf{f}_{\Theta_{i-1}}$ and the conditioning variables t and ψ. This allows the model to learn a different transformation for each combination of the conditioning variables.

$$
\text{NFs}: \quad \begin{cases} \mathbf{z} = \mathbf{f}(\mathbf{x}) = \mathbf{f}_{\Theta_n} \circ \cdots \circ \mathbf{f}_{\Theta_2} \circ \mathbf{f}_{\Theta_1}(\mathbf{x}) \\ \mathbf{x} = \mathbf{f}^{-1}(\mathbf{z}) = \mathbf{f}_{\Theta_1}^{-1} \circ \cdots \circ \mathbf{f}_{\Theta_n}^{-1}(\mathbf{z}) \\ \text{with } \Theta_i = \text{NN}_i(\mathbf{f}_{\Theta_{i-1}}(\mathbf{x})) \end{cases} \tag{2}
$$

$$
\text{CNFs}: \quad \begin{cases} \mathbf{z} = \mathbf{f}(\mathbf{x}, t, \psi) = \mathbf{f}_{\Theta_n} \circ \cdots \circ \mathbf{f}_{\Theta_2} \circ \mathbf{f}_{\Theta_1}(\mathbf{x}) \\ \mathbf{x} = \mathbf{f}^{-1}(\mathbf{z}, t, \psi) = \mathbf{f}_{\Theta_1}^{-1} \circ \cdots \circ \mathbf{f}_{\Theta_n}^{-1}(\mathbf{z}) \\ \text{with } \Theta_i = \text{NN}_i(\mathbf{f}_{\Theta_{i-1}}(\mathbf{x}), t, \psi) \end{cases} \tag{3}
$$

3.3 Application

In our approach, we utilize CNFs to model the distribution of the target's position under varying conditions of time and target dynamics. Specifically, we implement CNFs with a technique known as Masked Autoregressive Flow (MAF) combined with rational quadratic splines.

MAF [17] is a method that leverages autoregressive models to generate samples from a complex distribution. It does so by modeling the distribution as a sequence of conditional distributions where each dimension depends on the previous ones, giving it a high degree of flexibility. This autoregressive property makes it suitable for our task of predicting the target's position under different conditions.

Rational quadratic splines [18] are used as transformation functions $\mathbf{f}_{\Theta_i}$ to further enhance the flexibility and expressive power of the model compared to affine transformations as used in RealNVP [19]. They allow to capture complex patterns in the data by adjusting the shape of the spline as needed, making the model adaptable to a wide range of target distributions.

While a detailed technical description of MAF and rational quadratic splines is beyond the scope of this paper, this combination plays an important role in our approach, enabling us to effectively model the probabilistic distribution of the target's future position.

Using CNFs, the model can be used to transform samples from the complex distribution to the base distribution for a given time t and additional parameters ψ regarding the target dynamics which serve as the conditioning variables. Thus, the pdf $p(\mathbf{x}|t, \psi)$ can be evaluated by transforming a position $\mathbf{x}$ from the complex distribution to the base distribution and calculating the probability of the sample in the base distribution. Moreover, the model can be used to sample from the complex distribution for a given time t and parameters ψ by sampling from the base distribution and transforming the samples with the learned invertible transformations. Not only is the sampling using the CNFs much faster than Monte Carlo simulation, the use of CNFs also allows for an exact likelihood computation, which is not possible with the Monte Carlo trajectory generation method.

Since the model is translation and rotation invariant for the simple scenario used in this paper, the position of the target can be predicted for any initial position and orientation of the target by transforming the position of the target to the origin pointing north. This results in the prediction task depicted in Equation 4, where the value of the pdf $p(\mathbf{x}|t)$ needs to be predicted conditioned on the time t, leading to a function f that maps $\mathbf{x}$ and t to the pdf.

$$
p(\mathbf{x}|t) \approx f(\mathbf{x}, t) \tag{4}
$$

Nevertheless, if the dynamics depend on the initial position and orientation, i.e., if the model is not translation and rotation invariant, the initial position and orientation of the target can be used as additional conditioning variables to allow for the prediction of the position of the target for any initial position and orientation.

4 Data Generation

As a practical illustration of the problem described in section 2, consider a target flying in the horizontal plane with a constant velocity and performing random maneuvers. This scenario serves as an instructive example of a stochastically moving target. Three different types of maneuvers are assumed: left turn, right turn, and straight flight. At the beginning of a trajectory (at the origin, pointing north), the type and duration of the maneuver and the radius of the turn are randomly chosen from a uniform distribution with parameters depicted in Table 1. After the maneuver is completed, new maneuvers are randomly chosen until the total duration of the trajectory is reached. Since no additional parameters are required for this simple scenario, $\psi = \emptyset$.

Trajectories created with this approach are shown in Figure 3. Each dot denotes the change of the maneuver. Since the trajectories are simulated with a time discretization of 0.1 seconds, trajectory data can be obtained and saved for all the simulated time steps.

Figure 4 depicts histograms of samples of the distribution of the target positions for different times. The data was created with a computationally expensive Monte Carlo simulation. To improve training, the position of the target is normalized between -1 and 1 in all dimensions and time.

Table 1 Properties of the target maneuvers.

Property	Value
Trajectory duration	100 s
Time discretization	0.1 s
Target speed	200 m/s
Minimum maneuver duration	5 s
Maximum maneuver duration	50 s
Minimum lateral acceleration	3 m/s^2
Maximum lateral acceleration	20 m/s^2

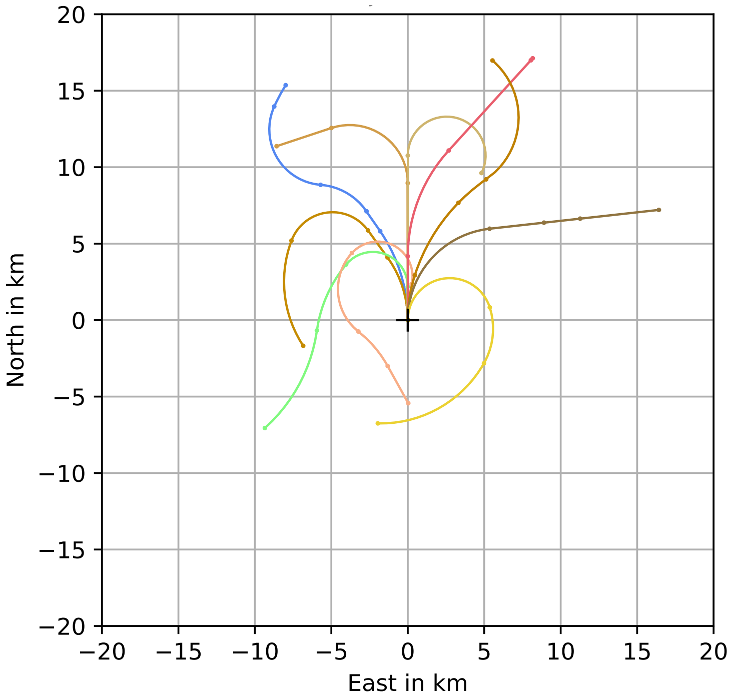


Fig. 3 Ten randomly generated target trajectories with a duration of 100 seconds.

5 Results

With the above-described approach and suitable training data, the distribution of the position of the target at any time t can be predicted. This is demonstrated in subsection 5.1, where the target performs the maneuvers described in Table 1. Further, subsection 5.2 shows the results for targets that can only perform one of three deterministic maneuvers: left turn, right turn, or straight flight, where the radius is fixed and the duration of the maneuver is infinitely long, akin to the Multi-Hypothesis Guidance (MHG) approach [5]. Lastly, subsection 5.3 proves the applicability of the approach to more complex scenarios, where the position of a target following nonlinear dynamics is predicted in three dimensions.

The procedure for the training of the model is always the same: First, the training data is created by simulating the target trajectories with their respective dynamics. According to Table 1, 1e4 trajectories are simulated for each scenario. 80% of the trajectories are used for training and 20% for validation. A separate test set with 1e4 trajectories is used to compare the model against the Monte Carlo simulation.

5.1 Stochastic Maneuvers

First, the scenario described in section 4 is examined. Table 1 depicts the properties of the target maneuvers and Table 2 depicts the parameters of the model. The latent dimension of the model, i.e., the dimension of the base distribution, is set to 2, which means that the model can model the distribution of the position of the target in two dimensions, namely the x- and y-coordinate. Thus, a two-dimensional Gaussian distribution is used as the base distribution that is transformed by the model to obtain the distribution of the position of the target.

The results of the model are shown in Figure 5. The figure depicts the absolute frequency from 1e4 samples (as a representation of the pdf) of the position of the target at different times. When comparing the results to the test data depicted in Figure 4, it can be seen that the model is able to predict the position of the target quite well with a validation loss of -2.85. While the overall shape of the pdf is similar, the model fails to account for the limited velocity of the target, which is why the pdf is more spread out than the test data.

The Monte Carlo trajectory generation method requires about 6.64 seconds[1] of computation time, whereas the CNF model requires only about 0.17 seconds to create 1e4 samples.

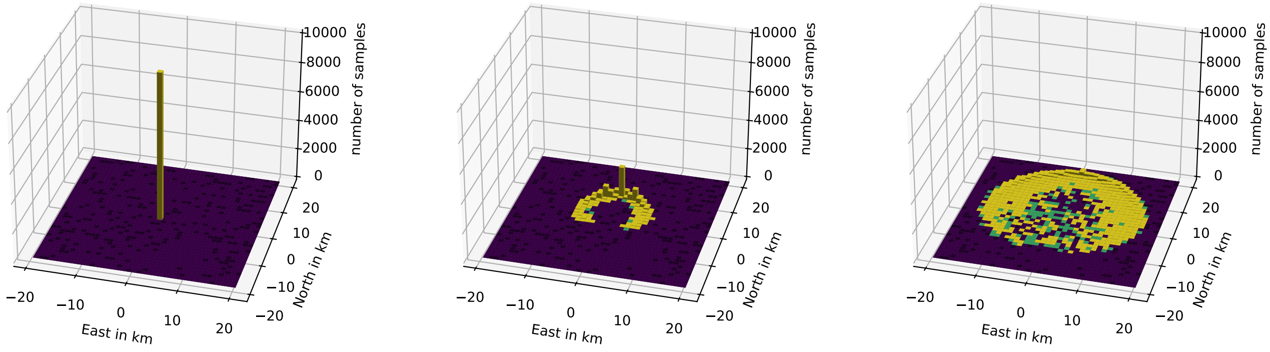
Figure 6 depicts the learned pdf of the position of the target at different times. When comparing the pdf and the histogram of the samples in Figure 5 to the histogram of the training data in Figure 4, it can be seen that the model is able to learn the distribution of the training data quite well.

5.2 Deterministic Maneuvers

To prove that the same model can also be used for deterministic maneuvers, it is applied to predict the position of the target after a deterministic maneuver of infinite duration. The maneuvers tested were flying straight, a left turn, and a right turn, each with a fixed lateral acceleration of 3 m/s^2. Compared to the previous scenario, the target dynamics have changed, thus new training data is created in a similar way as before, but with the new target dynamics. Figure 7 depicts the distribution of the target positions at different instances in the test data.

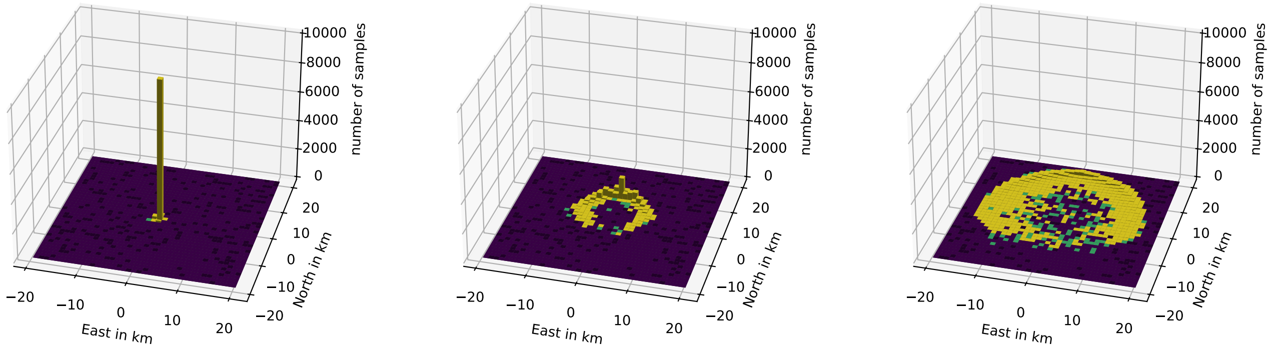
With the help of the new training data, the model is trained to predict the pdf of the position of the target at different times. Figure 8 depicts samples of the learned distribution of the predicted target positions for different times. When comparing it to the test data in Figure 7, it can be seen that the model is able to predict the position of the target quite well. Nevertheless, there are some outliers, which are caused by the fact that the values of the learned pdf are not exactly zero. Thus, the model predicts a very low, but non-zero, probability for the target to be at the wrong position.

[1]All computations were performed with a Ryzen 7 6800U processor with 16 GB RAM.



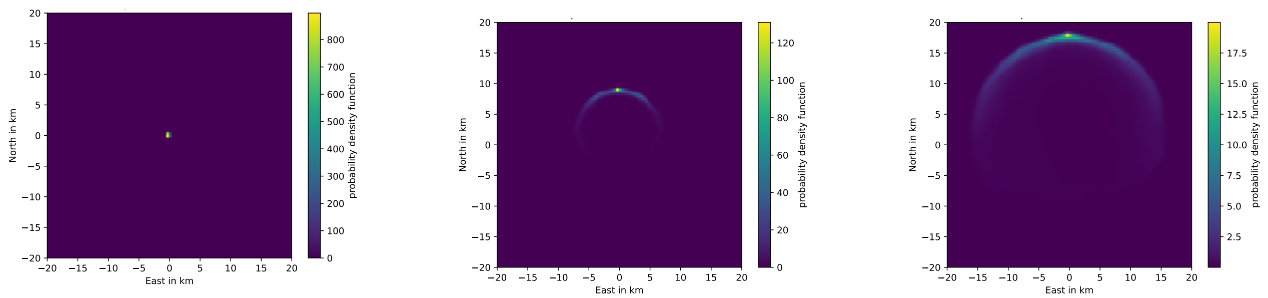
(a) 0 seconds **(b) 45 seconds** **(c) 90 seconds**

Fig. 4 Histograms of 1e4 samples of the probability density function of the position of the target at different times obtained by Monte Carlo simulation.



(a) 0 seconds **(b) 45 seconds** **(c) 90 seconds**

Fig. 5 Samples of the learned probability density function of the position of the target at different times.

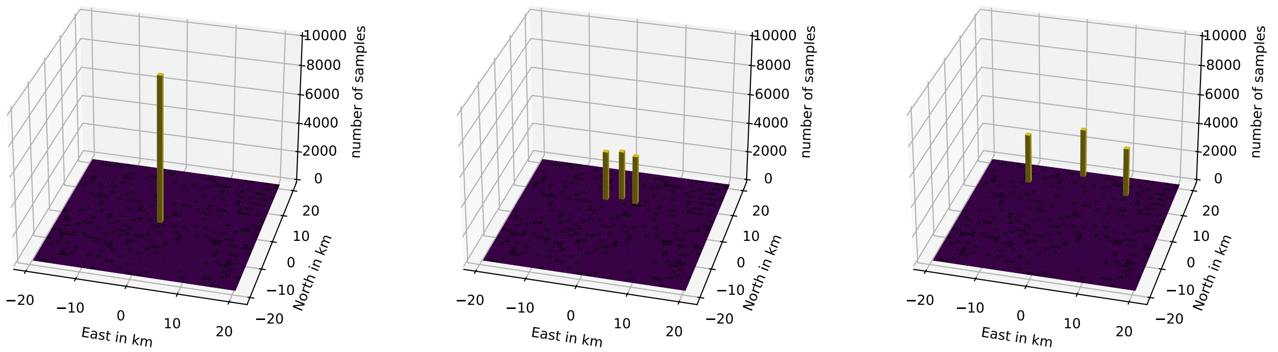


(a) 0 seconds

Fig. 6 Learned probability density function of the position of the target at different times.

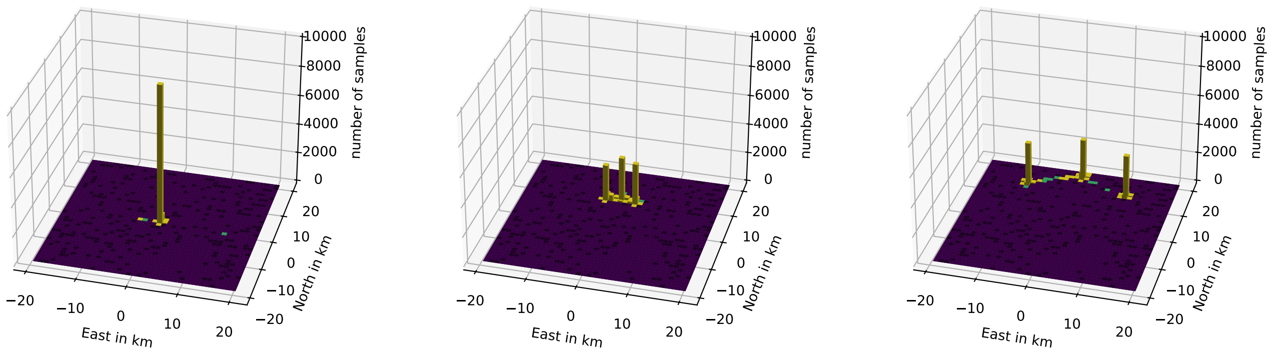
Table 2 Parameters of the model and the training process.

Parameter	Value
Number of flow layers	4
Number of hidden layers	2
Number of hidden units	32
Activation function	ReLU
Batch size	1000
Learning rate	0.003
Number of epochs	1000
Optimizer	Adam
Loss function	Negative log-likelihood
Latent dimension	2
Number of trajectories	1e4
Training data	80%
Validation data	20%
Noise injection standard deviation	0.01



(a) 0 seconds **(b) 45 seconds** **(c) 90 seconds**

Fig. 7 Test data: Histograms of samples of the probability density function of the position of the target at different times.



(a) 0 seconds **(b) 45 seconds** **(c) 90 seconds**

Fig. 8 Learned distribution: Histograms of samples of the probability density function of the position of the target at different times which does not change maneuvers.

5.3 Ballistic Targets

Using the same approach from the previous section, the stochastic trajectory of a ballistic target, flying along a ballistic trajectory influenced by some disturbances, is predicted. Due to the disturbances, the target does not fly the ballistic trajectory exactly, but with some deviations, leading to a distribution of possible trajectories, which is to be modeled. The ballistic trajectory is calculated with a simple ballistic model, which is described by Equation 5 in the North-East-Down frame.

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \mathbf{g} + \mathbf{d} + \mathbf{w} \\
\mathbf{d} &= -\frac{1}{2 \cdot \mathrm{BC}} \cdot \rho \cdot |\mathbf{v}| \cdot \mathbf{v} \\
\mathbf{g} &= \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \\
\mathbf{w} &\sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \\
\mathrm{BC} &= \frac{m}{A \cdot C_D}
\end{aligned} \tag{5}$$

The change of the position $\mathbf{x}$ is equal to the velocity $\mathbf{v}$ and the change of the velocity $\mathbf{v}$ is equal to the sum of the gravitational acceleration $\mathbf{g}$, the deceleration $\mathbf{d}$ due to drag, and the disturbances $\mathbf{w}$. The disturbance $\mathbf{w}$ is modeled as a zero-mean Gaussian noise with a covariance matrix $\mathbf{\Sigma}$.

$\mathbf{d}$ is calculated with the ballistic coefficient BC, which is the ratio of the mass m of the target to the product of the cross-sectional area A and the drag coefficient C_D of the target. The goal of the model is to calculate the distribution of the position $\mathbf{x}$ of the ballistic target after a certain time t.

Table 3 depicts the parameters of the ballistic trajectories used as training data. The ballistic coefficient of the target is sampled from the uniform distribution $\mathcal{U}(200, 800)$ kg/m^2 to allow for the prediction of the distribution of the position of the target for any ballistic coefficient in the learned range.

Table 3 Parameters of the ballistic trajectories used as training data.

Parameter	Value
Trajectory duration	100 seconds
Time discretization	0.1 seconds
Air density ρ	1.225 $\frac{\mathrm{kg}}{\mathrm{m}^3}$
Target ballistic coefficient	$\mathcal{U}(200, 800)$ $\frac{\mathrm{kg}}{\mathrm{m}^2}$
Target initial position	[0, 0, -1000] m
Target initial velocity	[100, 0, 0] $\frac{\mathrm{m}}{\mathrm{s}}$
Covariance matrix $\mathbf{\Sigma}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \frac{\mathrm{m}^2}{\mathrm{s}^4}$
Gravitational acceleration g	9.81 $\frac{\mathrm{m}}{\mathrm{s}^2}$

Compared to the scenarios in subsections 5.1 and 5.2, the dynamics here differ dramatically:

- The target velocity is not constant but changes nonlinearly over time due to drag and the disturbance forces.

- The scenario takes place in three dimensions, instead of two dimensions.
- An additional parameter ψ is used during the training process, namely the ballistic coefficient of the target.

Figure 9 depicts samples of the distribution of the target positions for different times and BC = 500 kg/m^2, obtained by Monte Carlo simulation. It can be seen that the distribution of the target positions widens over time, due to the influence of the noise.

The same CNF model as in the previous subsections is used to predict the distribution of the ballistic trajectories, but with a latent dimension (the dimension of the base distribution) of 3, since the ballistic trajectories are simulated in three dimensions. The training process was similar to the previous subsections and required about 119 seconds.

With the help of the trained model, the distribution of the ballistic trajectory can be predicted without the need to simulate the ballistic trajectory multiple times. The required computation time for the prediction of 1e4 samples using the CNFs is about 0.27 seconds, compared to the 178 seconds of the Monte Carlo simulation which was used to generate the training data.

Figure 10 depicts the samples created for a time of 0, 45, and 90 seconds into the flight of the ballistic trajectory, i.e., samples of $p(\mathbf{x}|t, \psi)$ for $t = 0, 45, 90$ s and ψ = BC = 500 kg/m^2.

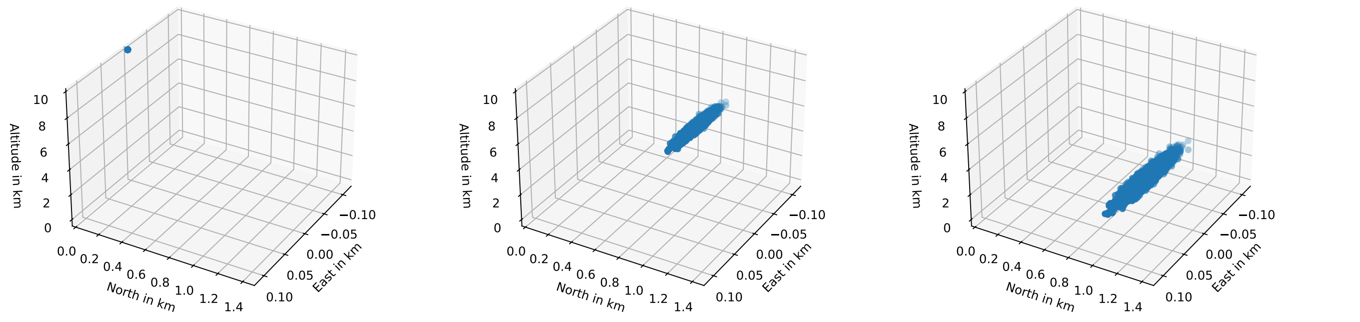
When comparing the learned distribution to the test data depicted in Figure 9, which was obtained through costly Monte Carlo simulation, it can be seen that the model is able to predict the distribution of the ballistic trajectory quite well. While there are some outliers, the overall shape of the distribution is similar to the test data depicted in Figure 9, with a correct position of the mean of the distribution and a correct shape of the distribution. Compared to the 1e4 samples generated by the model, the number of outliers is pretty low.

Only the distribution of the position of the target at 0 seconds is not predicted correctly: Whereas the mean of the distribution is predicted correctly, the shape of the distribution is not predicted correctly, being elongated along the direction of flight. A reason for this could be the training data: According to Table 3, all trajectories start at the position [0, 0, -1000] m, leading to a singularity in the pdf at this position, since the value of the pdf at this position is infinite to conserve an integral of 1. The injection of noise mitigates this problem but does not solve it completely.

6 Conclusion

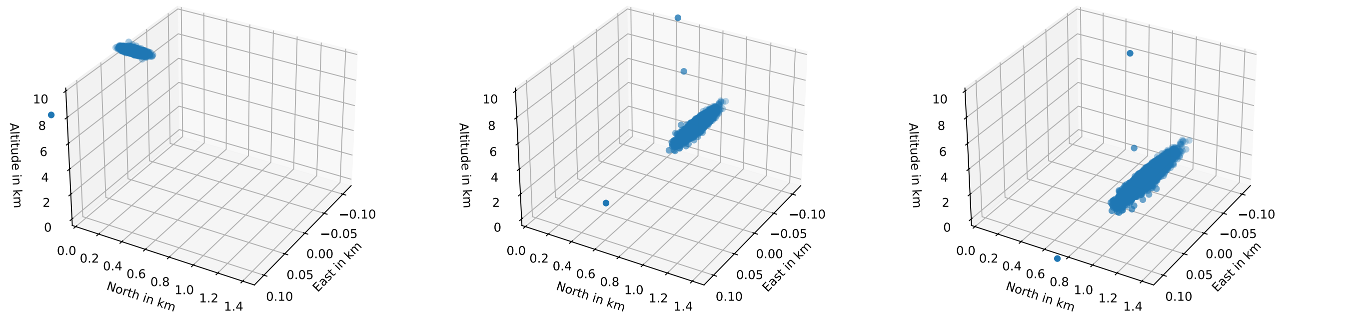
In this paper, an approach for the prediction of the future position of a target for a guided missile is presented. The approach is based on the usage of Conditional Normalizing Flows, which are trained with the help of Monte Carlo simulation data. The presented results demonstrate the method's effectiveness in predicting target positions for various scenarios, including targets with stochastic maneuvers, deterministic maneuvers, and ballistic trajectories with additional parameters. The approach is target-agnostic and can be applied to different target types with appropriate training data.

Using the proposed approach, both the probability density function as well as samples of the future position of the target can be obtained, giving rise to all applications where a prediction model of the target position is needed. The main example of this is the usage of the prediction model in a predictive guidance law, which can utilize the prediction to compute the control commands for the missile. Since most targets do not follow perfectly deterministic trajectories, the usage of a stochastic predictor can take the uncertainty of the target trajectory into account, potentially leading to a more robust guidance law.



(a) 0 seconds (b) 45 seconds (c) 90 seconds

Fig. 9 Monte Carlo simulation: samples of the probability density function of the position of the target with $\mathrm{BC} = 500\ \frac{\mathrm{kg}}{\mathrm{m}^2}$ **at different times.**



(a) 0 seconds (b) 45 seconds (c) 90 seconds

Fig. 10 Samples of the learned probability density function of the position of the target with the additional parameter $\psi = \mathrm{BC} = 500\ \frac{\mathrm{kg}}{\mathrm{m}^2}$ **at different times.**

References

[1] Fred P. Adler. Missile Guidance by Three-Dimensional Proportional Navigation. *Journal of Applied Physics*, 27(5):500–507, 05 1956. ISSN: 0021-8979. DOI: 10.1063/1.1722411.

[2] F. William Nesline. and Paul Zarchan. A new look at classical vs modern homing missile guidance. *Journal of Guidance and Control*, 4(1):78–85, 1981. DOI: 10.2514/3.56054.

[3] Yanning Guo, Matt Hawkins, and Bong Wie. Applications of generalized zero-effort-miss/zero-effort-velocity feedback guidance algorithm. *Journal of Guidance, Control, and Dynamics*, 36(3):810–820, 2013. DOI: 10.2514/1.58099.

[4] Prasiddha Dwivedi, Prashant Bhale, Abhijit Bhattacharyya, and Radhakant Padhi. Generalized estimation and predictive guidance for evasive targets. *IEEE Transactions on Aerospace and Electronic Systems*, 52(5):21111–2122, 2016. DOI: 10.1109/TAES.2016.140861.

[5] Marc Schneider and Walter Fichter. *Multi-Hypothesis Guidance With Interacting Multiple Model Filter*. DOI: 10.2514/6.2022-1846.

[6] Renhao Huang, Hao Xue, Maurice Pagnucco, Flora Salim, and Yang Song. Multimodal trajectory prediction: A survey. *arXiv preprint arXiv:2302.10463*, 2023.

[7] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, 2019. `https://api.semanticscholar.org/CorpusID:204509212`.

[8] Shane T. Barratt, Mykel J. Kochenderfer, and Stephen P. Boyd. Learning probabilistic trajectory models of aircraft in terminal airspace from position data. *IEEE Transactions on Intelligent Transportation Systems*, 20(9):3536–3545, 2019. DOI: 10.1109/TITS.2018.2877572.

[9] K. Guo, W. Liu, and J. Pan. End-to-end trajectory distribution prediction based on occupancy grid maps. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2232–2241, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society. DOI: 10.1109/CVPR52688.2022.00228, https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00228.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020. ISSN: 0001-0782. DOI: 10.1145/3422622.

[11] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[12] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. https://proceedings.mlr.press/v37/rezende15.html.

[13] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[14] Christoph Schöller and Alois Knoll. Flomo: Tractable motion prediction with normalizing flows. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7977–7984, 2021. DOI: 10.1109/IROS51168.2021.9636445.

[15] Yecheng Jason Ma, Jeevana Priya Inala, Dinesh Jayaraman, and Osbert Bastani. Likelihood-based diverse sampling for trajectory forecasting. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13259–13268, 2020.

[16] T. Maeda and N. Ukita. Fast inference and update of probabilistic density estimation on trajectory prediction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9761–9771, Los Alamitos, CA, USA, oct 2023. IEEE Computer Society. DOI: 10.1109/ICCV51070.2023.00898, https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00898.

[17] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf.

[18] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper_files/paper/2019/file/7ac71d433f282034e088473244df8c02-Paper.pdf.

[19] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. https://openreview.net/forum?id=HkpbnH9lx.